UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Habilitation à Diriger des Recherches

Spécialité : **Informatique**

Présentée par

Manuele KIRSCH PINHEIRO

Maître de Conférences – Section CNU 27

Apports de la Notion de Contexte à Différents Systèmes

(*Context across the systems*)

Habilitation soutenue publiquement le 13 Janvier 2021 devant le jury composé de :

Philippe Lalanda, Professeur, Université de Grenoble, Rapporteur
Philippe Roose, Maître de Conférences HDR, Université de Pau, Rapporteur
Florence Sedes, Professeur, Université de Toulouse, Rapporteur
Yolande Berbers, Professeur, Katholieke Universiteit Leuven, Examinatrice
Agnès Front, Professeur, Université de Grenoble, Examinatrice
Chantal Taconet, Maître de Conférences HDR, Telecom Sud Paris, Examinatrice
Massimo Villari, Professeur, Université de Messine, Examinateur
Bénédicte Le Grand, Professeur, Université Paris 1 Panthéon Sorbonne
Carine Souveyet, Professeur, Université Paris 1 Panthéon Sorbonne

Centre de Recherche en Informatique (EA 1445)

Context across the systems

Abstract

This document brings together several researches works carried out between 2002 and 2020, discussed in the habilitation entitled "*Apports de la Notion de Contexte à Différents Systèmes*". It synthetizes then the original document, written in French, presenting the problem statements tackled in the original document and summarizing the proposals that have been made, which are presented in more details thanks to the papers joined in annex. Thus, different contributions, crossing several Computer Science communities (CSCW, Ubiquitous Computing and Information Systems), are discussed here. All these contributions share a common guideline, the notion of context, which is applied all along these contributions on different kinds of system (notably Groupware Systems and Information Systems). All these contributions lead to a common perspective, a new generation of Information Systems called here Pervasive Information System, in which the notion of context plays a key role for adaptation purposes. Elements presented in this document intend then to synthetize all the different contributions leading to this vision.

Contents

List of Illustrations

List of Tables

I Introduction

This document synthesizes an important part of my research work, which can be characterized by the application of the notion of context into different kinds of systems (Groupware systems, middleware systems or Information Systems). This research is the result of a career started in 2006, date of my PhD thesis defense, and that continues until today (2020), at the University Paris 1 Panthéon Sorbonne.

All along my career, I had the opportunity and the privilege to join several research teams, working in different areas of Computer Science. First of all, between October 2002 and September 2006, I prepared my PhD thesis at the University Joseph Fourier - Grenoble I, within the SIGMA team, a recognized team in the field of Information Systems, whose "multimedia" axis (currently known as STEAMER team) was specialized on the adaptation of Web-based systems. Within the "multimedia" axis, I could carry out my research on the adaptation of Groupware systems, and more particularly, on the adaptation of group awareness information, whose support is an outstanding characteristic of this kind of software application. It was during my PhD thesis that I have started working with the notion of context, notably through the proposal of a context model that considers both physical and organizational aspects.

After my PhD thesis, I could continue my research in the CSCW (Computer Supported Cooperative Work) community, from which come the notions of groupware and group awareness, by integrating the ECOO team (now COAST team) at LORIA in Nancy, during a one-year position (ATER from 2006 until 2007) at the IUT Nancy Charlemagne of the University Nancy 2.

On September 2007, I had the opportunity to join the prestigious K.U.Leuven for a postdoctoral position within the DistriNet laboratory, as part of the European project IST-MUSIC. During this period (2007-2008), I was confronted with a new environment, and a new team working on Distributed Systems. Working in the community of Ubiquitous Computing, I had to learn new concepts for me, such as peer-to-peer networks and the notion of service, in order to better adapt myself to this new environment and thus bring a real added value to my team and to the IST-MUSIC project.

Starting on September 2008, I joined the “Centre de Recherche en Informatique” of the University Paris 1 Panthéon Sorbonne, as Associate Professor. Once again, I had to adapt myself to a new team, particularly renowned on Information System, and notably for its skills in Requirements Engineering as well as in Service Engineering. In this new environment, I had to learn in new concepts for me, such as the notion of intention, which I have integrated to my own research, while bringing to the team my skills concerning the notion of context.

All along these years, I have the opportunity to integrate several teams belonging to different communities on Computer Science, by incorporating concepts from these communities into my research and by bringing my own contributions in return. The notion of context appears thus as a guiding thread for my research, since this notion has been applied in all the communities I have crossed during my career (CSCW, Ubiquitous Computing, Information Systems). Like a backbone, the notion of context has been applied in these different communities, with contributions for each one.

All these experiences and contributions converge today on the evolution of the Information Systems (IS), in what we call here Pervasive Information Systems (PIS). Indeed, the introduction of new technologies and trends in IS leads inevitably to their evolution towards a new generation of Information Systems, the Pervasive Information Systems. These new technologies primarily impact the infrastructures used by these systems, but their influence is not limited to this purely technical level. All levels of an Information System can be impacted. In a schematic vision, we may consider that these new technologies and the opportunities they bring are likely to influence not only the infrastructures, but also the services offered, the applications and business processes, and even the management of these systems (see Figure 1).

New technologies, such as IoT, Cloud and Fog Computing, are bringing more dynamism to Information Systems, and are enabling more flexible IT systems that are better able to adapt themselves to changes on their environment. The notion of context may contribute to achieve this flexibility, which has become necessary in order to take better account of the dynamic environment in which Information Systems are gradually moving towards. Context information can thus be captured and fed back up, level by level, like events, and contribute to the adaptation of each level and of the system as a whole.

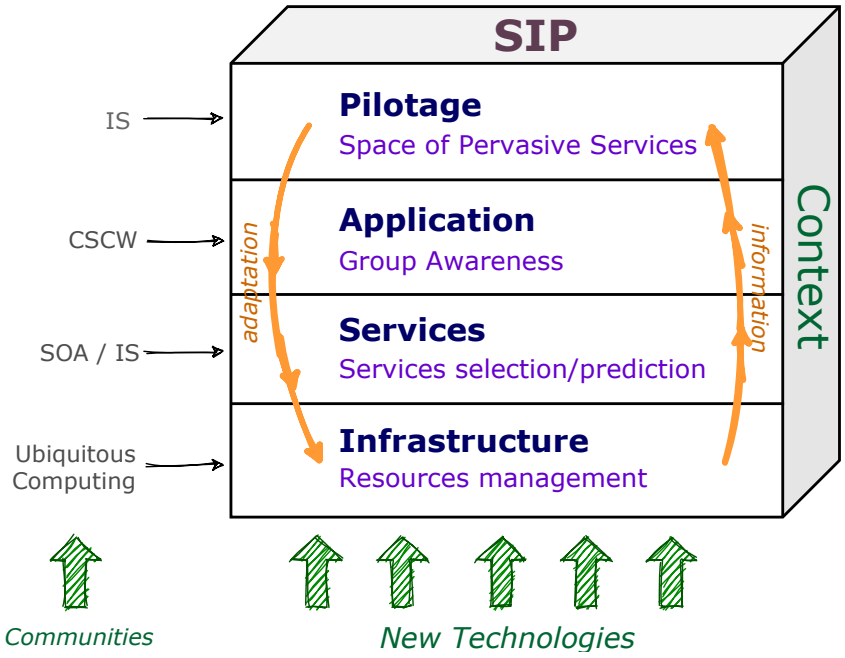


Figure 1. Contributions proposed in this document organized in a schematic view of a Pervasive Information System.

Over the years, the contributions I have made into these various communities find a direct application in this vision, illustrated in Figure 1. Each contribution presented in this document (noted in purple in Figure 1) is applicable to a different level of these systems. In each contribution, the notion of context appears as the key element allowing adaptation at a given level.

All the contributions presented in this document thus converges towards this new generation of IS. These contributions are presented thereafter, not in chronological order, but rather in a logical order (represented in Figure 2), organized according to the community in which these contributions were originally proposed. Besides, since the notion of context represents the common thread of this work, it is important to establish a common understanding of this concept and its characteristics. We therefore begin this document with a state of the art concerning the notion of context and its engineering. This "roadmap", originally proposed on [82], synthesizes my experience and my vision concerning the notion of context and its application (what we mean here by its engineering). This effort of popularization, initially intended for teaching, establishes the necessary bases for the understanding of this concept through a set of dimensions considered as necessary for its engineering. These dimensions find then their application in the various contributions presented in this document.

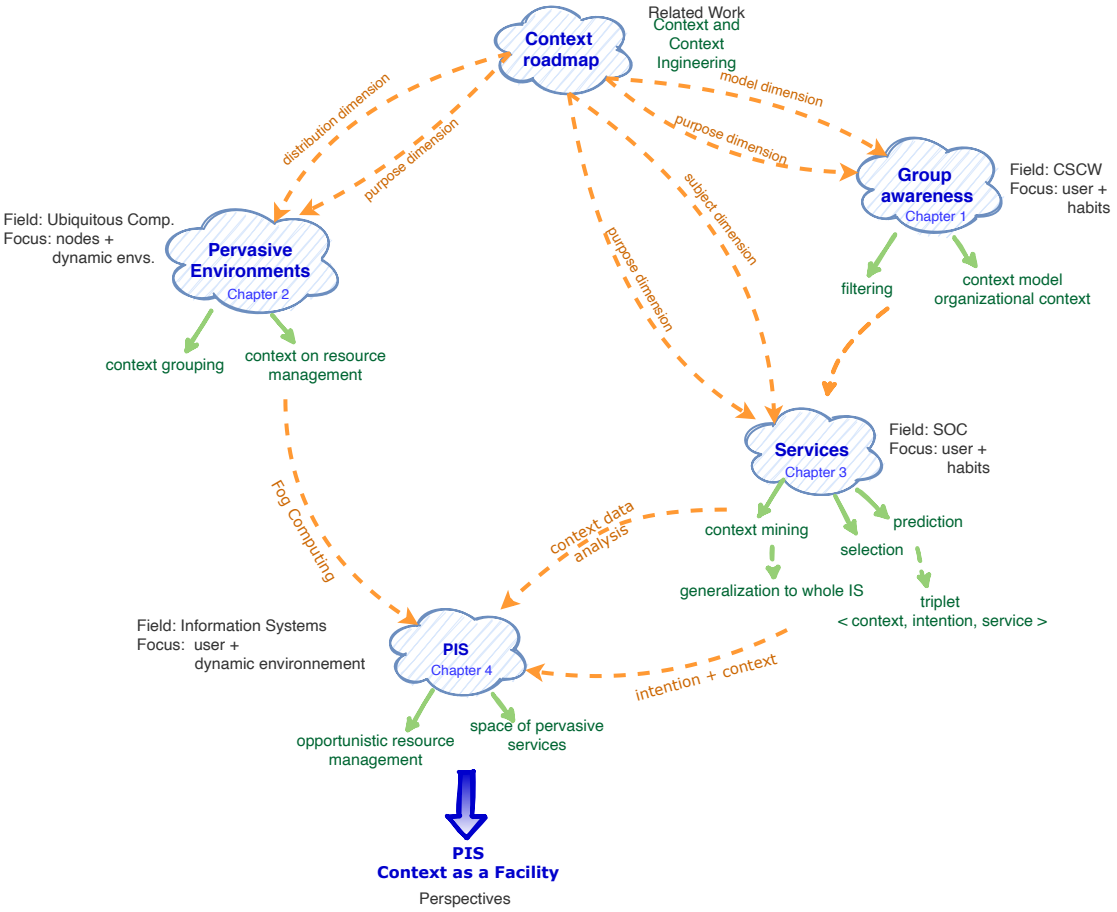


Figure 2. Cartography of the proposed contributions and their evolution.

The first of these contributions (Chapter 1) applies to the CSCW community. It concerns the work carried out mainly during my PhD thesis (2002-2006). This work aimed at adapting group awareness information, focusing particularly on end-users, organized in teams. This research raises the question of work practices and habits, proposing two major contributions: a context model including organizational aspects; and a filtering mechanism for group awareness information based on profiles representing users' habits.

These contributions, or rather their influence, can be found in my research work on the SOC (Service Oriented Computing) community. This work (chapter 3), carried out between 2008 and 2014, maintain this focus resolutely turned towards end-users and their work practices through three contributions on the selection and on the prediction of services, as well as a contribution on the analysis of contextual data, which we call here "*context mining*". This latter considers a major issue highlighted by the roadmap, the relevance of context information for a user, and *a fortiori*, for an application. This work also tackles the challenge of generalizing the context support to a whole Information System. Moreover, our researches on the selection and prediction of services introduces the triplet "*< intention, context, service >*" according to which a service is proposed in order to satisfy user's intentions in a certain context of use.

These contributions explored different dimensions present in the roadmap, including the "*model*" and "*subject*" dimensions. These contributions have been completed by more technical works, anchored in the Ubiquitous Computing community. These researches (chapter 2), carried out between 2008 and 2016, explore another dimension highlighted by the roadmap, the *distribution* of context data. They

consider particularly the dynamic nature of pervasive environments. The focus is no longer on the user (at least not directly), but on the environment itself, which becomes more and more dynamic, and on the nodes that compose this environment. These contributions use context information for promoting a better use of the resources in such dynamic environment. They also introduce in my work a new community, the Fog Computing community, and the possibility of using proximity resources to perform various tasks as advocated by this community.

All of these contributions converge on the evolution of Information Systems towards Pervasive Information Systems (PIS). This new generation of systems is the subject of my latest research, carried out from 2016 onwards and presented in chapter 4. This research, anchored in the Information System community, no longer focuses only on user or only on the environment, but on both, together, notably through the definition of a conceptual framework for Pervasive Information Systems, called Space of Pervasive Services, as well as on the study of an opportunistic resource management adapted to this new generation of Information Systems.

Figure 2 proposes a cartography illustrating all these contributions and their reciprocal influences. The contributions illustrated in Figure 2 represent the most significant contributions of my career, those in which I have had a more significant impact on the community considered by the contribution. This work carried out in several different teams and communities, which illustrates my vision of what a research work in Computer Science should be: a collaborative work, of collective construction, built through the contribution of each one to the resolution of different research problems. When integrating each community, I had to assimilate several concepts and practices in order to understand the issues handled by each community and then bring my own added value back to the community. As this work were therefore carried out within different teams, the rest of this document is voluntarily written in the first-person plural (“we”), in order to underline what is perhaps the major characteristic of a Pervasive Information System, its pluridisciplinarity.

This document is thus organized as follows: we start with a state of the art on the notion of context (Part II), before introducing the contributions. Part III details the contributions. Chapter 1 introduces my contributions in groupware systems; Chapter 2 presents the contributions made around pervasive systems; while Chapter 3 presents my contributions in service-oriented systems. Finally, chapter 4 introduces my research around Pervasive Information Systems, before concluding (Part IV) with my perspectives and future works.

II State of art

The notion of context can be defined as any piece of information that characterizes the situation of an entity, whatever this entity could be a person, a place or another object (user, application, etc.), considered relevant for the interaction between the user and the application [37]. This notion has accompanied my research since my PhD thesis work. It has been applied to different research problems, and to various communities in Computer Science. It is important to understand this notion and its characteristics before being able to discuss the contributions I had proposed using it. Thus, this chapter presents a state of the art on the notion of context. Derived from [82], this state of the art synthesizes my experience and my perception of the notion of context and its engineering. Intended first of all for the popularization of the notion of context and its engineering among computer science students, this state of the art is presented in the form of a roadmap, presenting the different dimensions necessary to take into account this notion in software.

1 Context Roadmap

The notion of context is becoming more and more used today within applications that could be called "intelligent" (or "smart"), because they are able to observe the environment and to react accordingly. This phenomenon can already be observed with a growing number of applications capable of observing elements of the environment, such as the user's location, her/his physical activity, etc. The observation of the environment is now possible thanks to different types of sensors and technologies. The development of sensors, actuators, nanocomputers and other low-cost technologies related to the Internet of Things (IoT) allows developers to easily propose applications that observe and interact with the physical environment. This kind of application is already part of our daily life but, in most cases, its development is still performed in an ad hoc manner, despite all the research that has been done on the notion of context and on context-aware applications (applications capable of adapting their behavior to the changes observed on its context of use [6,7]). Today, the main challenge no longer lies in the technologies themselves, but mainly in understanding the challenges and the issues that may raise when exploring this notion and all the possibilities offered by these new technologies. Indeed, in order to go further in the use of technology, it is necessary to better understand the notion of context and its issues, since this notion is central to the design and to the implementation of such new "smart" solutions.

Context-aware systems can be seen as applications capable of responding to these challenges. They are defined as applications capable of observing changes in their execution context and of adapting their behavior accordingly [6,7]. Compared to traditional applications, context-aware applications can be considered more complex because they have to cope with heterogeneous and dynamic environments. They must operate, often continuously, under changing conditions. They need to observe different elements located in their environment and to react to their changes, often using limited computing resources (*e.g.* nanocomputers or smartphones with battery and connectivity constraints). Such a dynamic and constrained execution environment has a significant impact on the architecture and on the development of such software applications, particularly in terms of modularity, integration, interoperability and a growing number of non-functional constraints (e.g. robustness and scalability). Under these conditions, the qualities traditionally expected from software applications, such as flexibility, modularity and extensibility, become more difficult to meet, especially with ad hoc development processes, which are often adopted when developing context-aware applications, as observed in [7, 8].

One aspect in particular makes the development of these applications more complex than “traditional” applications: the notion of context itself. This notion corresponds to a broad and ambiguous concept that has been studied and defined in several different ways, both in Computer Science and in other science fields [12,17,18,108]. Supporting this notion in a computer application raises several challenges ranging from the identification of relevant context information, its acquisition and modeling, to its interpretation and exploitation for different purposes [6,12,84,85]. It becomes quickly arduous for non-expert designers to design and to build new applications using this notion.

Understanding the notion of context and its support is a complex but necessary task. It is complex because the notion of context is itself a complex and ambiguous notion, whose integration within an application involves several technical issues. It is necessary because it is only by understanding this notion and the ways in which it can be integrated into an application that one can truly explore its full potential and all the opportunities it opens up. Only a better understanding of this notion will allow the establishment of a true context engineering process, allowing the development of new complex and extensible context-aware applications. More than ever, it becomes necessary to educate and to prepare a new generation of designers and developers capable of reasoning around context elements in the same way they are trained to handle concepts such as components and object-oriented programming.

It is therefore important to provide these young, non-expert designers with the necessary knowledge to reason about the challenges and issues related to context management and support. Through a literature review, we were able to identify a set of dimensions that can be considered as necessary for such support [84], and we could analyze the impact of the quality of context information on these dimensions [85]. These dimensions have been seen as guidelines in a requirement analysis process, helping non-expert designers to identify the possible issues around the support of the notion of context in context-aware applications. This review, detailed on [82], highlighted existing solutions and open questions related to context support and management. This study is intended to serve as a basis for the training of new "context engineers", capable of understanding and building new context-aware applications, especially for tomorrow's Information Systems.

Besides, this state of the art also considers the study of three application scenarios that illustrate possible uses cases that can take profit of context information, as well as the challenges for their implementation. These scenarios were used as examples to illustrate different issues raised by each dimension of the roadmap. Also, a study, carried out with a group of 50 master degree students, is discussed on [82]. The objective of this study is to better understand the perception that these young developers have of the notion of context. The roadmap and each of its dimensions are thus discussed, both in its functional aspects, but also in the consideration of qualitative aspects, which are particularly relevant when considering context information. The roadmap thus raises many questions that must be taken into consideration when designing context-aware applications. Several clues and elements of answer that are highlighted by the literature review were also discussed for each dimension.

The proposed roadmap was fully detailed in the journal paper indicated below [82]. It is presented in the Annex I.

- Kirsch-Pinheiro, M. & Souveyet, C. "Supporting context on software applications: a survey on context engineering" (« Le support applicatif à la notion de contexte : revue de la littérature en ingénierie de contexte »), *Modélisation et utilisation du contexte*, 2(1), **2018**, ISTE OpenScience. Available: https://www.openscience.fr/Le-support-applicatif-a-la-notion-de-contexte-revue-de-la-litterature-en/ (Last visit: Oct. 2020).

2 State of art: Final considerations

The notion of context has been widely explored in various ways through different applications. This use is likely to progress in the near future, notably thanks to the democratization of the IoT and its technologies, which allow easy observation of the physical environment using inexpensive devices. Nevertheless, the notion of context remains an obscure and ambiguous concept. The question of which information can be considered as part of context and which information is not, illustrates quite well this issue. Information such as the available memory, the battery level or the role of the user in an organization can be considered as a context element by some application [50, 127, 140], or as simple parameters by others [65, 142]. Some authors, including [26], have tried to distinguish between context data and application data. For these authors, context data correspond to a set of parameters, which are external to the application and which influence the behavior of the application. Despite efforts to clarify this distinction, the boundaries remain often blurred, as does the notion of context itself, which is often misunderstood by many young software designers.

This same ambiguity is also visible between context-aware applications and the so-called “self-adaptive” applications. The smart agriculture and GridStix scenarios proposed in the roadmap [82] illustrate these ambiguities. Both use context information to adapt their behavior, but the authors of the latter consider it as a self-adaptive application [142]. According to Khan [75], the concepts of context-awareness and self-adaptation are often confusing because self-adaptive applications use to adapt their behavior in response to stimuli from context information. It is therefore difficult to make a clear distinction between these two concepts. Both can be seen as adaptive systems which, according to Colman *et al.* [30], aim at achieving a certain goal by defining a form of loop in which the environment and/or system itself is monitored, the information collected is analyzed, a decision is made about necessary changes, and these changes are then implemented by the system. For these authors, "self-awareness" means that changes can often be processed automatically compared to conventional systems that require offline redesign, implementation or redeployment. This is also true for context-aware systems, since they adapt their own behavior, without human intervention, according to the changes observed in the context information. Although some authors have tried to make some distinction between these concepts [30,75], the most important question is not really these potential differences (if they really exist), but the support of context information in these systems. These two particularly complex concepts are based on context information, a very dynamic, heterogeneous and, moreover, uncertain kind of information. Context management thus raises several challenges that must be taken into consideration when developing a system that observes this notion.

Thus, the main question is how to manage and to exploit context information in a given system? As Coutaz *et al.* [32] pointed out, it is commonly accepted that context information concerns the evolution of a structured and shared information space, and that this space is designed to serve a particular purpose. Whatever information is considered as context depends profoundly on the system in question and on its objectives. Whatever this information is, it needs to be managed appropriately in order to realize its full potential. This requires an understanding on the challenges involved in using this notion and on its main characteristics, such as its heterogeneity, dynamism and uncertainty.

The main objective of the roadmap discussed in [82] is precisely to contribute to this understanding. This is particularly necessary in this document, since all along my career I have applied the notion of context into different systems, which often implied considering some of the issues highlighted by this roadmap. Without this understanding of the notion of context and its support, it is difficult to fully understand the research issues raised by each of the contributions presented in this document.

III Contributions

1 Context on Groupware Systems

1.1 Problem statement

This chapter presents the contributions resulting directly from my PhD thesis work (2002-2006) on the adaptation of group awareness information in groupware systems, especially those supporting a mobile use. The notion of group awareness designates a set of information through which the members of a group, while engaged in their individual activities, capture what other participants do (or do not) and may then adjust their own activities accordingly [55, 144]. Group awareness information can be defined as the knowledge a user has about the group, her/his colleagues and their activities, which provides a context for her/his own individual activities. This context is used to ensure that individual contributions are relevant to the group as a whole, and to evaluate individual actions in relation to the group's goals and progress [40]. This notion is indispensable for groupware systems, in which it heavily contributes to the team coordination in this kind of software exclusively dedicated to teamwork.

Group awareness information plays a very important role in coordinating team's activities notably when considering teams working in an asynchronous mode or in geographically remote way. Unfortunately, the sad reality of 2020 has put back in the spotlight this kind of software and the need for coordination on teams working in a distributed manner. Many employees who had to work at distant because of the pandemic situation (during lockdown and even afterwards) felt that they have lost contact with their colleagues and the activities performed by these colleagues during this period. The lack of adequate support for group awareness information was then cruelly felt.

Even before pandemic crisis, the need for an adequate group awareness information was already underlined by the literature, particularly when considering distributed and mobile teams. Indeed, as new technologies have freed up teams, working in mobile situations has become a reality, the adage "*anytime, anywhere*" being now applied to our daily professional life. Nevertheless, the risk of losing contact with other team members has also intensified with this new mobility, increasing the importance of adequate group awareness support.

When correctly observed, group awareness information can be abundant and the risk of cognitive overload becomes real: if the complete set of available group awareness information is proposed to the user, she/he risks being "overloaded" by all this information, preventing her/him from assimilating the relevant information. According to Bouthier [15], on the one hand, the information exchanged and presented can be very abundant, especially when the group is composed by a large number of active members or when many artifacts are manipulated. On the other hand, the user's mental resources, such as memory and attention, are limited. Yet, the user must interpret and integrate this information in order to coordinate her/his own actions on the group. The cognitive overload occurs when the user is faced with too much information to process. The user then experiences a stressful situation that can lead him or her to reject all of the proposed information. This stress can cause difficulties for a group member, which can lead to disruptions in the user's performance and on the information flow that may penalize the group as a whole.

The user in a situation of mobility is potentially confronted to a constrained environment (e.g. terminals with limited capacities, limited connectivity, inadequate environment, noisy, etc.), which improves the risk of cognitive overload. It is therefore imperative to adapt as much as possible group

awareness information in order to reduce the risk of cognitive overload. In other words, the user must not spend more time becoming aware of what is happening in her/his team than she/he does performing the tasks assigned to her/him.

Hence, it is necessary to reduce the total volume of information presented to the user, in order to proposed her/him only the most relevant one, offering her/him the "right information" at the "right moment". As this user is mobile and potentially equipped with a terminal with restricted capacities, it becomes important to consider the context in which this user is accessing group awareness information. Group awareness information should then be adapted to the user's context, but also to the user's particular interests regarding this kind of information. Indeed, the relevance of group awareness information may vary, for the same user, depending on the context in which he or she accesses the information. Users in groupware system tend to develop some work habits and practices, forming a kind of routine (e.g. consulting their messages in certain places or from certain devices, using certain terminals in particular for the performance of certain tasks, etc.). This routine may have an influence on the user's preferences concerning the group awareness information.

Two sub-problems then arise from the issue of adapting group awareness information to the user's context: firstly, the question of the relevance of group awareness information, and secondly, the question of the representation of the notion of context within a groupware system. On the one hand, there is the question of the expression of the user's preferences and the adaptation process itself, which should take these variable preferences into account. On the other hand, a groupware being a collaborative application, it is important to consider the user not only as an individual, but also as a team member. The notion of physical context, commonly considered at this time (2002-2006) by context-aware applications, proved to be too narrow for a groupware system. It was necessary to enlarge the concepts considered by the notion of context and its support thanks to an appropriate model that takes into account also collaborative aspects of groupware systems.

To sum up, two main issues were tackled by this research work: the adaptation of group awareness information through a context-aware filtering mechanism, as well as an object-oriented context model that takes into account the physical and organizational aspects that characterize users on a groupware system. The proposed filtering mechanism filters group awareness information based on a set of user's profiles representing her/his preferences for this kind of information in a given context. Each profile is then associated with a context description representing a situation in which these preferences are valid. Such description uses a context model, which includes both information about the user's physical environment and about her/his organizational environment. Such "organizational context" allows considering the user not only individually, but also as part of a group (or an organization), which is particularly important for groupware systems. During the filtering process, the context description associated with the user's profiles and the current user's context are compared using similarity measures, defined based on the proposed context model, taking profit of the class and associations defined in this object-oriented model.

The proposed context model was originally introduced in [89]. This paper is reproduced in the Annex II.

- **CRIWG 2004** [89] **:** M. Kirsch-Pinheiro, Jérôme Gensel, Hervé Martin, "Representing Context for an Adaptative Awareness Mechanism". In: Gert-Jan de Vreede, Luis A. Guerrero, Gabriela Marín Raventós (eds.), 10th International Workshop Groupware: Design, Implementation and Use, **CRIWG 2004**, LNCS 3198, 339-348 (**2004**)

1.2 Bibliometrics

This research on groupware systems has produced several publications, mainly between 2003 and 2008, including a PhD thesis in 2006. Out of a total of 17 publications, 11 of which were published during the thesis (2002-2006), and the remaining ones after 2006. Nine of those, listed below, can be highlighted by the number of citations or by their content. These publications have been analyzed, using scholar.google.com and www.researchgate.net in relation to their number of citations, as an indicator of their impact. These citations were ranked according to their publication date in three periods: before 2008, between 2008 and 2013, and after 2013. Self-citations have also been accounted for. Table 1 details the data obtained, illustrated in Figure 3.

- **COMIND 2003** [78]**:** M. Kirsch-Pinheiro, José Valdeni de Lima, Marcos R. S. Borges, "A framework for awareness support in groupware systems". **Computers in Industry**, 52(1): 47-57 (**2003**)
- **CRIWG 2004** [89]**:** M. Kirsch-Pinheiro, Jérôme Gensel, Hervé Martin, "Representing Context for an Adaptative Awareness Mechanism". In: Gert-Jan de Vreede, Luis A. Guerrero, Gabriela Marín Raventós (eds.), 10th International Workshop Groupware: Design, Implementation and Use, **CRIWG 2004**, LNCS 3198, 339-348 (**2004**)
- **MATA 2004** [90]**:** M. Kirsch-Pinheiro, Jérôme Gensel, Hervé Martin, "Awareness on Mobile Groupware Systems". In: Ahmed Karmouch, Larry Korba, Edmundo Roberto Mauro Madeira (eds.), First International Workshop on Mobility Aware Technologies and Applications, **MATA 2004**, LNCS 3284, 78-87 (**2004**)
- **SAC 2005** [91]**:** M. Kirsch-Pinheiro, Marlène Villanova-Oliver, Jérôme Gensel, Hervé Martin, "Context-aware filtering for collaborative web systems: adapting the awareness information to the user's context". In: Hisham Haddad, Lorie M. Liebrock, Andrea Omicini, Roger L. Wainwright (eds.), Proceedings of the 2005 ACM Symposium on Applied Computing (**SAC 2005**), 1668-1673 (**2005**)
- **CSCWD 2005** [92]**:** M. Kirsch-Pinheiro, Marlène Villanova-Oliver, Jérôme Gensel, Hervé Martin, "BW-M: a framework for awareness support in Web-based groupware systems". In: Weiming Shen, Anne E. James, Kuo-Ming Chao, Muhammad Younas, Zongkai Lin, Jean-Paul A. Barthès (eds.), Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design, **CSCWD 2005**, Volume 1, 240-246 (**2005**)
- **PhD thesis** [77]**:** M. Kirsch-Pinheiro, « Adaptation Contextuelle et Personnalisée de l'Information de Conscience de Groupe au sein des Systèmes d'Information Coopératifs », **PhD Thesis**, Université Joseph Fourier - Grenoble I, Grenoble, France (**2006**)
- **CAISE 06 Workshop** [93]: M. Kirsch-Pinheiro, Marlène Villanova-Oliver, Jérôme Gensel, Hervé Martin, "A Personalized and Context-Aware Adaptation Process for Web-Based Groupware Systems". Proceedings of the **CAISE 06, Workshop** on Ubiquitous Mobile Information and Collaboration Systems, **UMICS 2006** (**2006**)
- **Ubicomm 2008** [94]**:** M. Kirsch Pinheiro, Marlène Villanova-Oliver, Jérôme Gensel, Yolande Berbers, Hervé Martin, "Personalizing Web-Based Information Systems through Context-Aware User Profiles", *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, **Ubicomm 2008**,* (**2008**)
- **EGC 2008** [51]**:** Jérôme Gensel, Marlène Villanova-Oliver, M. Kirsch-Pinheiro, « Modèles de contexte pour l'adaptation à l'utilisateur dans des Systèmes d'Information Web collaboratifs », *8èmes Journées Francophones d'Extraction et Gestion des Connaissances (**EGC'08**), Atelier sur la Modélisation Utilisateur et Personnalisation d'Interfaces Web,* 5-15 (**2008**)

Table 1. Bibliometric analysis of selected publications.

Reference	Year	Total	≤ 2008	> 2008 & ≤ 2013	> 2013	Self-citation
COMIND 2003	2003	90	49	20	10	11
CRIWG 2004	2004	71	13	21	13	24
MATA 2004	2004	11	4	4	1	2
ACM SAC 2005	2005	45	16	22	4	2
CSCWD 2005	2005	8	2	3	2	1
PhD thesis	2006	8	4	2	1	1
CAiSE 06 Workshop	2006	13	0	7	4	2
Ubicomm 2008	2008	7	0	4	3	0
EGC 2008	2008	18	2	6	7	3
Total / %		271	33,95 %	32,84 %	16,61 %	16,61 %

Several elements can emerge from the analysis of this information (Figure 3 and Table 1). The most frequently cited article is the one published in the journal COMIND, which deals with mechanisms used to support group awareness. Although this is not technically part of these contributions, it provided the basis on which these contributions were built. One can also observe the important number of self-citations for the CRIWG 2004 article. This can be explained by the founding aspect of this article in relation to later work. It introduces the proposed context model, thus constituting the reference point for the continuation of this work.

It is also worth noting a change in the community focused by these publications during this period. These contributions have targeted both communities, Pervasive Computing as well as CSCW (Computer Support for Cooperative Work) community. The first works (COMIND 2003, CRIWG 2004, CSCWD 2004) have particularly targeted CSCW community, while the publications that followed (MATA 2004, SAC 2005, CAISE 06 Workshop, Ubicomm 2008) have mainly targeted Pervasive Computing community, which is also target by the contributions discussed in the next chapter.

Besides, the period in which this work had more influence (quantified by the number of citations) is the period up to 2013, corresponding to a period in which researches on context-aware systems were also numerous.

Finally, it is also important to underline the impact of this work on later contributions: the operations used in the proposed filtering mechanism have influenced my researches on services selection, as well as the context model has influenced my work on Information Systems, to which we may add the collaborations carried out since 2008 [21, 76] made possible thanks to this work.

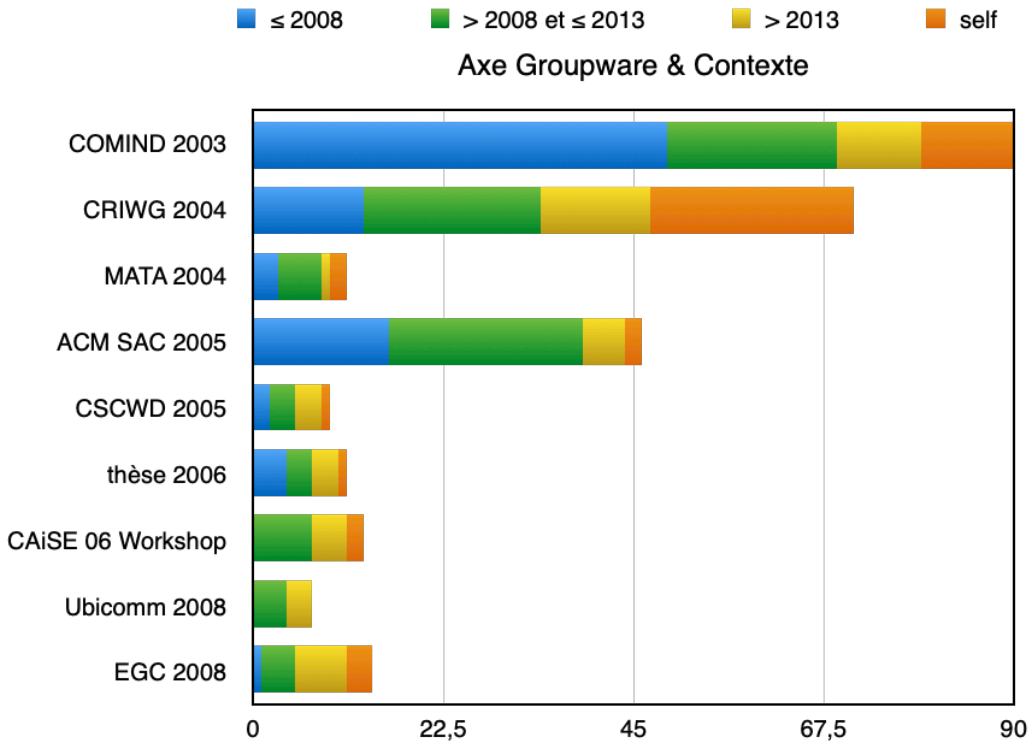


Figure 3. Evolution of citations over time.

As discussed above, among all papers resulting from this research work, the one published at CRIWG 2004 appears as the most relevant one, defining the basis of this work. It was then chosen as the most illustrative one, being consequently attached to this document in the Annex II.

2 Context on Pervasive Environments

Research works presented in this chapter have target particularly the community of Pervasive Computing, which can be defined as the transparent integration of IT and its devices into our daily life [6]. Also known as Ubiquitous Computing, this term represents, according to Moran & Dourish [104], a trend towards environments enriched by numerous computing devices, often mobile or embedded in the environment, connected by fixed or wireless networks. Originally proposed by Weiser [171], this vision of computing that has become invisible to our eyes is gradually becoming a reality, since, as Bell & Dourish [10] point out, we continually use computing resources without necessarily perceiving them as computers.

The challenges raised in this research domain are multiple and quite often related to the heterogeneity and dynamic nature of these so-called pervasive environments. Indeed, these environments are characterized by their heterogeneity, including devices as varied as network equipment (router, switch, etc.), "traditional" personal computers (fixed or portable), smartphones and tablets, and even devices used for IoT (*e.g.* RaspberryPI, Arduino). These environments are also often very dynamic, integrating devices that can easily join the network or leave it, by a simple disconnection or being switched off for different reasons. New devices can easily appear in the perimeter, while others can disappear depending on their use, on their mobility (or that of their owners), on their battery or power level, and so on. The notion of context becomes thus a key element here, promising to better take into consideration these environments. Observing the context in which these resources are executed represents a particularly relevant information for applications and platforms that desire to take advantage of these environments and of all the possibilities of interaction, storage and even computing capacities that can be found in these environments.

In this chapter, we are particularly interested in these pervasive environments, focusing on two distinct problems: firstly, we focus on how to make context information available in these environments (Section 2.1); secondly, how, based on the context information, could we rationalize the use of computing resources in these environments, particularly for the execution of "Big Data" applications (Section 2.2).

2.1 Context grouping

2.1.1 Problem statement

Context-aware applications are often presented as distributed applications. They may entail different nodes that can cooperate to achieve application goals or simply to provide a better user experience. Different scenarios can illustrate this trend, such as, for example, scenarios involving multi-scale systems [139], the adaptation of an application to the available resources (hardware and software) [34], or the possibility to propose an opportunistic composition of services to the user [35]. In any case, the distributed nature of these applications and services may also require the distribution of context information across the different nodes present in the environment.

Indeed, in order to take full advantage of the pervasive environment that surrounds them, some applications must rely on the exchange of information concerning the surrounding resources and their execution context. For example, a communication application may take advantage of the presence of a nearby node with a better display or a better connection capacity and then try to use these resources by deploying certain tasks on neighboring nodes. This scenario, considered for example in [33], requires that neighboring nodes share their execution context. Similarly, scenarios involving sharing

resources in a smart building, as considered by [49], would also require sharing context information concerning available resources in order to exploit the full potential offered by these environments.

However, since pervasive environments are characterized by their dynamism, sharing context information about the resources that are available in the environment raises different challenges. First of all, context information concerning these resources will evolve over the time. The composition of the environment itself will also change, with new resources entering or leaving the environment or simply becoming unavailable. In addition, not all applications are necessarily interested in all the context information that can be available, without mentioning the obvious security and privacy issues that raise when considering sharing context information among available nodes.

Thus, in this contribution, carried out essentially between 2008 and 2009, partially within the European project named IST-MUSIC, we focused on the question of how to share context information in a pervasive environment. Without going into security and privacy issues, this research work focuses on the dynamism of context information itself and on the dynamism of the environment. Context information must be updated on a regular basis, according to changes observed in the surrounding environment, whose composition is fluctuating. We propose in this research work a peer-to-peer mechanism for distributing context information in a dynamic environment whose composition and nature varies over time.

In the proposed context distribution mechanism, the nodes that are available in the environment are dynamically organized into groups, according to a common context shared by the nodes participating of each group. The use of context information is twofold: it is used to organize the groups and is also shared among the nodes belonging to these groups. The nodes, which represent the resources available in the environment, are organized into groups according to a common context defined, at the application level, by a criteria set. For example, a group can be defined for resources that are co-located (*e.g.* located in the same room), have the same network connection, or belong to users sharing the same role in the organization. Within these groups, context information concerning the nodes is distributed among the other group members, who can thus become aware of the current situation of the other resources belonging to the same group. For example, in a group defined on a common location (*i.e.* co-located resources), it is possible to share information about available memory and display capabilities among the members. Such an information about neighborhood nodes would be useful, for example, in a communication application such as the one considered in [33].

This context distribution mechanism was originally published at DOA 2008 [86] conference and has, after that, been improved using notably the Formal Concept Analysis [131, 173], a data analysis technique, for discovering possible groups in the environment. The original paper can be consulted in the Annex III.

- **DOA 2008 [86]** : Kirsch-Pinheiro, M.; Vanrompay, Y.; Victor, K.; Berbers, Y.; Valla, M.; Fra, C.; Mamelli, A.; Barone, P.; Hu, X.; Devlic, A.; Panagiotou, G., “Context Grouping Mechanism for Context Distribution in Ubiquitous Environments”, In: Robert Meersman, Zahir Tari et al.(eds.), *10th International Symposium on Distributed Objects, Middleware, and Applications (DOA'08), OTM 2008 Conferences, Lecture Notes in Computer Science*, 5331, **2008**, 571-588.

2.1.2 Bibliometric

The research work described in this chapter was carried out between 2008 and 2009, with the publication of a paper at the DOA (Distributed Objects, Middleware and Applications) conference [86]. After, on 2012, this research work has improved, resulting in a new publication in 2013 [164]. Additionally, this research work has allowed the development of other collaborations concerning context distribution topic, in particular within the IST-MUSIC project. Each of these collaborations resulted in a publication. A first collaboration has concerned the definition of a P2P architecture

allowing context distribution on the IST-MUSIC project [63], and a second one has tacked the privacy issues on context distribution thanks to the use of user-specific policies (as opposed to application-specific policies, as in our case) [36]. All the publications related to context distribution is then summarized in the list below.

- **DOA 2008 [86]**: Kirsch-Pinheiro, M.; Vanrompay, Y.; Victor, K.; Berbers, Y.; Valla, M.; Frà, C.; Mamelli, A.; Barone, P.; Hu, X.; Devlic, A.; Panagiotou, G., "Context Grouping Mechanism for Context Distribution in Ubiquitous Environments", In: Robert Meersman, Zahir Tari et al.(eds.), *10th International Symposium on Distributed Objects, Middleware, and Applications (DOA'08), OTM 2008 Conferences, Lecture Notes in Computer Science*, 5331, **2008**, 571-588.
- **ChapCtxGrp 2013 [164]**: Vanrompay, Y.; Kirsch Pinheiro, M.; Ben Mustapha, N.; Aufaure, M.-A., "Context-Based Grouping and Recommendation in MANETs", In : Kolomvatsos, K., Anagnostopoulos, C., Hadjiefthymiades, S. (Eds.), *Intelligent Technologies and Techniques for Pervasive Computing*, IGI Global, **2013**, 157-178.
- **ISD 2008 [63]**: Hu, X.; Ding, Y.; Paspallis, N.; Bratskas, P.; Papadopoulos, G.A.; Vanrompay, Y.; Kirsch Pinheiro, M.; Berbers, Y., "A Hybrid Peer-to-Peer Solution for Context Distribution in Mobile and Ubiquitous Environments", In: Papadopoulos G., Wojtkowski W., Wojtkowski G., Wrycza S., Zupancic J. (eds), *17th International Conference on Information Systems Development (ISD2008), Information Systems Development: Towards a Service Provision Society*, **2008**, Springer, 501-510. DOI : 10.1007/b137171_52
- **CoMoRea 2009 [36]**: Devlic, A.; Reichle, R.; Wagner, M.; Kirsch Pinheiro, M.; Vanrompay, Y.; Berbers, Y.; Valla, M., "Context inference of users' social relationships and distributed policy management", *6th IEEE Workshop on Context Modeling and Reasoning (CoMoRea), 7th IEEE International Conference on Pervasive Computing and Communication (PerCom'09)*, Galveston, Texas, 13 March **2009**. DOI : 10.1109/PERCOM.2009.4912890

As for the previous chapter, these publications were analyzed in terms of number of citations. We used various sources for this, including scholar.google.com, www.researchgate.net and hal.archives-ouvertes.fr, as well as the publisher's site, when available. These citations were then organized into three categories: before 2013, which corresponds, approximately, to the first 5 years after the first publication; between 2013 and 2016; and after 2016. Self-citations were also accounted for and distinguished from other citations. Table 2 details the data obtained, illustrated in Figure 3.

Table 2. Citations concerning context grouping publications.

Reference	Year	Total	≤ 2013	> 2013 & ≤ 2016	> 2016	Self-citation
DOA 2008	2008	13	7	2		4
ChapCtxGrp 2013	2013	6	1			3
ISD 2008	2008	6	5			1
CoMoRea 2009	2009	22	12	6	3	
Total / %		47	53,19 %	17,02 %	6,38 %	23,4 %

As one might expect, citations for these articles are mainly concentrated on the first 5 years following their publication (between 2008 and 2013). This corresponds to a very intense period for research on context-aware computing. This work was therefore adopted by other authors from this community during this period, but, since it was not continued, this work has gradually ceded its place to more recent approaches on context distribution.

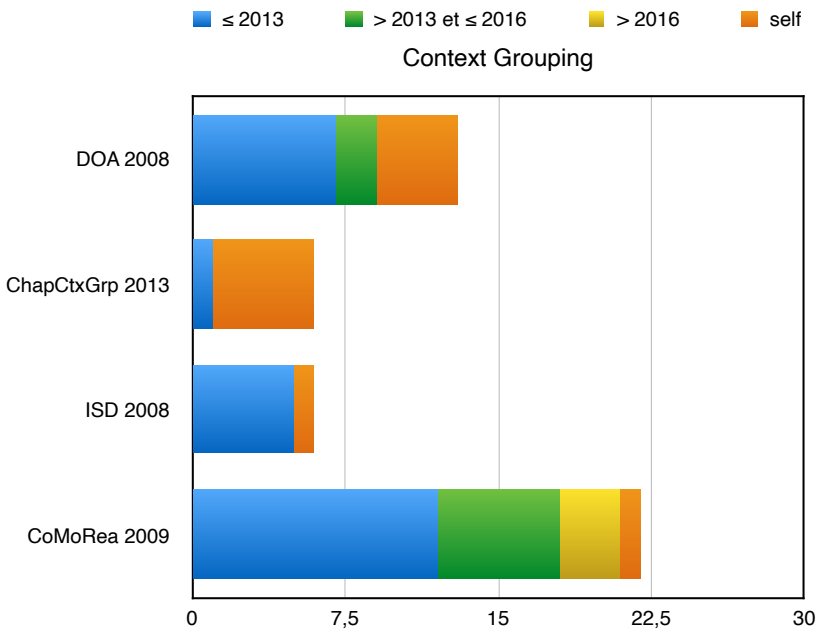


Figure 4. Citations evolution over different periods of time.

2.2 PER-MARE project

2.2.1 Problem statement

Unlike what one might expect, pervasive environments can offer computing capacities distributed among all the units (nodes) available on it. These different units can, in theory, collaborate for collecting and processing data from sensors in order to autonomously perform certain tasks. This scenario, which is envisaged, among others, by Ausiello [5], illustrates the interest of using resources integrated in pervasive environments for the execution of certain computing tasks, and particularly in the case of Big Data applications.

However, the heterogeneity that characterize pervasive environments represents an important challenge when considering using resources on these environments for task execution. With resources than can be very varied, even diametrically opposed, ranging from high-performance servers (HPC) to nanocomputers (Raspberry PI, Arduino, etc.), the use of such resources for computational purposes represents a challenge, notably concerning the placement of computational tasks on these very heterogeneous resources. As pointed out by Breitbach *et al.* [16], the placement of computing tasks in a heterogeneous environment is more complex than the one performed on a Cloud environment or on computing grids. Indeed, it is particularly difficult to distribute computing tasks and to guarantee their execution on resources that do not behave in the same way and whose performances can be very varied and variable. This heterogeneity can have a significant impact on the execution of these tasks and on their performance. The dynamism of these environments, with particularly volatile nodes that can disappear or join the network during the execution, will also have an impact on performance. Under these conditions, it is difficult to anticipate the execution performance of a set of tasks in such an environment. This is particularly true for Big Data type applications, which must combine their needs of computing capabilities with the management and the transfer of large amounts of data.

The PER-MARE project was born from this observation. Running from 2013 to 2014, this project is an international cooperation CAPES/MAEEA/ANII STIC-AmSud (project number 13STIC07) involving the University of Reims Champagne-Ardennes and the University Paris 1 Panthéon Sorbonne in France, as well as the Universidade Federal de Santa Maria (UFSM) in Brazil and the Universidad de la República in Uruguay. The main objective of the PER-MARE project was to provide a support for the execution of MapReduce applications in a pervasive environment. MapReduce is a programming model for data-intensive applications in which the processing is organized in two phases: the map, in which the data is divided into several blocks and processed to form a set of "key, value" peers; and the reduce, in which the results of the first phase are aggregated to produce a final result [172]. The data divided into several blocks are thus distributed among the cluster nodes and processed in the map phase, resulting in a set of "<key, value >" peers, which are then grouped again into several blocks according to the key values found, to be finally consolidated in the reduce phase (see Figure 5). Because of its easily distributed nature, the MapReduce principle has been widely used on Big Data platforms, including Apache Hadoop, which was the main Big Data platform at the time the PER-MARE project started.

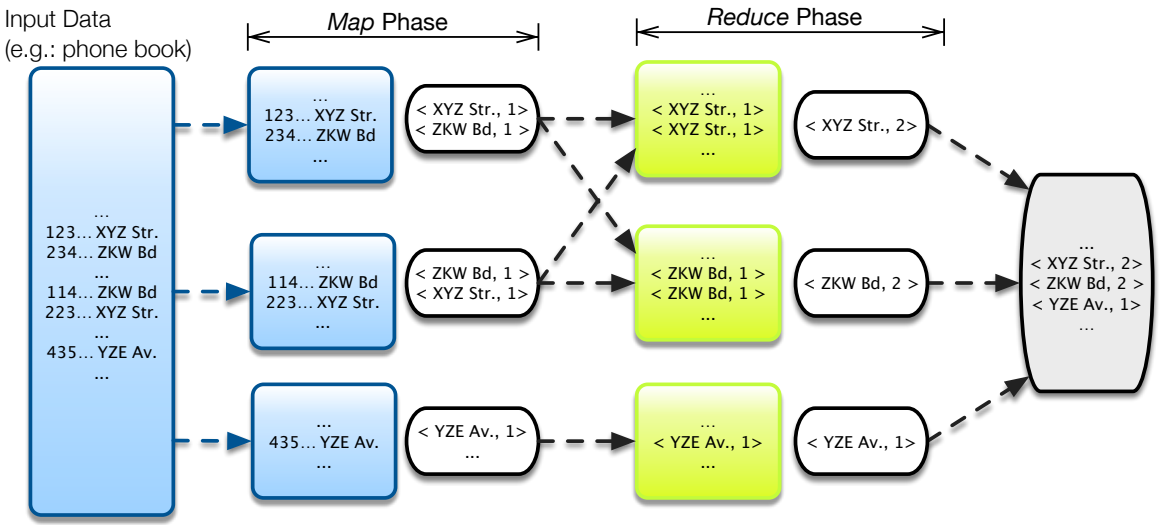


Figure 5. illustration of the MapReduce programming model considering a telephone directory, in which the number of telephones ("123…") by addresses ("XYZ Str…") is counted.

The execution of MapReduce applications in a pervasive environment requires a rationalized use of the resources present in these environments, in order to obtain a good distribution of data and processing tasks according to the capacities of the available devices. In the PER-MARE project, we have considered the use of pervasive grids [126], whose goal is to dynamically and opportunistically build computing grids from nearby available resources. According to Parshar & Pierson [126], pervasive grids represent the extreme generalization of the notion of computing grid, in which resources are pervasive. These grids can thus integrate both sensor or actuator-type devices and conventional high-performance terminals. Desktop grids, grids composed of desktop PCs made available by voluntary users, would thus be a particular case of pervasive grids, very heterogeneous by definition [153].

Pervasive grids can be assimilated to Fog or Edge Computing platforms [29, 122]. Fog / Edge Computing can be seen as a new trend complementary to Cloud Computing. It is an alternative to the "all-cloud" model in which all processing is done exclusively in the cloud. The aim would then be to use of nearby devices to carry out some processing tasks (data aggregation, pre-processing, anonymization, simple calculations, etc.), instead of systematically transferring all data processing for remote processing on cloud platforms or datacenters. This model is particularly interesting for minimizing problems related to network latency or to the transfer of large amounts of data to the cloud, as well as to problems

related to data privacy or security [60, 143], since data remain as close as possible to its production environment. By minimizing or even avoiding data transfers to remote platforms and taking advantage of nearby resources, this model not only reduces these problems but also makes better use of available (and often underutilized) resources at the "edge" of the network or close to the user.

Thus, the main goal of the PER-MARE project was to explore the use of heterogeneous resources for Big Data processing. Two complementary approaches have been explored during this project: (i) to consider the execution context within the Hadoop platform; and (ii) the use of available resources within a Fog/Edge computing platform. In both cases, the problem remains the use of heterogeneous resources for Big Data in an opportunistic way. Each approach has given rise to its own contributions.

In the first approach, the Apache Hadoop has been modified in order to support heterogeneous clusters. Indeed, Apache Hadoop [172] was especially designed for the execution and the deployment of MapReduce applications on homogeneous computing clusters, *i.e.* homogeneous computing environments with a stable number of available homogeneous resources. The platform was not originally designed to run on heterogeneous and dynamic environments (*i.e.* environments whose composition and state may vary over time), such as pervasive grids. According to Hagras [56], the dynamic and ad-hoc nature of pervasive environments requires adapting to changing operating conditions and variations in user's preferences and behavior in order to promote more efficient and effective operation, while preventing system failures. Unfortunately, the Hadoop platform, in its original configuration, was unable to dynamically adapt its operation to a variable environment, in which resources are heterogeneous in their nature and can enter or exit the environment at any time. These variations lead to a degraded performance of the Hadoop platform is such environments.

In order to overcome this problem, we have proposed, within the PER-MARE project, to adapt the Hadoop platform in order to consider such heterogeneous environments. We have modified the *ResourceManager* and the *NodeManager*, elements in charge of the resource management in Hadoop, in such way they could capture context information of the executing nodes (number of available cores, available memory, etc.), and use this information instead of the static configuration of the node. To do this, we have integrated a *ContextCollector* into the *NodeManager*. This *ContextCollector* has been inspired by our previous work on context modeling [89] (see Chapter 1), as well as those carried out in the framework of the IST-MUSIC project [170]. It combines an object-oriented approach with an ontology approach, allowing a semantic description of context elements, while keeping the advantages of an object-oriented implementation, and mainly the lightness of this implementation, which is particularly important for high-performance applications.

This approach was the subject of different publications (see section 2.2.2), among those a journal paper published at JAIHC in 2016 [24], which presents latest results obtained by the project. This paper is available at the Annex IV.

- **JAIHC 2016** [24] : Cassales, G.W.; Charão, A.S.; Kirsch-Pinheiro, M.; Souveyet, C. & Steffenel, L.-A. "Improving the performance of Apache Hadoop on pervasive environments through context-aware scheduling", *Journal of Ambient Intelligence and Humanized Computing*, 7(3), **2016**, 333-345.

The second approach adopted by the PER-MARE project consists in proposing an alternative to Hadoop clusters through an independent Fog Computing platform. The CloudFIT platform [151], developed by the University of Reims Champagne Ardennes, was proposed in this sense. This platform aims at using the resources available on the environment for the execution of Big Data applications, without having to use a heavy platform such as the Hadoop platform. CloudFIT platform considers heterogeneous and volatile resources, such as the resources of a pervasive grid. According to Coronato & De Pietro [31], pervasive grids must be able to self-adapt and self-configure in order to accommodate mobile devices.

In these environments, the challenge is thus not limited to the heterogeneity of the available devices, but also to the volatility of these devices. As these devices are not dedicated to the computing tasks, they can easily connect and disconnect from the network according to their movements (or those of their owners), the availability of the network, or the state of their power supply.

The CloudFIT platform [151] was designed during the PER-MARE project to allow the execution of Java applications, including Big Data applications, targeted by the project, using resources available at the edge of the network. For this, the platform relies on a peer-to-peer (P2P) network in which participating devices (also called nodes) share tasks and data. Managing the volatility of these devices is done through both the use of the P2P network (and more precisely a P2P network overlay), and through the distributed nature of the task management, in which tasks are shared among the available nodes. Each new node wishing to participate in the computing effort can then request to join the community. It can make this request to any other node belonging to the community. As soon as it joins it, it receives from the other nodes the list of tasks to be performed and becomes a candidate to store data (data it produces itself or replicas of data present on the other nodes). Each node decides by its own which task it will execute according its own execution context and the availability of the data necessary for the task execution.

During the development of the CloudFIT platform, various experiments have been performed, whose results have been published in several articles, including CLIoT 2015 [153], which first details the CloudFIT platform, and which is available at the Annex V:

- **CLIoT 2015** [151] : Steffenel, L. & Kirsch-Pinheiro, M. "CloudFIT, a PaaS platform for IoT applications over Pervasive Networks", In: Celesti A., Leitner P. (eds). *3rd Workshop on CLoud for IoT (CLIoT 2015)*. Advances in Service-Oriented and Cloud Computing (ESOCC 2015). Communications in Computer and Information Science, vol. 567, **2015**, 20-32.

2.2.2 Bibliometric

The research works on the PER-MARE project have resulted in several publications, from 2013 to 2016, going well beyond the official duration of the project (2013-2014). In this section, we try to analyze the impact of these publications, listed below, in terms of citations. As in the previous chapter, we have considered the citations visible from the scholar.google.fr platform, which condenses information from the many other sources (IEEExplore, SpringerLink, ScienceDirect, Arxiv, etc.), but also from the www.researchgate.net and www.semanticscholar.org platforms, as well as from the publisher's website where applicable. The citations are organized in two categories: those up to 2017 (dating from the first 5 years from the beginning of the project); and those from 2018 onwards. To these categories are added the self-citations, which have been accounted for separately. Table 9 presents the figures obtained from this analysis, while Figure 6 illustrates the proportion of these citations.

- **3PGCIC 2013** [153]**:** Steffenel, L. A.; Flauzac, O.; Charao, A. S.; Barcelos, P. P.; Stein, B.; Nesmachnow, S.; Kirsch Pinheiro, M. & Diaz, D., "PER-MARE: Adaptive Deployment of MapReduce over Pervasive Grids", *8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC'13)*, **2013**, 17-24.
- **UBICOMM 2014** [23]: Cassales, G.W.; Charão, A.S.; Kirsch-Pinheiro, M.; Souveyet, C. & Steffenel, L.A. « Bringing Context to Apache Hadoop », In: Jaime Lloret Mauri, Christoph Steup, Sönke Knoch (Eds.),*8th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2014)*, August 24 - 28, **2014,** Rome, Italy, ISBN: 978-1-61208-353-7, IARIA, 252-258.

- **JCS 2014** [152]: Steffenel, L. A., Flauzac, O., Charao, A. S., P. Barcelos, P., Stein, B., Cassales, G., Nesmachnow, S., Rey, J., Cogorno, M., Kirsch-Pinheiro, M. & Souveyet, C., "Mapreduce challenges on pervasive grids", *Journal of Computer Science*, 10(11), July **2014**, 2194-2210.
- **CLIoT 2015** [151]: Steffenel, L. & Kirsch-Pinheiro, M. "CloudFIT, a PaaS platform for IoT applications over Pervasive Networks", In: Celesti A., Leitner P. (eds). *3rd Workshop on CLoud for IoT (CLIoT 2015)*. Advances in Service-Oriented and Cloud Computing (ESOCC 2015). Communications in Computer and Information Science, vol 567, **2015**, 20-32.
- **CN4IoT 2015** [154]: Steffenel, L.A. & Kirsch Pinheiro, M. "When the cloud goes pervasive: approaches for IoT PaaS on a mobiquitous world". In: Mandler B. et al. (eds), *EAI International Conference on Cloud, Networking for IoT systems (CN4IoT 2015), Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST)*, 169, **2015**, 347–356.
- **JAIHC 2016** [24]: Cassales, G.W.; Charão, A.S.; Kirsch-Pinheiro, M.; Souveyet, C. & Steffenel, L.-A. "Improving the performance of Apache Hadoop on pervasive environments through context-aware scheduling", *Journal of Ambient Intelligence and Humanized Computing*, 7(3), **2016**, 333-345.
- **Big2DM 2015** [155]: Steffenel, L.A. & Kirsch-Pinheiro, M., "Leveraging Data Intensive Applications on a Pervasive Computing Platform: the case of MapReduce", *1st Workshop on Big Data and Data Mining Challenges on IoT and Pervasive (Big2DM)* , London, UK, June 2 - 5, 2015. *Procedia Computer Science*, vol. 52, Jun 2015, Elsevier, 1034–1039. doi: 10.1016/j.procs.2015.05.102.
- **ANT 2015** [22]: Cassales, G.W., Charao, A., Kirsch-Pinheiro, M., Souveyet, C. & Steffenel, L.A., "Context-Aware Scheduling for Apache Hadoop over Pervasive Environments", *The 6th International Conference on Ambient Systems, Networks and Technologies (ANT 2015)*, London, UK, June 2 - 5, 2015. *Procedia Computer Science*, vol. 52, Jun **2015**, Elsevier, 202–209. doi: 10.1016/j.procs.2015.05.058.

In addition to these articles, which are directly related to the themes addressed by the PER-MARE project, the project has also enabled other collaborations between the project members around the topic of the use of heterogeneous resources. These collaborations, which can be called satellites, have also given rise to some publications, listed below:

- **ANT 2014** [45]: Engel, T.A., Charao, A., Kirsch-Pinheiro, M., Steffenel, L.A. "Performance Improvement of Data Mining in Weka through GPU Acceleration", *5th International Conference on Ambient Systems, Networks and Technologies (ANT 2014)*, Hasselt, Belgium, June 2 - 5, 2014. *Procedia Computer Science*, vol. 32, **2014**, Elsevier, pp. 93–100.
- **JAIHC 2015** [44]: Engel, T.A., Charao, A., Kirsch-Pinheiro, M., Steffenel, L.A. "Performance Improvement of Data Mining in Weka through Multi-core and GPU Acceleration: opportunities and pitfalls", *Journal of Ambient Intelligence and Humanized Computing*, Springer, June **2015**. doi:10.1007/s12652-015-0292-9.

As the PER-MARE project has been built up over the years, the number of self-citations on certain publications is naturally high, notably on those establishing the working bases [152, 153] and detailing the CloudFIT platform [154]. Nevertheless, it worth noting that the number of citations for some works (notably [23, 24, 44]) has been increasing since 2017. This is mainly due to a certain democratization of Big Data and data analysis applications (subject treated by the last two publications).

Finally, there is a deliberate willingness in the PER-MARE project to give priority to publications on an "open access" mode. Among the publications presented above, half are open access, being both free and peer-reviewed. These contribute to 48.83% of the citations (excluding self-citations).

Table 3. Synthesis of citations concerning PER-MARE project papers, organized by publication year.

Reference	Year	Total	≤ 2017	> 2017	Self-citations
3PGCIC 2013	2013	11	4	0	7
JCS 2014	2014	7	2	1	4
Ubicomm 2014	2014	3	0	1	2
CN4IoT 2015	2015	7	0	2	7
CLIoT 2015	2015	6	3	1	2
Big2DM 2015	2015	4	1	1	2
ANT 2015	2015	9	4	3	2
JAIHC 2016	2016	9	1	6	2
ANT 2014	2014	9	2	6	1
JAIHC 2015	2015	6	1	4	1
Total / %		**73**	**24,66 %**	**34,25 %**	**41,10 %**

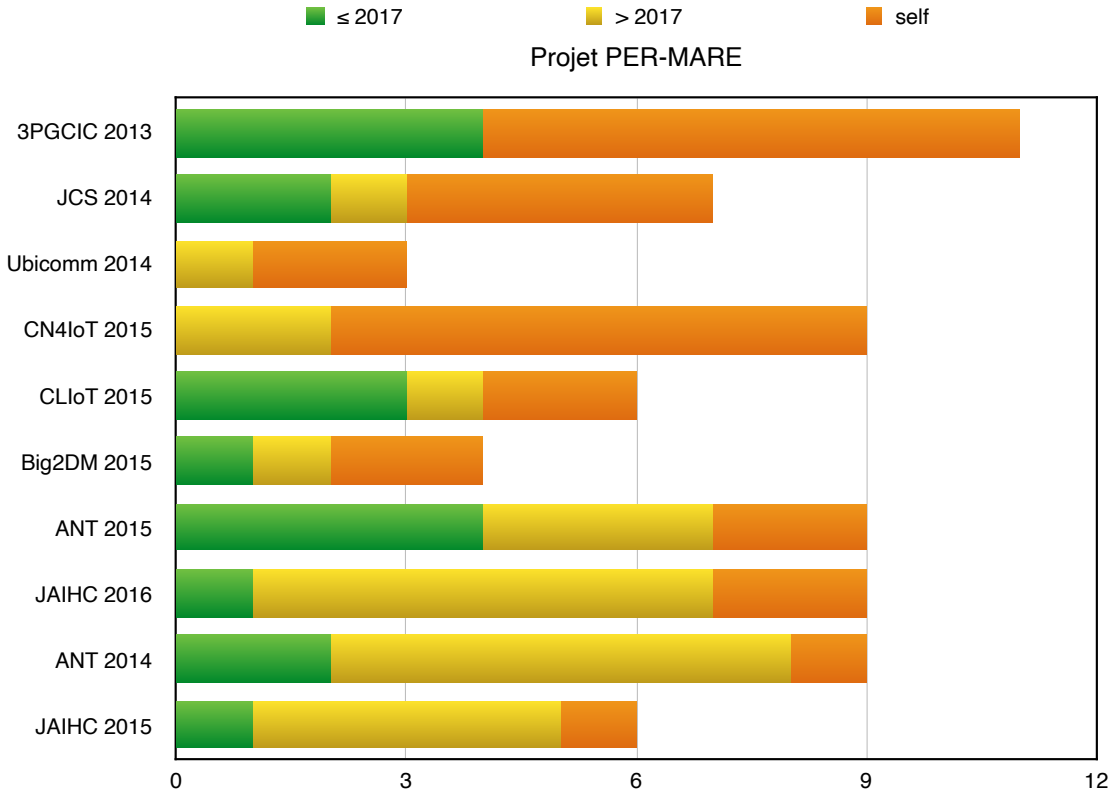


Figure 6. Illustration of the evolution of PER-MARE citations over the time.

2.3 Chapter Summary

In this chapter, we have discussed research works taking place from 2008 to 2010 (context grouping) and from 2013 to 2015 (PER-MARE project) and considering two distinct projects: IST-MUSIC for the first (cf. section 2.1), and PER-MARE for the second (cf. section 2.2).

The main concern of these works is the pervasive environments. Contrary to the work presented in Chapter 1, in which the human element (the user) was at the center of the analysis, here the focus was on these environments. Particular attention was given to the heterogeneity and the dynamicity of such environments. These characteristics raise several questions, including the opportunistic use of available resources for computational tasks, for which considering the context information during execution can represent a determining factor, illustrating hence the relevance of work such as ours on the distribution of context information within these environments.

Considering the pervasive environments and the resources these environments may offer represents a growing trend with the development of Fog/Edge Computing platforms and other related concepts [16, 28, 43, 64, 122, 168, 169]. All these concepts share the same vision: they aim at using resources with a certain computing capacity that would be available (or potentially under-exploited) in the environment around the user or the data. This proximity computing model is particularly interesting when associated with the IoT. The data and/or the treatments from/related to the IoT can thus be handled by devices close to the objects at the origin of these data or treatments. This would allow, for example, to significantly reduce the volume of data transmitted over the network, which represents a significant advantage in a Big Data context, or to reduce the reaction time following a decision.

The PER-MARE project (2013-2014) has focused on Big Data application, which represented, at the time of the project proposal, a rapidly growing field. The project's central proposal of using pervasive grids to execute these applications was very innovative at this moment, since at this specific period the use of dedicated platforms (cluster or cloud) was still largely dominant, as illustrated by the 2012 Gartner's Hype Cycle shown in Figure 7. It is important to note that the terms Fog/Edge Computing were still mostly unknown. These terms appeared around 2012 [14, 29, 48, 122] and have gained more attention more recently, as a complement to Cloud Computing, especially in the context of IoT and data analysis applications.

The current development of Fog Computing [28, 43, 66, 122, 161, 168] highlights the relevance of the PER-MARE project's proposals, especially as they remain innovative compared with the literature. Indeed, the predominant vision remains of fog platforms as an intermediary stage before data is sent to Cloud platforms. This dependency is underlined by Alrawais *et al.* [3], for whom the goal of Fog Computing is to reduce the volume and the traffic of data to cloud servers, thus reducing latency and increasing quality of service. In the PER-MARE project, this constraint was removed, since we did not consider the constant presence (or rather availability) of a dedicated platform, whether a cluster or a cloud, for running applications at one time or another. The objective has always been an opportunistic use of whatever resources are available. By freeing ourselves from this constraint, we have been able to consider the use of any resource at hand, be it resources from the IoT (such as the RaspberryPIs used in our experiments), laptops or even clusters or resources on the cloud.

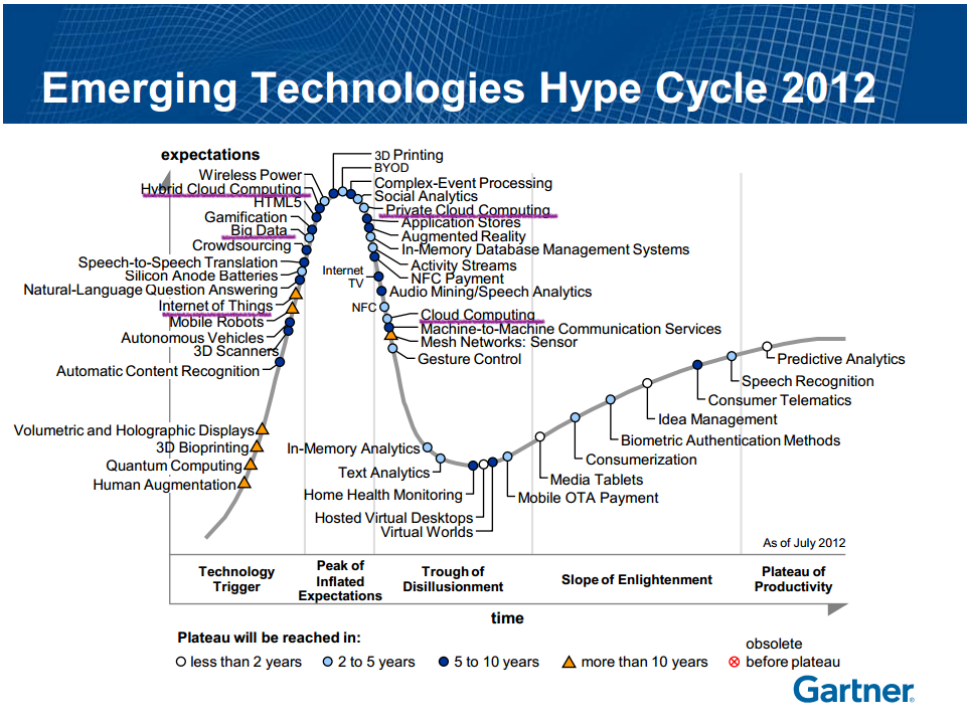


Figure 7. Gartner's emerging technology hype cycle de technologies for 2012[1].

Besides, today's availability of an unprecedented mass of data, including data from the IoT, combined with the available computing offer (whether on Cloud or Fog Computing platforms) opens up new application perspectives, especially in terms of data analysis. We are talking here about what some people call "Edge AI" or the use of Fog Computing platforms, in association with Cloud platforms, for the application of Artificial Intelligence techniques, and in particular Deep Learning, on data mostly from the IoT. The entry of this term into the 2019 Gartner Hype Cycle (see Figure 8) illustrates the industry's growing interest for this kind of solution. The PER-MARE project can be seen as a precursor, proposing since 2013 the use of local resources for processing data, whether or not it comes from the IoT. The "Edge AI" represents a new field of application for resources already available in an Information Systems, which could benefit a wide range of sectors: Industry 4.0, logistics, but also Human Resources (HRIS), financial, and so on. Whatever the field of application is envisaged, in order to become a reality these scenarios ought to observe the execution context. Context information must be taken into account appropriately, so that the heterogeneity of these environments could be considered. Context-awareness becomes then essential, as highlighted by the results we obtained within the PER-MARE project, and consequently context distribution mechanisms, as the one presented in section 2.1, will become more and more necessary in order to allow the growing development of such applications.

[1] Source : https://res.infoq.com/news/2012/08/Gartner-Hype-Cycle-2012/fr/resources/hype1.png

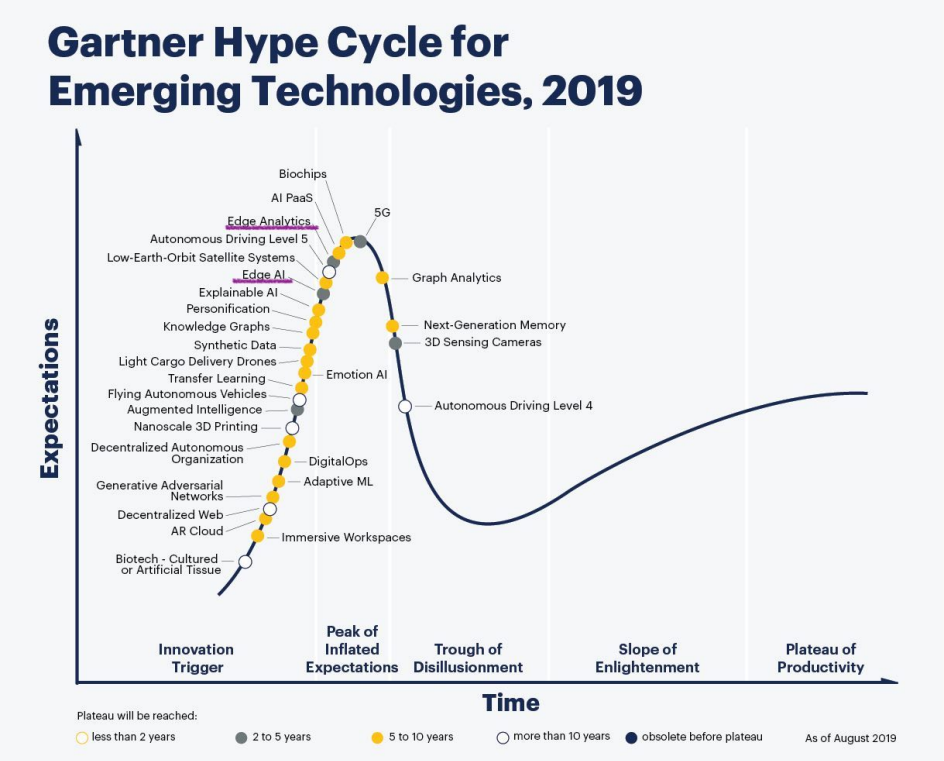


Figure 8. Gartner's emerging technologies hype cycle for 2019[2].

[2] Source : https://www.gartner.com/smarterwithgartner/5-trends-appear-on-the-gartner-hype-cycle-for-emerging-technologies-2019/

3 Context on the Service Orientation

Among the questions raised by Pervasive Computing, we may cite the context management, but also the management of the interoperability, considering the heterogeneity that characterize pervasive environments. In this context, service orientation has emerged as a viable approach for dealing with these issues, providing interesting solutions to the problems at hand, but also benefiting from context awareness. Service Oriented Computing (SOC) can be seen as a computing paradigm relying on the notion of service as a basic unit for faster, more reliable and cheaper design and development of distributed applications over heterogeneous environments [123,124,125]. A service can be defined as an independent entity with well-defined interfaces that can be invoked in a standard way, without requiring any knowledge from the customer about how the service actually performs its tasks [67]. They are self-described software elements, independent of the platform and accessible through a standard interface [2]. Customers and service providers are thus independent, communicating only through the interface of these services. Services can thus be exposed, published, discovered, composed and negotiated at the request of a customer and invoked by other applications [125].

Service orientation can thus be characterized by its loose coupling, technological independence and scalability [67]. It is this loose coupling that makes the notion of service particularly attractive for pervasive environments, since these environments are characterized by the volatility of their elements [166]. Indeed, service orientation, through its loose coupling and the self-describing nature of services, allows, on the one hand, to better manage the volatility of pervasive environments and to better isolate applications from the different technologies involved. On the other hand, by observing context information made possible the emergence of services that can better adapt their own behavior to their execution context and therefore to these highly dynamic environments.

Since 2007, the notion of services in the field of Pervasive Computing has been an important part of my research. Different research issues have been treated over the years: (i) the identification of relevant context elements for a service (or an application); (ii) the selection of the most appropriate service (or implementation) according to the execution context; (iii) the integration of the user and her/his business goals in the definition of these services; and (iv) the forecasting of user's goals for a more proactive behavior. These different issues have led to several contributions presented in the following sections. The common element of these proposals, beyond the notion of context and the main community in which they were made (Pervasive Computing), is the notion of service itself: in each of these contributions, we find the vision of a system as a set of services proposed to the user, and whose adaptation according to the context proves to be necessary. The following sections summarize the proposed contributions and their impact.

3.1 Context mining

3.1.1 Problem statement

A particularly important challenge for the development of context-aware applications is the identification of the context elements that will be observed by the application [7,37,53,54]. Indeed, since context information is a key element for controlling the behavior of these applications, the identification of the relevant elements for these applications becomes a crucial task in their development, forcing the designers of these applications to anticipate their combinations and their relevant characteristics before implementation [7]. The same applies to context-aware services, whose behavior must be adapted to the execution context.

Even if this delicate question remains, until now, without a satisfactory answer, it raises a second related question: would it be possible to characterize the use of a service (or an application) by observing its execution context? In other words, by observing the context in which a service is executed, would it be possible to determine recurring context elements capable of characterizing the use of this service (or application) by an ordinary user?

This research work on “context mining” derives from this question. This work aims at using data mining techniques on context data, in order to identify relevant context elements that could characterize the use of a service (or a simple application).

In this work, carried out mainly between 2012 and 2014, first during Ali Jaffal’s master thesis [72], then during the beginning of his PhD thesis [71], a methodology has been proposed allowing to associate the use of a service to characteristic situations, recognized thanks to the observed context elements. This methodology allows identifying, from data concerning application usage, context elements that characterize the use of a given application. Through the applications, it is the notion of service that is focused. The objective is to recognize the influence of one (or several) context element(s) in the choice of a service (represented here by an application) by a user. In order to do this, this methodology proposes the application of Formal Concepts Analysis (FCA) [131, 173], which allows services to be organized into several clusters according to the context elements observed during their use.

The choice of the FCA is a particularly innovative element in this work. Most works on context mining [133, 147] use statistical methods, such as Bayesian Networks [133] or Markov chains [102, 175]. The use of the FCA was then quite original at this time (2012-2014). Moreover, most of these works [133, 147, 102] use data mining techniques for the recognition of the user’s activities (*e.g.* running, walking, sleeping, etc.), and not for the analysis of the relevance of a context element as here. In this case, we may cite [132, 133] as a similar approach, since these authors seek to learn, from the data, associations between contextual elements and the actions to be taken by the system (the actions taking the place of services here). Besides, the adoption of the FCA brings an interesting feature to this work: a “multi-class” classification. Indeed, unlike many statistical clustering methods, the FCA allows the overlapping of the different “classes” identified from the data. In other words, the same service can be simultaneously placed in different formal concepts (which represent the “classes”), thus identifying several situations, recognized thanks to a set of context elements, in which a given service has been invoked.

However, the most interesting impact of this work does not necessarily lay on its results, but rather on the limitations it has highlighted. Indeed, the heterogeneity of the context data appears to be a major obstacle for a generalized and fully automated analysis of these data. Very often, data mining methods are applied to certain “categories” of data. For example, the FFT (Fast Fourrier Transformation) used in [133] is particularly adapted to numerical data, whereas the FCA applied is more adapted to symbolic data (*i.e.* labels). The localization data demonstrate this issue particularly well. A pre-processing was necessary in order to translate the localization data, extracted during a first experiment we have performed using an Android tablet, into "labels" (localization_1, localization_2, etc.) that could be more easily exploited by the FCA. During this experiment, a “spy” application has observed several context elements, including GPS coordinates and services executed in the tablet. The GPS coordinate data had to be classified, during a pre-processing phase, into several geographic regions, to which labels were assigned. It is these labels that were used during FCA analysis. Without this transformation, the application of FCA was unviable. Therefore, given the heterogeneity that characterizes context data, it is difficult to imagine the application of a single technique for data analysis regardless of the context element considered, and at the same time, it is difficult not to consider the risk of losing information on the possible links between the different context elements by splitting the data set according to their nature.

It seems clear that current data mining techniques are not sufficiently adapted to manage the heterogeneity of context data without a significant pre-processing or training phase. Based on this

observation, we wanted to analyze to what extent the difficulties we encountered with the application of FCA could also occur when using other Machine Learning methods. Thus, in 2019-2020, a collaborative project [11] was launched to study the use of Machine Learning (ML) techniques with context data. We analyzed how ML approaches are used for context mining and the conditions necessary for this analysis considering the specificities of context information. These limitations, which we have observed during the application of the FCA, suggest an issue when considering the possible generalization of ML techniques for context mining on a large scale, *i.e.* on the scale of an entire Information System (IS). Indeed, in [11], we introduce a vision of a "context mining facility", in other words, context mining functionalities proposed as a service offered by an IS for all its applications. Such "context mining facility" would open up new application perspectives within IS (adaptation to the context, recommendation of services or content, prediction/anticipation of user's needs or actions, decision-making, etc.). We may thus envisage the generalization of these behaviors, which could be described as "smart" or "intelligent", by offering an appropriate support for context mining as a service specific to an IS.

Last but not least, these difficulties encountered in the analysis of context data have led to a change of focus in Ali Jaffal's PhD thesis [71], in which part of this work has been carried out. These difficulties have demonstrated the limitations of FCA when applied to heterogeneous data such as context information, but also the difficulties in the application of this technique for non-expert analysists. The focus of this PhD thesis has then changed to this method of analysis and the possibility of making the application of this later better handling and easier for non-expert users. Pervasive Computing thus became a case study for the PhD thesis, and not its main subject, especially considering the limitations for an automatic application of this method to context-aware systems.

The paper published at CoMoRea 2020 [11] represents the latest published result mentioned in this document. It opens up several interesting perspectives, mainly considering the vision of a "context facility" and its challenges. For this reason, this paper has been chosen to represent this work. It can be found in the Annex VI.

- **CoMoRea 2020** [11] : Ben Rabah, N.; Kirsch Pinheiro, M.; Le Grand, B.; Jaffal, A. & Souveyet, C., "Machine Learning for a Context Mining Facility", *16th Workshop on Context and Activity Modeling and Recognition, 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, **2020**, pp.678-684.

3.1.2 Bibliometric

This research work on context mining began in 2012, with Ali Jaffal's MSc. work, and continued until 2014 as part of his PhD thesis, under the supervision of Bénédicte Le Grand. Three papers directly resulting from this work can be cited here [117, 116, 120], to which it can be added our last article on the applicability of Machine Learning techniques to context information [23]. Moreover, this work has highlighted the potential use of FCA with context data, which has opened the door to several collaborations in other fields, notably BPM (Business Process Modeling). Several publications mentioned below (Adaptive 2013, BPMDS 2013, Chapter BPM 2014, CoMoRea 2020) are the result of these collaborations, which would not have taken place without this work on context mining. All these publications are listed below. As for the previous chapters, each publication has been analyzed, using the scholar.google.com website, concerning the number of citations. These citations have been organized into two categories, according to their publication date: before 2016 (up to 3 years after the first publication) or after 2016. Self-citations were also included. Table 9 details the data obtained, illustrated in Figure 9.

- **ANT 2015** [69]: Jaffal, A. ; Grand, B. L. ; Kirsch-Pinheiro, M., "Refinement Strategies for Correlating Context and User Behavior in Pervasive Information Systems", *1st Workshop on Big Data and Data Mining Challenges on IoT and Pervasive (Big2DM), 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015)*, Procedia Computer Science, vol. 52, Jun **2015**, pp.1040-1046. DOI : http://dx.doi.org/10.1016/j.procs.2015.05.103
- **Ubicomm 2014** [70]: Jaffal, A. ; Kirsch-Pinheiro, M. & Le Grand, B., "Unified and Conceptual Context Analysis in Ubiquitous Environments", In : Jaime Lloret Mauri, Christoph Steup, Sönke Knoch (Eds.), *8th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2014)*, August 24 - 28, **2014**, ISBN 978-1-61208-353-7, IARIA, pp. 48-55.
- **EGC 2016** [73]: Jaffal, A.; Grand, B. L. & Kirsch-Pinheiro, M., « Extraction de connaissances dans les Systèmes d'Information Pervasifs par l'Analyse Formelle de Concepts », Extraction et Gestion des Connaissances (EGC 2016), Revue des Nouvelles Technologies de l'Information, RNTI-E-30, 2016, pp. 291-296
- **AdaptiveCM 2013** [88]: Kirsch-Pinheiro, M. & Rychkova, I., "Dynamic Context Modeling for Agile Case Management", In : Y.T. Demey and H. Panetto (Eds.), *2nd International Workshop on Adaptive Case Management and other non-workflow approaches to BPM (AdaptiveCM 2013), OnTheMove Federated Workshop (OTM 2013 Workshops)*, LNCS 8186, Graz, Austria, 9-13 September **2013**, Springer, pp. 144–154, 2013.
- **BPMDS 2013** [134]: Rychkova, I. ; Kirsch Pinheiro M. & Le Grand B., "Context-Aware Agile Business Process Engine: Foundations and Architecture", In : Nurcan, S., Proper, H., Soffer, P., Krogstie, J., Schmidt, R., Halpin, T. & Bider, I. (Eds.), *Enterprise, Business-Process and Information Systems Modeling, Proceedings of the 14th Working Conference on Business Process Modeling, Development, and Support (BPMDS 2013)*, Lecture Notes in Business Information Processing, vol. 146, Valence : Espagne, **2013**, pp. 32-47.
- **Chapitre BPM 2014** [135]: Rychkova I. ; Kirsch-Pinheiro M. & Le Grand B., "Automated Guidance for Case Management: Science or Fiction?", In : Ficher, L. (Ed.), *Empowering Knowledge Workers: New Ways to Leverage Case Management*, Series BPM and Workflow Handbook Series, Future Strategies Inc., **2014**, pp. 67-78. ISBN : 978-0-984976478
- **CoMoRea 2020** [11]: Ben Rabah, N.; Kirsch Pinheiro, M.; Le Grand, B.; Jaffal, A. & Souveyet, C., "Machine Learning for a Context Mining Facility", *16th Workshop on Context and Activity Modeling and Recognition, 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, **2020**, pp.678-684

As can be seen from Table 9, these articles still have a small number of citations, most of them concentrated in the early years of their appearance (see Figure 38). This behavior is not surprising since, on the one hand, context mining remains a relatively confidential and developing field, especially when aimed at identifying the relevance of contextual information. Few authors have addressed this issue, with most works focusing on identifying a user's activities. On the other hand, the use of the notion of context in BPM is not yet generalized, even if this use is developing more and more (*e.g.* [129,140]), which explains the higher number of citations for publications in this field. From this information, it seems evident that, for the moment, the main impact of this work cannot be measured in terms of citations. Instead, it must be evaluated in terms of the opportunities it has opened up and in terms of the collaborations it has allowed to develop. In addition to the publications mentioned above (Adaptive 2013, BPMDS 2013, Chapter BPM 2014, CoMoRea 2020), born of exchanges with other researchers, the exchanges around context mining also led to an invitation to Arun Ramakrishnan's PhD thesis jury [133], at the K.U. Leuven university, as an external member of the jury.

Table 4. Bibliometric analysis concerning papers on context mining.

	Year	Total	≤ 2016	> 2016	Self-citation
ANT 2015	2015	2	1		1
Ubicomm2014	2014	7	1	1	5
EGC 2016	2016				
AdaptiveCM 2013	2013	5	2	2	1
BPMDS 2013	2013	10	4	5	1
Chapitre BPM 2014	2014	2	1	1	
CoMoRea 2020	2020				
Total / %		**21**	**42,86 %**	**28,57 %**	**28,57 %**

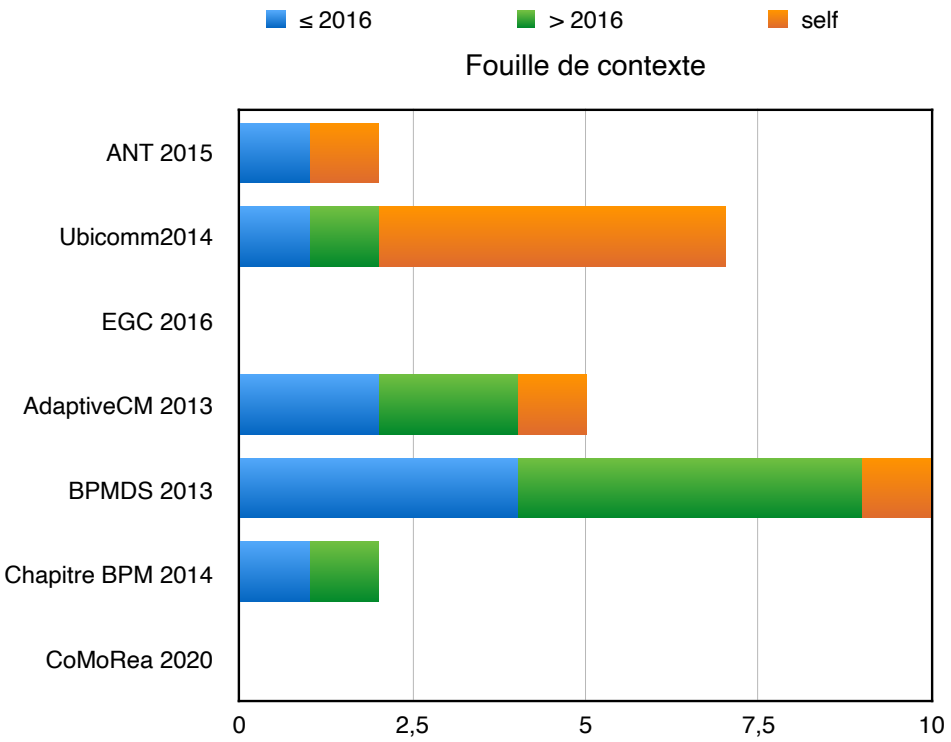


Figure 9. Citations evolution for context mining works over different periods of time.

3.2 Service selection using graphs

3.2.1 Problem statement

In a pervasive environment, characterized by its dynamism and its heterogeneity, the same service can have several distinct implementations. Indeed, a service offers an interface allowing a customer to invoke it without knowing how it is implemented. This principle of an invokable unit through a "standard" interface makes service orientation an interesting paradigm for managing interoperability. Under this interface, it is possible to imagine different implementations (for example, proposed by different vendors or offering different qualities). As pervasive environments are particularly heterogeneous, proposing alternative implementations for the same interface become interesting, even desirable, in order to better support variability on these environments.

Such variability raises the question of choosing the implementation that adapts the best to the execution circumstances. These circumstances can be assimilated to the execution context of a service (or its client). However, since the context information is uncertain and often incomplete, the lack of information can significantly impact the selection mechanism. In this case, a second question arises: how to choose the implementation of a service that is best adapted to the execution context, knowing that this information may be incomplete? The work described here addresses precisely this question.

Carried out between 2008 and 2009 as part of the European project IST-MUSIC, this research work proposes a context-aware service selection mechanism based on graph comparison, inspired by the similarity measures proposed for groupware systems (chapter 1). From a first selection, in which only the functional aspects are considered, this mechanism seeks to identify the implementation (already meeting the functional demand) that best suits the current context of use. The mechanism considers that each implementation is designed to target specific situations (*e.g.* a service aimed particularly at customers in a particular geographical area or requiring specific resources). These situations are described by a “required” context associated with the OWL-S description of the service. This context is then compared to the customer's execution context. These two descriptions (the required context and the client context) are interpreted as graphs, in which the context elements are seen as nodes and their relationships as arcs connecting them. The two graphs are compared using different similarity measures: first so-called “local” measures, which compare the nodes individually, and then “global” measures, which look at both graphs as a whole. Finally, the results obtained by these measures are used to rank the implementations responding to the functional aspects, thus making it possible to select the one that is the most favorable to the execution context on the client side.

In this mechanism, context information is then considered as a non-functional requirement associated with each implementation of a service. The objective is to sort out the different implementations that can satisfy the customer's request and select the one that seems to better match the context of use.

Besides, the present work has influenced in its turn other following works, such as Salma Najar’s PhD thesis [107] (whose contribution is discussed in the following sections), in which we adopt the same idea of a required context in the description of services presented in this work. The implementation also follows the same inspiration, with a “context” element added to the service description pointing to an external XML description, which can be updated more easily.

This service selection mechanism was first published in NFPSLA-SOC 2008 workshop [87]. This paper detailing the proposed mechanism is presented in the Annex VII.

- **NFPSLA-SOC 2008** [87] : Kirsch-Pinheiro, M.; Vanrompay, Y. & Berbers, Y., « Context-aware service selection using graph matching ». In: Paoli, F. D.; Toma, I.; Maurino, A.; Tilly, M. & Dobson, G. (Eds.), *2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC'08), at ECOWS 2008*, CEUR Workshop proceedings, vol. 411, **2008**.

3.2.2 Bibliometric

The research work described here has led to three publications, including a book chapter. It has also allowed the production of a fourth publication (Chapter IGI 2009), in which different concerns of context-aware applications are analyzed. These publications are listed below. Data concerning citations of these papers have been obtained, as for previous chapters, from Web site scholar.google.com. These citations have been organized in groups (see Table 4), according to the date of their publication (before 2013, between 2013 and 2016, and after 2016), in addition to self-citations. Figure 10 illustrates these results.

- **NFPSLA-SOC 2008** [87] : Kirsch-Pinheiro, M.; Vanrompay, Y. & Berbers, Y., « Context-aware service selection using graph matching ». In: Paoli, F. D.; Toma, I.; Maurino, A.; Tilly, M. & Dobson, G. (Eds.), *2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC'08), at ECOWS 2008*, CEUR Workshop proceedings, vol. 411, **2008**.

- **CAMPUS 2009** [165] : Vanrompay, Y.; Kirsch-Pinheiro, M. & Berbers, Y., "Context-Aware Service Selection with Uncertain Context Information", *Context-Aware Adaptation Mechanism for Pervasive and Ubiquitous Services 2009 (CAMPUS 2009), Electronic Communications of the EASST, vol. 19*, **2009**.
- **Chapitre IGI 2011** [166] **:** Vanrompay, Y.; Kirsch-Pinheiro, M. & Berbers, Y., "Service Selection with Uncertain Context Information", In: Stephan Reiff-Marganiec and Marcel Tilly (Eds.), *Handbook of Research on Service-Oriented Systems and Non-Functional Properties: Future Directions*, IGI Global, pp. 192-215, **2011**.
- **Chapitre IGI 2009** [130] **:** Preuveneers, D.; Victor, K.; Vanrompay, Y.; Rigole, P.; Kirsch Pinheiro, M. & Berbers, Y. « Context-Aware Adaptation in an Ecology of Applications ». In: Dragan Stojanovic (Ed.), *Context-Aware Mobile and Ubiquitous Computing for Enhanced Usability: Adaptive Technologies and Applications*, IGI Global, **2009**.

The first of these publications (NFPSLA-ECOWS 2008 [87]) describes the principles of this mechanism, while the subsequent publications (CAMPUS 2009 [165] and Chapter IGI 2011 [166]) have further improved it by adding quality consideration through metadata and transformation functions. The last of these publications (Chapter IGI 2011 [166]) is the one with the most self-citations, since it summarizes, in a rather complete way, the contributions of the previous publications. It has thus been used as a reference for the works following this one.

Table 5. Bibliometric analysis of publications concerning graph service selection research works.

Reference	Year	Total	≤ 2013	> 2013 & ≤ 2016	> 2016	Self-citation
NFPSLA-ECOWS 2008	2008	43	22	11	3	7
CAMPUS 2009	2009	9	1	4	4	0
Chapitre IGI 2009	2009	19	7	6	0	6
Chapitre IGI 2011	2011	11	0	3	0	8
Total / %		**78**	**28,21 %**	**41,03 %**	**3,85 %**	**26,92 %**

Among these publications, the one with the most citations remains the first, presenting the mechanism's founding principles. These citations are spread over the first 8 years following its publication (from 2008 to 2016) and fade after this date. This period (2008 to 2016) corresponds to a period in which service orientation has been widely discussed in Pervasive Computing. Today, the notion of service, as described in the SOA community, is gradually being replaced by the notion of micro-service. Even though conceptually these two notions are fundamentally similar, the micro-service approach is now perceived as more advantageous because it is more suitable for implementation in a virtualization and Cloud Computing context. In any case, the proposals from this research work remain valid and could, if necessary, be adapted to the micro-service concept.

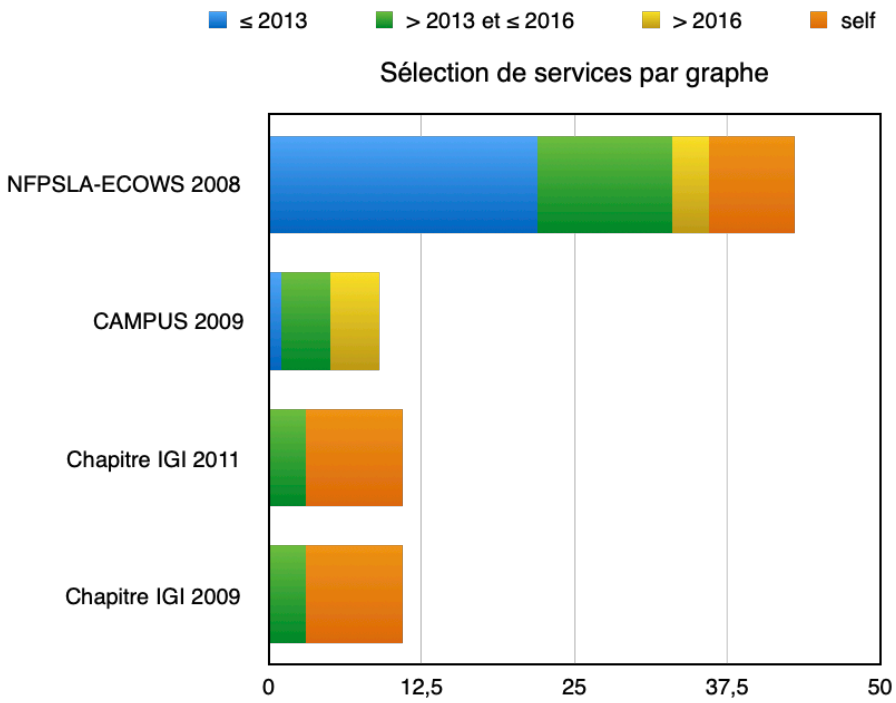


Figure 10. Distribution of the papers citations all along the time.

3.3 Service selection by the user's intention and context

3.3.1 Problem statement

The introduction of new technologies (such as smartphones, IoT, Big Data) brings profound changes in organizations and in their Information Systems (IS), as they are now facing a pervasive environment. These systems and their users are thus confronted with a growing heterogeneity that must be managed. Since these systems are transforming and becoming more and more complex with the arrival of new technologies, it becomes crucial to hide the heterogeneity of these systems and the services in such way users may focus more on the objectives to be achieved rather than on these technologies. More than ever, systems need to evolve towards the vision Ubiquitous Computing given by Weiser [171] in which technology becomes invisible to the user. These new systems, which can be called Pervasive IS (which will be discussed in chapter 4), have an increasing need for transparency. Therefore, this research work aims at improving the transparency of Information Systems by hiding the technical details concerning the services, by taking into account the notion of context and the user's business goals, represented by the notion of intention.

Carried out between 2009 and 2012 along with Salma Najar's Ph.D. thesis [107], this work proposes a service selection mechanism that takes into account both the user's context and her/his intention. The notion of intention corresponds to a widely recognized notion in the field of Requirements Engineering [174, 68, 163, 136]. It is used to represent the user's business goals and has already been associated with the notion of service in the past [46, 74, 103, 137, 141].

The use of the notion of intention denotes the user-centered focus adopted by the present work, which distinguishes it from previous works on service selection (cf. section 3.2). Although previous work considers the context in which a service is requested, it considers mainly the context of a client application requesting the execution of a service. In the present works, the user is directly considered, with her/his needs and context of use. The main focus when analyzing the notion of context is, therefore, the user herself/himself. This user-centered approach can also be found in our research works on Pervasive Information Systems, which will be discussed later (see Chapter 4).

Besides, this work is based on the hypothesis that an intention emerges in a particular context, which in turn influences the choice of an implementation. In other words, users would invoke a service because they have a particular intention (*i.e*. a goal to achieve) that arises in a particular context, which in its turn will determine the choice of the service implementation to be executed. This work was among the first ones (in 2009) connecting the user's intentions, her/his context of use, and the services invoked. We may cite [141, 46] as examples of works combining the notion of intention with the notion of context. However, it is worth noting that, unlike those works, ours propose a detailed semantic description of both the notions of intention and context, which is not necessarily the case in other works in the literature such as [141, 46], who focus more on one or another of these aspects.

The selection mechanism itself is a two-step mechanism, taking advantage of both an ontology describing the notion of intention and an ontology modeling the notion of context. In the first step, the intention requested by the user and likely to be satisfied by a service is analyzed and related to those declared by the available services (through their descriptors). It is important to note that the service look up is made directly on the basis of its intention and no longer according to the functionalities offered by the service, as traditionally in the SOA approach. The functional description of a service (corresponding to the functionalities expected by the service, expressed through the operations it is capable of performing) remains hidden from the user, who looks for a service only through its intention (*i.e*. its business goals). In the second step, the mechanism compares, using the context ontology, the elements present in the context required by the service and those present in the user's current context, evaluating to which extension the conditions expressed in the required context

are fulfilled by the elements in the user's context. The values present in the user's observed context (at user side) are thus confronted with these conditions in order to assess their level of satisfaction. The score obtained by each service over these two steps is then used to rank the available services. It is worth noting that, similar to our previous works (cf. section 3.2), a service can be proposed to a user even if its required context does not fully correspond to the user's current context.

This research work was the subject of multiple publications (see section 3.3.2), considering different aspects of this work, from the semantic descriptions proposed for it until the selection mechanism itself. Among these publications, the paper published at ICWS 2012 appears as the most complete one, detailing the proposed mechanism and some experiments we have performed in order to validate the mechanism. This paper is proposed on the Annex VIII.

- **ICWS 2012** [118] : Najar, S.; Kirsch-Pinheiro, M.; Souveyet, C. & Steffenel, L. A., "Service Discovery Mechanisms for an Intentional Pervasive Information System". *Proceedings of 19th IEEE International Conference on Web Services (ICWS 2012)*, Honolulu, Hawaii, 24-29 June **2012**, pp. 520-527.

3.3.2 Bibliometric

As with the works discussed previously, this research work also led to several publications (a total of 7), including a journal paper. This subject is also mentioned in two other publications, including our developments on service prediction (cf. section 3.4), which will be considered in the next section. Thus, similar to previous chapters, the publications cited below have been organized in different groups (see Table 6), according to their publication date (before 2013, between 2013 and 2016, and after 2016), in addition to self-citations. Figure 11 illustrates these results.

- **CIAO 2009** [108]: Najar, S.; Saidani, O.; Kirsch-Pinheiro, M.; Souveyet, C. & Nurcan, S., "Semantic representation of context models: a framework for analyzing and understanding". In: J. M. Gomez-Perez, P. Haase, M. Tilly, and P. Warren (Eds), *Proceedings of the 1st Workshop on Context, information and ontologies (CIAO 09), European Semantic Web Conference (ESWC'2009)*, Heraklion, Greece, June **2009**, ACM, pp. 1-10.
- **Service 2011** [110]: Najar, S.; Kirsch Pinheiro, M. & Souveyet, C., "Bringing context to intentional services". *3rd Int. Conference on Advanced Service Computing, Service Computation'11*, Rome, Italy, pp. 118-123, **2011**. Best Paper Awards.
- **REFS 2011** [109]: Najar, S.; Kirsch Pinheiro, M. & Souveyet, C., "The influence of context on intentional service". *5th Int. IEEE Workshop on Requirements Engineerings for Services (REFS'11), IEEE Conference on Computers, Software, and Applications (COMPSAC'11)*, Munich, Germany, pp. 470-475, **2011**.
- **WEWST 2011** [111]: Najar, S.; Kirsch Pinheiro, M. & Souveyet, C., "Towards Semantic Modeling of intentional pervaisve System", *6th International Workshop on Enhanced Web Service Technologies (WEWST'11), European Conference on Web Services (ECOWS'11)*, Lugano, Switzerland, **2011**, pp. 30-34.
- **IJAIS 2012** [106]: Najar, S. ; Kirsch-Pinheiro, M. & Souveyet, C., "Enriched Semantic Service Description for Service Discovery: Bringing Context to Intentional Services", *International Journal On Advances in Intelligent Systems*, volume 5, numbers 1 & 2, June **2012**, pp. 159-174, IARIA Journals / ThinkMind, ISSN: 1942-2679.
- **ICWS 2012** [118]: Najar, S.; Kirsch-Pinheiro, M.; Souveyet, C. & Steffenel, L. A., "Service Discovery Mechanisms for an Intentional Pervasive Information System". *Proceedings of 19th*

IEEE International Conference on Web Services (ICWS 2012), Honolulu, Hawaii, 24-29 June **2012**, pp. 520-527.

- **UbiMob 2012a** [119]: Najar, S. ; Kirsch-Pinheiro, M. ; Steffenel, L. A. & Souveyet, C., « Analyse des mécanismes de découverte de services avec prise en charge du contexte et de l'intention ». In : Philippe Roose & Nadine Rouillon-Couture (dir.), *8èmes Journées Francophones Mobilité et Ubiquité (Ubimob 2012)*, June 4-6, **2012**, Anglet, France. Cépaduès Editions, pp. 210-221. ISBN 978.2.36493.018.6.

Table 6. Citations concerning papers on context and intention service selection obtained from Google Scholar

Reference	Year	Total	≤ 2013	> 2013 & ≤ 2016	> 2016	Self-citation
CIAO 2009	2009	51	14	8	11	18
Service 2011	2011	6	2	2	0	2
REFS 2011	2011	13	1	2	2	8
WEWST 2011	2011	7	0	3	0	4
IJAIS 2012	2012	4	0	0	1	3
ICWS 2012	2012	18	4	4	3	7
UbiMob 2012a	2012	1	0	0	0	1
Total / %		**100**	**21,00 %**	**19,00 %**	**17,00 %**	**43,00 %**

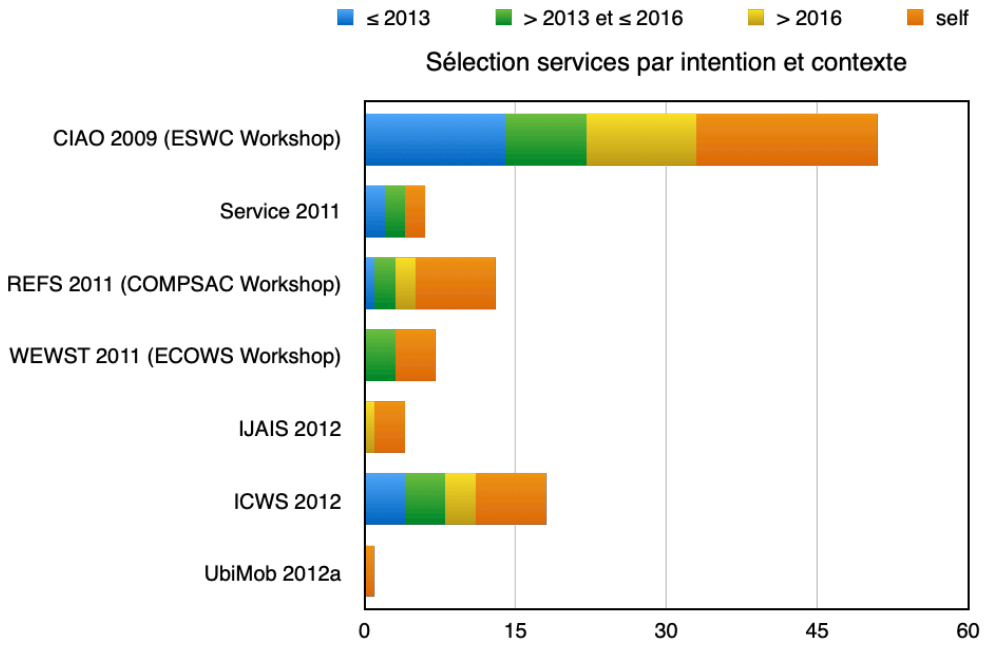


Figure 11. Graphic illustrating the evolution of papers citation over the time.

Among these publications, the one with the most citations (CIAO 2009 [108]) provides a framework for the analysis and the comparison of different context models. This framework has been widely used in our subsequent works for the analysis of the proposed context models. Focusing the analysis of context models in a general way may also explain the more significant impact of this article in terms of citations compared to other publications mentioned here. It should also be noted that the significant number of self-citations in these publications is justified by continuity of this work, many of which have served as a basis for further work.

Finally, one may also observe that most of the citations are concentrated in the first 4-6 years following the publications (from 2009 to 2016), which corresponds to a period when the selection of services reflected a popular research problem in the literature on Pervasive Computing.

3.4 Service prediction

3.4.1 Problem statement

When considering the evolution of Information Systems (IS) towards Pervasive IS, these new systems are characterized, among other things, by an increased need for transparency, as discussed earlier (see section 3.3). Hiding the technological complexity that now characterizes IS from end-users is essential, freeing users from technological constraints so that they could concentrate on tasks with an added value to the organization. To do this, proactivity becomes a necessity. Indeed, users expect an increasingly "intelligent" IS, capable of anticipating their needs and responding to them appropriately. According to Bauer & Dey [7], we can already witness a move towards increasingly sophisticated systems ("smart", "intelligent", "context-aware", "adaptive", etc.). Here, the notion of context is central as systems become aware of the context in which they are used and intelligently adapt their execution. We believe that the democratization of this kind of behavior that could be considered "intelligent" creates a certain expectation on the users: they now expect that a software and a system will be more intelligent, it will be able to recognize their situation, their behavior, and to adapt itself in a reactive way as well as in a proactive way. Thus, we advocate that Information Systems must be able to recognize user's habits and practices in order to be able to anticipate users' needs and proactively propose to these users the services that correspond the best to their needs.

Similar to the works presented in the previous section (cf. section 3.3), this research work on service prediction is also part of Salma Najar's PhD thesis [107]. In these works, carried out between 2012 and 2014, a service prediction mechanism is proposed. This mechanism allows, based on the user's history and her/his current context, to anticipate the user's future intentions and context of use, and thus to proactively provide her/him the next service that the user would probably request. The main objective here is to anticipate the user's next intentions and context of use from the her/his history and current context of use, and then be able to proactively provide her/him the next service that this user will most probably request.

The basic hypothesis of the one describing the user as someone with particular habits and work practices. During her/his professional activities, a user of an IS can acquire certain work habits related to her/his professional routine (for example, a salesman who solicits on her/his tablet data about a client when she/he arrives at this client's office). We assume that it is possible to characterize, and therefore automatically identify, these habits through the intentions behind these actions (the reason motivating the service request) and the context in which these intentions arise.

It is important to note that this research on service prediction work also considers the triplet "*< intention, context, service >*", similar to our works on intentional and contextual service selection,

presented previously (cf. section 3.3). In addition, the traces resulting from this selection mechanism are considered here. Thus, from the traces of previous solicitations, by applying a clustering algorithm, the proposed prediction mechanism will first group similar situations already observed by comparing them using different similarity measures. Each situation is represented by the triplet "*< intention, context, service >*" from a previous user's request (and thus from a previous service selection process). These clusters are organized using a Markov chain model, which calculates, from the known data about these situations, the probability of moving from one cluster (*i.e.* from a "typical" situation) to another. Thanks to this model, and as soon as the current situation (*i.e.* the expressed intention, the current context, and the service selected to meet this demand) is sufficiently similar to an already identified cluster, it is possible to anticipate the next most probable situation (intention, context, and service).

Anticipating the triplet *"< intention, context, service >"* thus responds to the habit hypothesis stated above, but also to the founding hypothesis of Salma Najar's PhD work, already used in the previous works on service selection (see section 3.3). This hypothesis establishes that an intention emerges in a particular context, which influences the choice of which service implementation is most likely adapted to satisfy this intention. Thus, even if this mechanism leads to the prediction of the next service to be invoked on the user's behalf, it determines not only the next service but all the triplet "*< intention, context, service >"*. Therefore, it is the intention and the context in which it emerges that are anticipated here. This behavior differs from other works on context prediction, such as [95, 102, 148, 167], which focus on predicting the next context of use or on missing elements in the observed context. Our work can also be distinguished from those, such as [62], which focus on predicting context-aware services. Generally speaking, service prediction can be compared to recommendation systems [1, 19, 128], whose objective is often to propose personalized content according to other users' traces, but also to works considering recommendation in business processes [20, 39, 59]. Here again, the notion of intention and context are not combined in an anticipation process, either of an activity or a service. To the best of our knowledge, our work remains one of the few to have combined these two elements (context and intention) in a prediction mechanism.

Finally, it should be emphasized that the clustering process used in our work is based on the similarity of concepts, which means that the user does not need to behave precisely as in a previous situation. This is particularly important, especially with regard to the dynamic nature of context information, which can vary between multiple observations. If these variations are not significant, it is still possible to classify a request, even if it is not exactly the same as those observed previously.

Similar to our previous works, this research work produced several publications, focusing on different aspects of the service prediction mechanism. Among these publications (see section 3.4.2), the paper published on ICWS 2014 [115] details the proposed mechanism and present some experimental results on it. This paper is proposed in the Annex IX.

- **ICWS 2014** [115]: Najar, S. ; Kirsch-Pinheiro, M. & Souveyet, C., "A context-aware intentional service prediction mechanism in PIS", In: David De Roure, Bhavani Thuraisingham & Jia Zhang (Eds.), *IEEE 21st International Conference on Web Services (ICWS 2014)*, 27 June - 2 July **2014**, Anchorage, Alaska, USA, IEEE CS, pp. 662-669. DOI : 10.1109/ICWS.2014.97

3.4.2 Bibliometric

This work on service prediction has resulted in 5 publications, cited below. As for previous works, we have studied the citations to these works, obtained from scholar.google.com Web site. According to their date of publication, these citations have been organized in two groups (pre-2016 and post-2016), covering the first years after the appearance of these publications and the last 4 years. Table 7 groups this information, illustrated in Figure 12.

- **UbiMob 2012b** [120] : Najar, S. ; Kirsch-Pinheiro, M. & Souveyet, C., « Mécanisme de prédiction dans un système d'information pervasif et intentionnel », In : Philippe Roose & Nadine Rouillon-Couture (dir.), *8èmes Journées Francophones Mobilité et Ubiquité (Ubimob 2012)*, June 4-6, **2012**, Anglet, France. Cépaduès Editions, pp. 146-157. ISBN: 978.2.36493.018.6
- **ICWS 2014** [115] : Najar, S. ; Kirsch-Pinheiro, M. & Souveyet, C., "A context-aware intentional service prediction mechanism in PIS", In: David De Roure, Bhavani Thuraisingham & Jia Zhang (Eds.), *IEEE 21st International Conference on Web Services (ICWS 2014)*, 27 June - 2 July **2014**, Anchorage, Alaska, USA, IEEE CS, pp. 662-669. DOI : 10.1109/ICWS.2014.97
- **ANT 2014** [116] : Najar, S. ; Kirsch-Pinheiro, M. & Souveyet, C., "A new approach for service discovery and prediction on Pervasive Information System", *5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014), Procedia Computer Science*, vol. 32, **2014**, Elsevier, pp. 421–428. DOI : http://dx.doi.org/10.1016/j.procs.2014.05.443
- **Chapitre IGI 2013** [117] : Najar, S.; Kirsch Pinheiro, M.; Vanrompay, Y.; Steffenel, L.A. & Souveyet, C., "Intention Prediction Mechanism in an Intentional Pervasive Information System", In : Kolomvatsos, K., Anagnostopoulos, C., Hadjiefthymiades, S. (Eds.), *Intelligent Technologies and Techniques for Pervasive Computing*, IGI Global, **2013**, pp. 251-275. DOI: 10.4018/978-1-4666-4038-2.ch014, ISBN : 978-1-4666-4040-5.
- **JAIHC 2015** [114] : Najar, S.; Kirsch-Pinheiro, M. & Souveyet, C. "Service discovery and prediction on Pervasive Information System", *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, issue 4, June **2015**, Springer Berlin Heidelberg, pp. 407-423. ISSN: 1868-5137. DOI: http://dx.doi.org/10.1007/s12652-015-0288-5

Table 7. Bibliometric analysis of papers related to the service prediction topic, based on Google Scholar data.

Reference	Year	Total	< 2016	≥ 2016	Self-citation
UbiMob 2012b	2012	2	2	0	0
ICWS 2014	2014	1	0	1	0
ANT 2014 :	2014	18	2	14	2
Chapitre IGI 2013	2013	2	1	0	1
JAIHC 2015	2015	8	0	5	2
Total / %		**31**	**16,13 %**	**64,52 %**	**19,35 %**

As can be seen from Table 7, the number of citations to these papers is still limited. The relatively recent nature of this work partly explains this phenomenon. Besides, one may observe in Figure 12 that most citations were made from 2016 onwards, suggesting a theme that is still in development. Indeed, the recent development of Machine Learning and Artificial Intelligence techniques could bring a new perspective to this work, carried out before the democratization of these techniques.

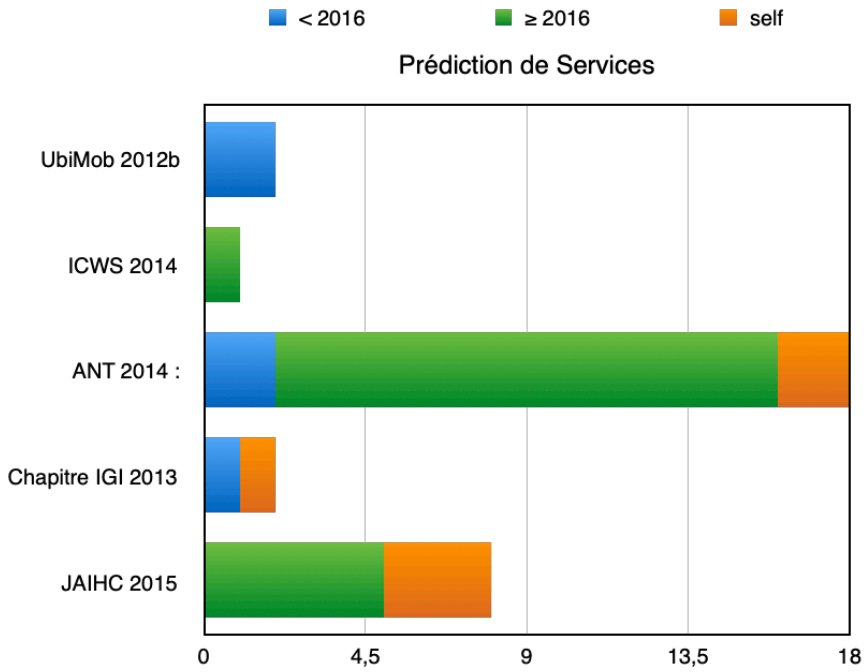


Figure 12. Evolution of citations concerning service prediction papers over the time.

3.5 Chapter Summary

In the previous sections, several contributions in the Pervasive Computing community have been highlighted: context mining by FCA, service selection based on the context of use and also service selection based on the user's context and intention, and finally service prediction. All these works were carried out between 2008 and 2014 (2012-2014 for context mining, 2008-2009 for graph selection, 2009-2012 for selection by intention and context, and 2012-2014 for service prediction), involving two Ph.D. thesis (Ali Jaffal 2012-2014 and Salma Najar 2009-2014), but also a European project (the IST-MUSIC project in 2008 for service selection with graphs).

The common denominator of these works, beyond the notion of context, is the use of the notion of service: the functionalities proposed by a system are encapsulated behind this notion. The service notion represents an interesting asset for the Pervasive Computing community, not only because of the interoperability services offer, but also because of the transparency they may provide, since different technologies can be hidden behind this notion. Besides, the loose coupling that characterizes service orientation enables better management of the dynamic nature of pervasive environments.

The various contributions presented here differs from others works considering context-aware services by considering the uncertainty of context data and the user's intention. Our works on service selection bring the idea of a required context, which explores the idea that a particular implementation is designed assuming a specific client-side execution context. Setting conditions to the user's context is a topic with limited coverage in the literature, mostly focused on the execution context at the provider side. Indeed, the numerous works involving SLA and QoS mostly focus only on the provider-side, considering whether or not the provider can provide the service as desired. However, taking the user's context into account in the service selection process opens the door to a better consideration of this user and places her/him in the center of the process. We find here the basis of a user-centered vision that particularly characterizes our work.

These works on service selection by intention and context have consolidated our user-centric vision, based on the assumption that a service is solicited to satisfy an intention in a given context. Intention

emerges in this context, which also influences the service selection and its execution. This hypothesis remains innovative and opens up exciting perspectives for Information Systems, in which the notion of intention can easily be associated with expected or high value-added functionalities and services.

Besides, other aspects should also be emphasized in the contributions presented here, starting with the application of the FCA for the identification of usage situations from the observed data. The use of FCA is an innovative approach to context data mining since it allows a multi-class classification of these elements, which can be particularly interesting for the identification of complex situations. A situation can be seen as a set of context elements that characterize the circumstances of an action. The notion of situation allows representing a higher level of granularity than simple context elements. On this level, it is easier to assign a certain semantics from the user's point of view to this set of elements, hence the importance of the overlaying character of the FCA. Having possible overlaps between classes provides more flexibility for better identification of these situations. Nevertheless, the challenges for a fully automated (or semi-automated, with minimal human intervention) use of this method remain.

Finally, it should also be noted that the use of the triplet *"< intention, context, service >"*, especially in our work on service prediction, is one of the most characteristic features of these works. It is a rather innovative approach that foresees a more proactive functioning for Information Systems. More than anticipating the next service to be offered to a user, our work on prediction enables us to anticipate the user's next intention and a possible context for this intention to appear. We are again approaching the notion of situation, with a set of context elements now characterized by an intention. Furthermore, we find here again the user-centered vision that characterizes most of our work in this field. This vision is further developed in Chapter 4, which focuses on the Pervasive Information Systems.

4 Context on Pervasive Information Systems

The work described in this chapter focuses particularly on Information Systems and their "new generation", here called Pervasive Information Systems (PIS), in which the notion of context plays a central role.

In order to better understand this evolution of Information Systems (IS), we shall first look at the recent transformations undergone by IS. The last decade has witnessed several technological evolutions and new uses that have strongly impacted IS. Among the new trends that have emerged in recent years, we may cite 4G and BYOD, IoT, Big Data, Cloud Computing and Fog Computing, and the democratization of Machine Learning.

The development of mobile technologies, including 4G, has contributed to the democratization of the Internet access with a reasonable bandwidth almost everywhere, which has also contributed to the adoption of the BYOD (Bring Your Own Device) practice. BYOD consists in using one's own personal computer at work. According to this practice, employees use their own personal terminals to work, navigating seamlessly between their private and work spaces, instead of accumulating multiple terminals according to circumstances, location or professional needs [27]. This mix of personal and professional hardware represents a significant change for organizations IT departments, which traditionally govern, deploy and control all technologies used by employees/collaborators for their professional activities [41]. Today, it becomes common (or usual) to use your own personal devices (which are no longer limited to laptops) to access your company's information system, wherever you are. A ubiquitous access "Anytime, anywhere" from any kind of terminal has become a reality. According to Andriole & Bojanova [4], the use of new devices such as Microsoft HoloLens, Apple Watch, and other Bluetooth devices, creates new opportunities for businesses as these new devices are changing the way we browse, search, shop, and even live. It is therefore natural to think that the arrival of these new personal devices in organizations can also change the way we work.

Similarly, IoT also offers new opportunities of interacting with the physical environment, and through these new interactions, it brings new business perspectives. According to Sundmaeker *et al.* [160], it is expected that IoT objects will become more and more active, participating in different aspects of society, through business, information and social process [160]. The informational aspect remains probably the most prominent one within today's organizations. Thanks to the IoT, it is possible to easily (and even continuously) collect information from the physical environment, but also to act upon this environment through sensors and actuators often connected to networked nanocomputers with some computing power. The physical environment can then become an integral part of business processes and, consequently, part of the Information System itself, as shown by the recent development of Industry 4.0, which heavily relies on the IoT and on the data coming from it, as observed by Lu [100].

The data collected from IoT objects enriches an already large set of available data within organizations. Big Data platforms allow to better control this impressive data volume and to exploit it properly. The recent success of Data Lakes [186], often built on the top of platforms such as HDFS, is an excellent illustration of the definitive adoption of Big Data into organizations. This massive volume of data is now available to data scientists, who can extract an added value from it, thanks to multiple data analysis techniques, including those derived from Machine Learning, whose success often depends on the availability of such a large volume of data. Nonetheless, the possibility of performing such analysis depends on the availability of an appropriate infrastructure allowing this kind of exploitation.

Last but not least, the rise of Cloud Computing has enabled many organizations to rationalize their IT infrastructure. Cloud Computing can be seen as the ability to access a pool of resources owned and maintained by a third party via the Internet. It is not a new technology by itself, but a new way of consuming computing resources [47]. In the cloud model, the resources no longer belong to the organization, but they are most often "leased" from one or more providers according to the

organization's needs. Cloud resources are thus perceived as having a low maintenance cost, switching to an on-demand model in which organizations may adapt their consumption according to their needs and only pay for the resources they actually consume. However, the adoption of the Cloud model is often accompanied by some fears related to the outsourcing of data and data processing. These fears concern in particularly security, confidentiality and network latency issues. The choice between deploying a certain service in an internal organizational resource or outsourcing it into a public Cloud resource becomes now as strategic as technical. Consequently, resources are more and more visible and must now be managed from more than just a technical perspective.

Fog Computing, discussed in Chapter 2, has reinforced this point. Fog computing can be seen as a new paradigm for disseminating computing, storage and service management closer to the end user, all along the continuum between the cloud, and objects (IoT) and end devices. Indeed, thanks to Fog Computing platforms, it is possible to consider the use of proximity resources for the execution of certain services. This makes it possible to consider the use of resources other than those located in data centers or in cloud platforms to run services, offering new perspectives for further rationalizing the use of available resources.

All these new technologies and trends are gradually entering into the composition of Information Systems, leading to their evolution. Today, we are observing the emergence of a new generation of IS that could be called pervasive, both by their distribution beyond the organization's boundaries, and by the pervasive nature of the environment they integrate. Thanks to these new technologies and practices, Information System can extend well beyond the physical limits of the organization. They are now accessible everywhere, they include resources both inside and outside the organization, and they can even integrate the physical environment itself. Notions of what is inside or outside an organization have become blurred with processes that use resources other than those within the organization's traditional perimeter [25]. The environment has become more and more heterogeneous, integrating very different devices, which can moreover be mobile, adding dynamism to the heterogeneity. Thus, we have Information Systems that are increasingly confronted with a heterogeneous and dynamic environment, integrating resources and services internal and external to the organization, and even the physical environment surrounding both these resources and the users. We may we expect from these systems more flexibility and a certain "smartness" in order to better carry out the organization's activities and better satisfy user's and organization needs.

A Pervasive IS can thus be seen as an emerging class of IS in which IT is gradually embedded in the physical environment, capable of accommodating users' needs and desires when necessary [96]. The term "Pervasive Information Systems" was introduced by Joel Birnbaum in 1997 [13]. In this article [13], the author considers a technology that becomes pervasive, and thus invisible to the human eyes: "Today's schoolchildren don't think of TVs and telephones as technology-they can't imagine life without them. Tomorrow's children will feel the same way about computers, the networks connecting them, and the services they perform". This corresponds to the "cognitive invisibility" reported by Bell & Dourish [10]. These authors mention a technology that is invisible to us, since we use it continuously without necessarily perceiving it as computers. Birnbaum [13] talks about an information technology that should become intuitively affordable for everyone and that should bring enough added value to justify the necessary investments. Considering the aforementioned evolutions and trends, as well as the opportunities they offer to the organizations, we may say that this point has been reached. And the consequences for IS are not insignificant. Birnbaum [13] emphasizes in particular the expectations with regard to the services offered. For this author, in the same way that people expected (in 1997) to have a dial tone when they picked up a telephone handset, people will (nowadays) wait for useful information to be available and ready for use. To sum up, even if Birnbaum [13] does not precisely define the notion of PIS as Kourouthanassis & Giaglis [96] do, the elements that he enumerates in his article, *i.e.* the technology that becomes "invisible", the importance of services and the added value of information, the paradigm shift with people paying by use, modifying what was before a capital investment in service, etc., characterize quite well what today's information systems are becoming.

Therefore, we are confronted with the emergence of Information Systems that extend beyond the physical (and logical) boundaries of the organization, that integrate new technologies and an environment that has itself become pervasive (in the technologically charged sense) in a more or less transparent way, and from which we expect more intelligent behavior, both reactive and proactive. For Kourouthanassis & Giaglis [96], unlike traditional IS, PIS encompass a more complex and dynamic environment, composed of a multitude of artefacts (and no longer just desktop computers), capable of perceiving the users' context and of managing the mobility of these users. In the literature, the term "mobile" IS [98] is also employed, with the notion of mobility used in a broad sense: spatial, temporal, but also contextual. Krogstie *et al.* [98] refer to systems characterized by their dynamism, by frequent changes of context (spatio-temporal, environmental context, but also relative to users, their tasks and even available information), and thus requiring an important capacity of adaptation from the system to the users. Even if these authors [98] mention in particular the adaptation of interfaces for a better interactivity with users, whatever the terminals they use, it is easy to imagine that this adaptation should be extended to the proposed services and their implementation.

This new generation of Information Systems (the Pervasive Information Systems) can be distinguished from traditional IS by different characteristics, obtained from the analysis of the literature, which can be associated with requirements that must be observed when designing a PIS [112]:

- Heterogeneity: a PIS must support the heterogeneity of devices and technologies that compose up pervasive environments;
- Transparency: a PIS must be transparent, being able to hide from the users the heterogeneity of pervasive environments;
- Context-awareness: any so-called pervasive system must be able to accomplish the requested functionalities, despite changes in the surrounding conditions or in the state of the system [138]. A PIS must be able to perceive its execution environment and to adapt itself accordingly;
- Goal-oriented: A PIS must be designed to meet and satisfy the users' goals in their business activities;
- Predictability: A PIS is expected to meet the user's business goals in a predictable and controlled manner. While it must take advantage of the dynamic environment and the opportunities that such an environment may provide, the behavior of a PIS, with all services and functionalities it offers to its users, must remain predictable, in order to ensure the governance of these systems and the confidence the users would have on them.

Thus, we may refer to PIS as an IS particularly characterized by the heterogeneity and the dynamic nature of the environments and resources involved, but also by their need for adaptation and context awareness. Unlike traditional IS, in which the users have often to adapt themselves to the system, PIS must consider the environment and the context of use in order to adapt itself and to provide users with the service that best corresponds to their needs and current context. These are systems, whose intention would be to increase the productivity of the users (and infrastructures) by providing them with adapted services, has to consider the heterogeneity of the environment, which turns to a pervasive environment. Context awareness becomes thus a key element since it provides the possibility of adaptation at different levels: interaction modes, services and information access, but also the infrastructure itself.

This evolution of IS towards Pervasive IS leads to several challenges, for which new approaches, adapted to the characteristics and to the constraints of this new generation, are necessary. Among these challenges, two retained our attention from 2013-2014: the design and the management of these systems; and the necessary resource management on these systems.

The first challenge that we focus concerns thus the design and the management of such a new generation of Information System, faced with unprecedent heterogeneity and dynamism, but that is always subject to constraints (and practices) specific to an IS. Indeed, a PIS is above all an IS. Its main objective remains to propose to users the necessary functionalities for a smooth and efficient running

of the organization and its processes. These functionalities may assume many different forms, depending on the components and on the available technologies, but also depending on the context in which they are invoked. The heterogeneity of the overall environment, which has become pervasive, leads to more variability. This variability becomes essential for the system to better adapt its behavior to both the organization's and the user's context and goals. The design and managing of such a PIS can then be seen as a complex problem, as it becomes necessary to identify which services should be accessible to users and under what circumstances, while considering the heterogeneity of the environment and of the services involved. PIS can thus be characterized by an increased need for transparency, both for end-users and for the designers themselves, as the complexity of the pervasive environment must be hidden without losing completely (mainly for designers). The question then is how to design such a system while mastering its complexity? Or, more precisely, how to manage the services that need to be offered in order to meet the system's needs, while at the same time guaranteeing the transparency required for these new IS?

We should not forget that we are considering here systems that already exist, that are present and well established in organizations, and that are destined to evolve. It is thus essential to guide this evolution, to move the existing services towards a wider and better adapted offer. The challenge is to make the wide range of services offered by the IS accessible wherever and whenever they may be needed, regardless of the technologies involved.

Unfortunately, only a few abstractions or conceptual tools exist today to help Information System designers (or managers) in this transition from a traditional IS to a Pervasive IS. The IT departments are often left by their own face this challenge. Therefore, the first contribution we present in this chapter tackles this issue, by proposing an abstraction for these systems. This abstraction, known as the "Space of Pervasive Service", is presented as a conceptual tool that can be used to better handle the complexity of a PIS, proposing a first step towards a better management of PIS as a whole.

The second challenge that we are focusing on here concerns the management of resources in a PIS. For years, the notion of resources (in the sense of IT resources) that enables the execution of services has received little attention in modeling and conceptualization of IS. This is mainly due to the fact that the resources available in an IS were mostly stable and homogeneous. This is particularly true when concerning "data centers" and similar structures, in which all services offered by the IS are executed. Consequently, resources were not perceived as something strategic for the IS: whatever the service is, it would be executed over a stable infrastructure. However, the introduction of Cloud Computing, and more recently of Fog Computing, is changing the way these resources are perceived. Moreover, the current trend towards increased use of micro-services in organizations, which advocate for a finer breakdown of functionalities, is enabling applications to be deployed more easily over differ kind of infrastructures. It is now possible, with the help of micro-services, to envisage an opportunistic use of available resources, as supported by [105,169]. All the conditions are thus in place to enable the dynamic deployment of IS services over resources as varied as cloud resources (private or public), traditional data center resources, network devices, IoT, or mobile terminals, in a transparent way. All these developments have transformed the nature of the resources available in Information Systems. These resources have become more distributed, heterogeneous, and organized in an infrastructure that has itself become more dynamic. The placement of services on these resources has thus become a non-trivial problem.

Although resource management and task placement are long-standing issues in distributed systems and in High Performance Computing (HPC), as demonstrated [149,162], the characteristics of the resources involved in a PIS make this task even more complex. The resources in a PIS are similar, in terms of heterogeneity, to those considered by Fog Computing. On the one hand, they include servers running in data centers, but also virtual machines running on Cloud platforms, as well as "micro data centers", server resources that are specially deployed close to users for Fog Computing. In addition, we may have resources such as RaspberryPi and other nanocomputers, often used for and by the IoT, as well as mobile and/or personal use resources such as laptops, desktops, tablets and even

smartphones. Each of these resources can be both a source of information and an execution platform for some services since they propose some processing capabilities.

Therefore, we are confronted to a highly heterogeneous and dynamic pool of resources, since these resources can come and go, become available or disappear at any time, depending on circumstances (*e.g.* whether its owner/user moves or leaves, in case of connectivity or power supply problems, or even in case of an ending contract). Moreover, these resources are not necessarily dedicated to the execution of these services and do not necessarily belong to the organization. They may, for example, belong to partners or collaborators; they may be used for the execution of services specific to the organization but also for private tasks.

The resources available in a PIS can then be characterized by their heterogeneity and the dynamism of the environment, just like the resources considered by Fog Computing. Several authors [52,57,61] point out that these characteristics increase the hardness of resource management as well as the difficulty of placing tasks in these resources, especially when compared to Cloud Computing platforms.

We are particularly concerned by this problem in PIS. We are focusing more specifically on the question of opportunistic use of available resources. Indeed, like any IS, PIS have a large pool of available resources. As these resources may evolve very quickly, the question of their opportunistic use arises, as long as they remain available and provide suitable conditions for the execution of certain services. The last part of this chapter focuses on this question of an opportunistic resource management in PIS, with contributions that follow those proposed on the PER-MARE project, based on the CloudFit platform (cf. section 2.2), but which are still under construction, notably through David Becerra's PhD thesis, started at the end of 2016.

4.1 Space of Pervasive Services

4.1.1 Problem statement

As discussed in the previous section, PIS are characterized by their heterogeneity. This heterogeneity affects both the available services and their implementations as well as the resources used for their execution. The technologies involved are multiple and lead to increased complexity. This complexity makes difficult for both, end-users and those who have to design and manage such systems, to understand and assimilate the system. As Dey [38] has pointed out, when users experience difficulties in establishing a mental model of how applications work, they are less disposed to adopt and use them. A misunderstanding of a PIS and how it works may affect the acceptance of such a system (entirely or of some of its components), and thus compromise its adoption and the transition from a traditional IS to a PIS. Given the strategic role of Information Systems (and therefore PIS) within organizations as a support to their activities, the consequences of not adopting such a system can be very significant.

It is important for PIS users to understand these systems it without having to know or understand the technologies involved on those. Similarly, designers need to be familiar with PIS and its functionality, without necessarily being encumbered by the details concerning the involved technologies. In other words, transparency is the key to mastering the complexity of a PIS. Transparency is necessary to hide the heterogeneity that characterizes these systems and that affects their resources, infrastructure, services and uses. This transparency is necessary for both: in order to enable users to focus only on the tasks they have to perform and not on the technologies behind those tasks; and in order to make it easier for designers to think about the services that might be offered and under which circumstances, without also having to focus, at a first moment, on the technologies needed for those services.

In order to achieve this level of transparency, abstractions are needed to enable the representation of these systems. The Space of Pervasive Service proposal represents a first step in this direction, presenting a conceptual tool allowing an abstract representation of a PIS with the functionalities it is

supposed to fulfill and the resources considered to enable their execution. In its first version (published in 2013-2014), the abstraction of Space of Pervasive Service allowed to represent, in an abstract way, the notions of “services” and “sensors”, to which the notion of “resource” was added in 2019. Indeed, each of these notions were used to represent a role that an entity composing a PIS may assume. Even if a given entity may assume several roles, the fact of pulling a part each role allows to simplify the analysis through simple yet powerful abstractions. The Annex X presents the paper published in RCIS 2014 [112] which details the first version of the Space of Pervasive Service abstraction.

- **RCIS 2014** [112]: Najar, S.; Kirsch Pinheiro, M.; Le Grand, B. & Souveyet, C., “A user-centric vision of service-oriented Pervasive Information Systems”, *8th International Conference on Research Challenges in Information Science (RCIS 2014)*, IEEE, **2014**, 359-370

4.1.2 Bibliometric

As for the previous chapters, the work described in this section has produced a few publications. These have been analyzed, using the scholar.google.com website, in relation to the number of citations. Table 8 details the results of such analysis. These publications are mostly recent and target mainly the French community, which explains the limited number of citations to these papers. Another factor contributing to this number is the lack of a widely accepted terminology for Pervasive Information Systems. This term is not yet widely adopted by all communities dealing with Information Systems, other terms are also used. For example, Hauser *et al.* [58] and Schreiber *et al.* [145], as well as Kourouthanassis *et al.* [96, 97] and Birnbaum [13] speak of “Pervasive Information System”, Bell [9] and Maass & Varshney [101] refer to “Ubiquitous Information System”, while Neumann *et al.* [121] use “Evolutionary Business Information Systems”. The first two terms remain the most used (more than 900 references for each according to the site scholar.google.com), the latter being largely in the minority (barely 34 mentions according to the same site). In terms of community, the first one, “Pervasive Information System”, seems to be the most used in the field of Computer Science, while the others seem to be more used by researches in Management Sciences. This topic remains a recent and relatively restricted topic in the IS community, for which there is not yet a real consensus in the community. It is a subject that will certainly evolve, in our opinion, in the coming years.

- **INFORSID 2013** [81] **:** Kirsch Pinheiro, M.; Le Grand, B.; Souveyet, C. & Najar, S., « Espace de Services : Vers une formalisation des Systèmes d'Information Pervasifs », *XXXIème Congrès INFORSID 2013 : Informatique des Organisations et Systèmes d'Information et de Décision,* **2013**, 215-223
- **UbiMob 2013** [113] **:** Najar, S.; Kirsch Pinheiro, M.; Le Grand, B. & Souveyet, C., « Systèmes d'Information Pervasifs et Espaces de Services : Définition d'un cadre conceptuel ». *UbiMob 2013 : 9èmes journées francophones Mobilité et Ubiquité,* Jun **2013**, Nancy, France. Disponible sur https://ubimob2013.sciencesconf.org/19119.html (Last visit: août 2020)
- **RCIS 2014** [112] : Najar, S.; Kirsch Pinheiro, M.; Le Grand, B. & Souveyet, C., “A user-centric vision of service-oriented Pervasive Information Systems”, *8th International Conference on Research Challenges in Information Science (RCIS 2014)*, IEEE, **2014**, 359-370
- **INFORSID 2019** [83] : Kirsch-Pinheiro, M. & Souveyet, C., « Le Rôle des Ressources dans l'Evolution des Systèmes d'Information », *Actes du XXXVIIème Congrès INFORSID (INFORSID 2019),* Paris, France, Juin 11-14 **2019**, 85-97
- **Atelier 2019** [150] : Souveyet, C.; Villari, M.; Steffenel, L. A. & Kirsch-Pinheiro, M., « Une approche basée sur les MicroÉléments pour l'Évolution des Systèmes d'Information », *Atelier Évolution des SI : vers des SI Pervasifs ?, INFORSID 2019,* **2019.** Disponible sur https://evolution-si.sciencesconf.org/data/book_evolution_si_fr.pdf (Last visit: août 2020)

Table 8. Bibliometric analysis concerning the Space of Pervasive Services proposal

Reference	Year	Total	≤ 2016	> 2016	Self-citations
INFORSID 2013	2013	1			1
UbiMob 2013	2013	2	1	1	
RCIS 2014	2014	3		2	1
INFORSID 2019	2019				
Atelier 2019	2019				
Total / %		**6**	**16,67 %**	**50,00 %**	**33,33 %**

4.2 Opportunistic resource management on Pervasive Information Systems

4.2.1 Problem statement

As discussed at the beginning of this chapter, PIS are characterized by the heterogeneity and the dynamism of their environment. Like the environments considered by Fog Computing, the environments involved in a PIS contain a varied and variable set of resources, which may include resources ranging from high-performance servers and virtual machines in the cloud to tablets and nanocomputers for the IoT. Many resources are already available and integrated into this environment. They are not necessary placed in a data center. These resources can be disseminated all along the organization, and even beyond, and they can be dedicated (or not) to different uses. Indeed, these resources are not always dedicated exclusively to certain services or tasks, and even if they are not always very powerful, they still offer significant power computing. Unfortunately, except for data center and cloud resources, many of the resources available in a PIS are often underused.

Many of these resources can be available only on a temporary basis, while others can be available in a more permanent way. These resources are very heterogeneous, they can be mobile and have very distinct characteristics. Their computing capacities can also vary over the time, since they are not necessarily resources dedicated to a specific task, and can therefore carry out different tasks simultaneously, which may affect their available capacities.

Considering the availability of all these resources, it is possible to envision the use of proximity resources for the execution of services on behalf of the PIS, in a similar way that resources in Fog platforms. However, this use would require a resource management that is opportunistic, since it should be guided by the availability of these resources and their available capacities. The objective is not necessarily to optimize the use of a fixed pool of resources, as it is often the case in data centers or in Cloud platforms [149, 162], but it is to try to make an effective use of a pool of heterogeneous resources, whose composition and available capacity can vary over the time and whose use is not dedicated to the execution of the considered services. Similar to resource management on Fog platforms [52,61], managing resources in such a pool is a complex problem, especially when compared to Cloud platforms.

In this section, we tackle this issue, focusing on an opportunistic use of resources in a PIS, which means using such resources for service execution while they are available, according the capacity they offer at a given moment. In our view, a full deployment of Pervasive Information System will require a context-aware management of available resources, which should consider particularly the heterogeneity and the dynamism of PIS environment. This research addresses this issue, focusing notably on the placement of services within the available resources and on the architecture required to meet PIS requirements. In this way, we pursue the vision initiated by the Space of Pervasive Services (see section 4), by focusing more on this specific point of the architectural view, which is necessary for the executability of the defined space of services.

The work related to the PER-MARE project (cf. section 2.2), carried out between 2013 and 2016, has revealed new perspectives for us concerning the resource management in Fog Computing environments. Since PIS are characterized by the same heterogeneity and dynamism as these environments (Fog Computing being part of the ecosystem fomenting the emergence of PIS), it appeared logical to continue our research on this topic beyond the PER-MARE project (which ended in 2014). Since 2016, we have continued our researches on this topic, using the CloudFIT platform proposed during PER-MARE project. This work highlights an observation, also reported by nowadays literature, concerning the difficulties of placing tasks in a Fog Computing environment [16,42,61,99,146] for which PIS are also concerned. Since 2016, we started to further explore the impact of heterogeneity in the execution of tasks in this type of environment, in order to better understand its effect in the case of a PIS, which is also characterized by this heterogeneity.

Thus, we have studied, on the one hand, the effects of heterogeneity on the execution of tasks in these environments, and on the other hand, the requirements and characteristics for resource management in general and on PIS. To do this, we have first carried out a set of experiments in a real environment (not in a simulator) using the CloudFIT platform, which complemented the results of the PER-MARE project (see section 2.2). Then we carried out a literature review on resource management, both in "traditional" computing environments (HPC and Cloud Computing) and in Fog Computing environments. The results of these two works led to the definition of a conceptual architecture for resource management in a PIS, which is part of David Beserra's PhD thesis (thesis still in progress).

Several lessons could then be learned from these experiments. First of all, it is possible to obtain, by observing context information, a reasonable use of available resources, which would avoid over-consumption of low-powered resources, while still taking advantage of their computing power. Experiments have also shown that the placement of the services consuming and/or producing data can be highly influenced by the placement of these data and by the volume of involved data, which is highly dependent on the application. Different strategies for service placement are thus possible and should coexist in a PIS. Furthermore, a totally distributed architecture, without any central elements, such as proposed by the CloudFIT platform, has proved to be particularly interesting for managing dynamic environments and for managing the scalability required to cover a PIS. Nevertheless, this kind of architecture comes at a price: it is certainly less sensitive to failures and provides better scalability than architectures that depend on a central element (a server, a broker or a proxy), as Ghobaei-Arani *et al.* [52] have pointed out, but it remains sensitive to network partitioning. In these cases, the tasks are still performed, but the risk incurred is a certain waste of resources.

All this demonstrates the complexity that can surround opportunistic resource management in a PIS. Our experiments suggest that this management is influenced by multiple factors involving resources and their execution context, but also applications (services). Multiple strategies remain possible, which leads us to believe that a configurable policy system, considering different factors, is necessary. These factors can go beyond the purely technical aspects. For example, although the use of cloud servers for storing large volumes of data may be technically attractive, it may not be suitable for some organizations, depending on the cost of the cloud itself or on data privacy. Conversely, even if it is possible to use locally available resources to run a service, this might be discouraged, depending, for example, on the resource ownership (e.g. resource belonging to another organization or for personal

use only). The ability to define resource management policies in a flexible and configurable way is therefore a necessity for PIS.

Based on this study, we may consider resource management in a PIS as a special case of resource management in which we have:

i. A heterogeneous and dynamic resource pool;
ii. A pool that can belong to different owners;
iii. An on-demand service execution scheduling (with service requests that cannot be necessarily anticipated);

And for which it is also required:

iv. A service placement policy that does not necessarily aim at optimizing resources, but rather at an opportunistic use of these resources;
v. Criteria (or metrics) to be observed that can vary according to the services requested; the same for the decision-making criteria that can vary according to the resources;
vi. Significant scalability and evolutivity, given the dimensions that a PIS can take on.

The first two points (*i* and *ii*) can be seen as a consequence of the way in which PIS are built: through the introduction of different technologies and new practices (or partnerships), which complement those already in place and add complexity to the resulting environment, while bringing new opportunities (*e.g.* new services, new business processes, etc.) to these systems. The third point corresponds to the need of providing these systems with greater flexibility, thanks to services that can better adapted to the demand. The fourth point is a consequence of the dynamic environment of PIS, which is more propitious to an opportunistic use of available resources than to an “optimal” use of those. The objective is not necessarily to optimize the use of resources (*e.g.* load balancing among resources, or minimizing the number of resources used), neither optimizing pure performance indicators (*e.g.* minimizing execution time, or maximizing compliance with SLAs), but above all to be able to use resources when they are available and thus avoid a certain "waste" of underused resources inside the organization. To do this, different information can be observed and used for decision making, since different strategies can be considered, depending on the service requested, and on the resources considered for its execution. This variability on the criteria that could be used leads to the need for configurable policies, both for services and resources, which should consider context information among these criteria, with a real context management (and especially Quality of Context concerns) associated with it.

Finally, the last point considers the need for evolutivity and scalability that can characterize a PIS. When talking about an IS, it is already possible to envisage a very large number of resources and services. The dynamic nature of an PIS only reinforces this point and emphasize the importance of considering scalability as a key aspect: a PIS must be able to evolve in response to changes in its environment, but also to changes in business strategy that may be decided at a business level. Even if PIS resources are organized into multiple spaces of services, scale remains an important factor. As is evolutivity: criteria that are important today in the management of certain resources (or services) may change and be replaced by new criteria in the future, as the organization itself evolves.

All of these considerations point to four major characteristics for an opportunistic resource management in a PIS: *dynamism, contextual awareness, flexibility, and scalability*. Only a few studies in the literature consider all these aspects. Taken together, these points considerably distinguish resource management in a PIS from resource management in Cloud Computing, or even Fog Computing platforms.

These points have inspired the definition of a conceptual architecture for an opportunistic resource management in a PIS. The definition of these components is accompanied by the definition of an expected behavior of each component and their interactions. This dynamic behavior establishes is

supposed to ensure, among other things, a distributed decision making between the different resources. This decision making is based on policies, which can be divided into several categories: those applicable to services and those applicable to resources, but also policies defined for a particular resource (called “local” policies) and those defined for an entire organization (called “global” policies). All of these policies may be based on a variety of information, including context information. We are therefore approaching the definitions gave by the Space of Pervasive Service abstraction, with notably constraints that apply to a resource or to a service. On the basis of these policies and the observed context information, three decisions can be made: immediate execution of a service, its delegation to a neighboring resource, or putting it on hold. Each time a service is delegated or put on hold, its priority is increased to ensure that it will be executed in a near future. The idea would be to execute a service that is considered a priority, even if some policies are not all met.

This architecture is still under definition as part of David Beserra’s PhD thesis. We are currently working on the definition of policies and on a formalism allowing their expression. It is important that these policies can express constraints on the use of resources, but also in relation to services, following the same principles established by the Space of Pervasive Services (cf. section 4.1). Thus, the same notions of required context, constraints and properties, as defined on [83], are explored on the policies definition. The scheduler becomes a key element of such architecture, resolving these definitions in such a way as that the resource will remain in a state considered acceptable, while executing services on behalf of the PIS.

This research represents an ongoing work for which only a few publications are available. Among those, the paper published at EUSPN 2018 [157], included in the Annex XI, contains interesting insights about how heterogeneity and dynamicity of the environment could impact resource management of such environments.

- Steffenel, L.A. & Kirsch-Pinheiro, M., "Improving Data Locality in P2P-based Fog Computing Platforms", *9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2018)*, Leuven, Belgium, November 5-8, **2018**. DOI: doi:10.1016/j.procs.2018.10.151

4.2.2 Bibliometric

As a work in progress, this contribution has been the subject of only a very few publications. Some results, still considered as preliminaries, could not be submitted for publication yet. We can only cite here the publications involving experiments carried out with the CloudFIT platform. As with the definition of the Space of Pervasive Service (cf. section 4.1), these publications aimed at primarily the French community, in order to gather from this community a feedback that we consider was mandatory for the development of the proposal. These publications are listed below and their impact, in relation to the number of citations, is summarized in Table 9. As expected, a very small number of citations were identified, without prejudging the impact that this work may have on the Information System community in the future.

- **UbiMob 2016** [159]: Steffenel, L.A. & Kirsch-Pinheiro, M., « Stratégies Multi-Échelle pour les Environnements Pervasifs et l’Internet des Objets ». *11èmes Journées Francophones Mobilité et Ubiquité (Ubimob 2016)*, 5 juillet **2016**, Lorient, France. Paper n°6. Disponible sur https://ubimob2016.telecom-sudparis.eu/files/2016/07/Ubimob_2016_paper_6.pdf (Dernière visite: aout 2020)

- **EUSPN 2018** [157]**:** Steffenel, L.A. & Kirsch-Pinheiro, M., "Improving Data Locality in P2P-based Fog Computing Platforms", *9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2018)*, Leuven, Belgium, November 5-8, **2018**. DOI: doi:10.1016/j.procs.2018.10.151
- **IJITSA 2018** [158]**:** Steffenel, L.A.; Kirsch-Pinheiro, M.; Vaz Peres, L. & Kirsch Pinheiro, D. "Strategies to implement Edge Computing in a P2P Pervasive Grid", *International Journal of Information Technologies and Systems Approach (IJITSA)*, IGI Global, 11(1), **2018**, 1-15. DOI: doi:10.4018/IJITSA/2018010101
- **COMPASS 2019** [156]: Steffenel, L.A. & Kirsch-Pinheiro, M., « Accès aux Données dans le Fog Computing : le cas des dispositifs de proximité », *Conférence d'informatique en Parallélisme, Architecture et Système (ComPAS'19)*, 25-28 July 2019, Anglet, France. Disponible sur https://hal.univ-reims.fr/hal-02174708 (Last visit: aout 2020)

Table 9.Bibliometric analysis of citations related to our research about resource management on PIS.

Reference	Year	Total	≤ 2018	> 2018	Self-citation
UbiMob 2016	2016				
EUSPN 2018	2018	2		2	
IJITSA 2018	2018	2		2	
COMPASS 2019	2019				
Total / %		**4**	**0 %**	**100 %**	**0 %**

4.3 Chapter summary

Unlike the previous chapters, the contributions presented here are still under development. The concept of the Space of Pervasive Services, even if it was the subject of a few publications, is still evolving. Several aspects related to this concept still require attention, notably the design methodology, which needs to be extended in order to take into account the resources definition.

Executability of such spaces remains also an open issue. Our research on an opportunistic resource management is a first step towards this direction. This work began with an empirical phase, based on the experiments we have performed using the CloudFIT platform. This empirical phase was followed by a literature review considering scheduling mechanisms and policies on Cloud and Fog Computing. These first phases allowed us to acquire the knowledge necessary to move towards the definition of a conceptual architecture for an opportunistic resource management on SIP. This architecture is still under construction as part of David Beserra's PhD thesis. We expect that this work will represent a basic foundation for the construction of a true execution platform for PIS.

Several issues still have to addressed in order to achieve this vision of Space of Pervasive Service at runtime. Not only the management of resources in dynamic environments needs to be considered, but also the execution business processes using the proposed services. The collaboration started with the Università di Messina is part of this context. The use of micro-containers [150] to encapsulate and organize services and sensors in process segments represents a first steps, which must be further developed in a near future.

The preliminary nature of the researches presented in this chapter does not allow us to propose a clear evaluation of the impact these researches may have, particularly in the IS community. Nevertheless, these researches still represent a first step towards a fully definition of PIS. Through the challenges considered in this chapter, we wanted to emphasize the importance of establishing a conceptual vision of these systems, but also of improving the coverage of the technical aspects necessary to achieve our vision of PIS. This work also demonstrates the highly multidisciplinary nature of PIS. The development of these systems brings several challenges for which skills from the different communities of Computer Science (and event beyond) will be necessary. This multidisciplinary can be illustrated by our research on opportunistic resource management, for which skills from the HPC and Cloud Computing communities, and more generally from Distributed Systems, have had to be mobilized.

The multidisciplinary of Pervasive Information Systems is perhaps the most striking feature of these systems. We strongly believe that the challenges that emergence together with these systems will only be addressed by a global and multidisciplinary approach.

IV Conclusion & Perspectives

All along this document, different contributions have been discussed. These range from 2002, when my PhD thesis began, to the present days. All these contributions have in common the notion of context, which has been applied to different communities of Computer Science.

Indeed, I have started my research in the CSCW (Computer Supported Cooperative Work) community by applying the notion of context to the adaptation of group awareness information (chapter 1). The main outcomes of this research were an object-oriented context model and a filtering process whose principles inspired other works years later. Then, chapter 2 presents my research works regarding pervasive environments. These researches have targeted directly the Ubiquitous Computing community. Here again, the notion of context was used, first in a peer-to-peer distribution mechanism for context information, and then in the use of context information for resource management during the PER-MARE project. This project was one of the precursors in exploring the use of Fog Computing for Big Data applications. This work opened up several research perspectives considering an opportunistic use of available resources thanks to the observation of the context in which these resources are employed.

In chapter 3, my research works integrating the SOC (Service Oriented Computing) community were presented. Most of these works are the result of my integration into the laboratory "Centre de Recherche en Informatique" (CRI), at the University of Paris 1 Panthéon Sorbonne. This research work can be characterized by the combination of the notion of intention, coming from Requirement Engineering community, to the notion of context in the service orientation. This triplet "< *intention, context, service* >" has been used as a basis for my contributions on service discovery and prediction. This work has allowed me to consider from a new perspective the question of the user's habits and practices. It also tackles the question of the relevance of context information: how to recognize whether a given context element may characterizes or not the choice (and then the use) of a given application or service by a user? My work on context mining brings some insights concerning this issue, thanks to the application of FCA (Formal Concepts Analysis) in a continuous improvement process. This work has also opened the perspective to a deeper reflection about the applicability of Machine Learning techniques to context data in a very large scale, in what we call a "context facility".

All these contributions converge towards the notion of Pervasive Information Systems (PIS), which is the subject of my latest researches, presented in chapter 4. These systems represent the new generation of Information Systems (IS). It is not a matter of new systems that would be created from scratch, but rather the evolution of existing systems, which are now being overturned by the arrival of new technologies and new practices. All these evolutions (Fog Computing, IoT, Big Data, Machine Learning, etc.) are driving these systems well beyond the boundaries traditionally accepted (and handled) by the IS still in place today. Pervasive Information Systems go well beyond the limits of the organization, integrating the physical environment, mobile technologies, Fog and Cloud Computing. Even if the data represents an important concern on these systems, notably thanks to IoT and Big Data related technologies, this evolution cannot be reduced to the availability data everywhere. It is not only a matter of data, it is also about a whole Information Systems that can be deployed everywhere, available all the time. In short, it is about the Weiser's [171] vision of Ubiquitous Computing becoming reality over current Information Systems.

At the heart of all these evolutions leading current Information Systems into PIS, there is an environment that becomes eminently dynamic and that must be mastered. This dynamism brings the promise of more flexibility for Information Systems, which could be able to easily (or more easily) adapt themselves to changes. The notion of context can thus play a key role in this transformation from so-called "traditional" IS to Pervasive Information Systems. The aim is to allow more adaptability to these systems at every level, from infrastructure to management, including event the business support

functions. We consider here context information in a broader sense, resulting from the observation of the users, the physical environment, as well as the organization itself, as considered in my PhD thesis work (chapter 1). Each level of a PIS can thus benefit from context information for its own adaptation, as illustrated by the contributions discussed in this document. Each at its own level, these contributions suggest that it is actually possible to bring more reactivity to infrastructures, services and applications by considering the context information.

However, the real challenge does not lie in adapting each level separately, in an independent way, but in creating a real synergy between the IS levels. Each level should be able to adapt itself according to its own conditions and goals, but also according to observed context information and changes coming from the neighboring levels. It is an entire dynamic between the different levels of an Information System that can be obtained from a global management of the notion of context.

Even if context information is seen here essentially as a trigger for adaptation purposes, it is not a question of automating everything in a PIS. A PIS is an Information System that evolves, and the very nature of these systems requires them to be predictable and manageable. One must be able to control an IS, its applications, processes, services, infrastructure, etc., in any situation. We must be able to manage a PIS despite the heterogeneity and the dynamism of the involved environment. Adaptation within a PIS can be led automatically, but it can also come from an active management from decision makers. Our research works on group awareness (chapter 1) and context mining (chapter 3) suggest the potential of context information for decision making. Context information can then become the cornerstone of a continuous improvement process that will undoubtedly be essential to the emergence and survival of future PIS.

All of this leads almost inevitably to the generalization of the context support to the entire Information System. This means considering this context support as a “facility” available to all Information System components. This idea, introduced in [11] (chapter 3), represents, in my opinion, the keystone of Pervasive Information Systems. If we see a PIS as a city, context management should be like a “facility” integrated into the city, such as water or electricity, a service available to all members of the community. Thinking of context management as a “context facility” implies generalizing this notion to the entire system. Everything may become observable. Each element in a PIS could thus be observed, become a source of context information, and at the same time, a consumer of this kind of information for different uses, from adaptation to decision-making.

This vision of a “context facility”, available for an entire PIS, raises several challenges, particularly related to the scale that this vision implies: it is no longer a particular application or service that benefits from such a platform, but potentially all the elements of a PIS, whatever their level. This can be illustrated by group awareness information. On the one hand, this information may be considered as organizational context in order to be better exploited in groupware applications, which will no longer have to deal with the management of this information, on the other hand, it may also be considered as a condition for the using certain services or resources, as the required context defined on [118] (chapter 3). Context information is no longer captured for a specific use, but for many different uses, even future ones. This raises the question of how to model this information, how to store it, but also how to process it on a very large scale.

Through this idea of a “*context facility*”, everything becomes a possible source of context information. This information can thus be fed back to all levels of a PIS and launch reactions on these levels, from adaptation to decision-making, reactions that may, in their turn, trigger new changes. It is through the notion of context as a “facility” that the synergy between all levels of a PIS can be created. Figure 13 illustrates this idea of a synergy among all PIS levels. In the lower part of the picture, we may observe different elements acting as a source of context information and feeding this “facility”, which in its turn makes this information available at all levels of the system. At each level, this information can trigger changes, which will feed back to this base new information, improving thus a dynamic interaction among PIS levels.

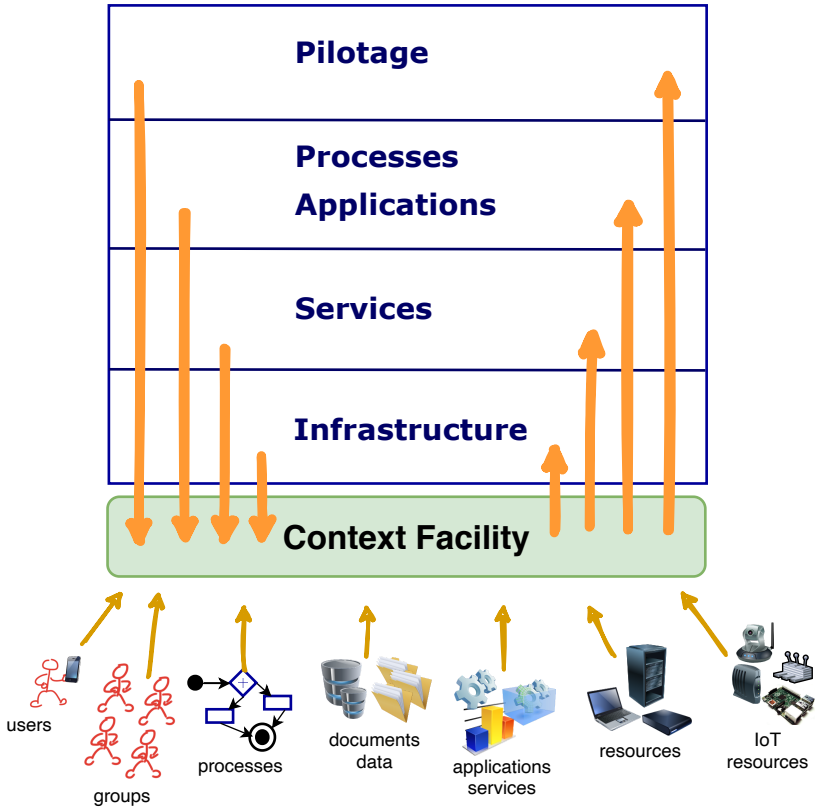


Figure 13. Context facility on a PIS.

This synergy could provide Pervasive Information Systems with the flexibility they will need to better take advantage of the opportunities offered by the new technologies to come and by the environment's own dynamics, whether physical, logical or organizational. Nevertheless, the challenges to make this vision a reality are numerous and particularly important. First of all, there are the challenges that can be described as technical, including the definition and the development of the platforms and methods that will be necessary for implementing this vision. The development of a platform that could be capable of executing the Space of Pervasive Services, mentioned in Chapter 4, or the definition of a platform capable of scaling up Machine Learning techniques, as discussed in [11] (Chapter 3), may be cited as examples of these technical challenges. However, it is worth noting that these challenges mentioned in this document are far from being the only ones. Questions concerning data and infrastructure security, or the robustness of these highly dynamic environments are just starting to be felt now in today's organizations.

These technical challenges are accompanied by methodological challenges, including the development of models and methodologies for managing and steering PISs. The definition of the Spaces of Pervasive Services theory (Chapter 4) represents a first step in this direction, which needs to be further explored in the next years.

Finally, PIS and this vision of a "context facility" also raise questions on a social and human acceptance levels. Are we ready to accept such a high level of observation of our daily business life? Will we be able to accept the increasing automation of our work environment? Like any new technology, like any change, all these upheavals bring with them hope and fear, which we, as a society, should learn to balance. These latter challenges are largely beyond my current areas of expertise, but I am curious to know where the future will take us.

V References

1. Adomavicius G.& Tuzhilin A., "Context-Aware Recommender Systems". In: Ricci F., Rokach L., Shapira B., Kantor P. (eds), *Recommender Systems Handbook*. Springer, **2011**, 217-253
2. Alonso, G.; Casati, F.; Kuno, H. & Machiraju, V., "*Web Services: Concepts, Architectures and Applications*". Springer, **2004**, p. 354
3. Alrawais, A.; Alhothaily, A.; Hu, C. & Cheng, X. "Fog Computing for the Internet of Things: Security and Privacy Issues". *IEEE Internet Computing*, 21(2), **2017**, 34-42. https://doi.org/10.1109/MIC.2017.37
4. Andriole, S.J. & Bojanova, I., "Optimizing operational and strategic IT", *IEEE IT Professional,* 16(5), September/October **2014**, 12–15
5. Ausiello, G., « Cooperative data stream analysis and processing », In : Ferscha, A. *(Ed.)*, *Pervasive Adaptation: Next generation pervasive computing research agenda,* **2011**, 15. Disponible sur https://www.pervasive.jku.at/rab/pdf/PerAda%20Research%20Agenda.pdf (Last visit: Sept. 2019)
6. Baldauf, M.; Dustdar, S. & Rosenberg, F., "A survey on context-aware systems", *International Journal of Ad Hoc and Ubiquitous Computing*, 2 (4), **2007**, 263–277
7. Bauer, C. & Dey, A., "Considering context in the design of intelligent systems: Current practices and suggestions for improvement", *Journal of Systems and Software*, 112, **2016**, Elsevier, 26-47
8. Bauer, J. S.; Newman, M. W. & Kientz, J. A., "Thinking About Context: Design Practices for Information Architecture with Context-Aware Systems", *iConference 2014 Proceedings*, **2014**, 398–411. DOI: 10.9776/14116.
9. Bell, D., "Ubiquitous Information Systems (UBIS): A design research study of intelligent middleware and architecture", *UK Academy for Information Systems Conference Proceedings 2009 (UKAIS 2009)*. Article n° 12, **2009**. Disponible sur : https://aisel.aisnet.org/ukais2009/12 (Last visit: août 2020)
10. Bell, G. & Dourish, P., "Yesterday's tomorrows: notes on ubiquitous computing's dominant vision", *Personal and Ubiquitous Computing*, 11 (2), Jan. **2007**, 133-143
11. Ben Rabah, N.; Kirsch Pinheiro, M.; Le Grand, B.; Jaffal, A. & Souveyet, C., "Machine Learning for a Context Mining Facility", *16th Workshop on Context and Activity Modeling and Recognition, 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, **2020**, 678-684
12. Bettini, C.; Brdiczka, O.; Henricksen, K.; Indulska, J.; Nicklas, D.; Ranganathan, A. & Riboni, D., "A survey of context modelling and reasoning techniques", *Pervasive and Mobile Computing*, 6(2), Apr **2010**, 161-180, **2010**.
13. Birnbaum, J., "Pervasive information systems", *Communications of the ACM*, 40(2), Feb. **1997**, 40–41. DOI: 10.1145/253671.253695
14. Bonomi, F.; Milito, R.; Zhu, J. & Addepalli, S. "Fog computing and its role in the internet of things". *Proceedings of the 1st MCC Workshop on Mobile Cloud Computing (MCC '12)*, ACM, **2012**, 13–16
15. Bouthier, C., « Mise en contexte de la conscience de groupe : adaptation et visualisation », Thèse de Doctorat, Institut National Polytechnique de Lorraine, Nancy, France, **2004.**
16. Breitbach, M.; Schäfer, D.; Edinger, J. & Becker, C., "Context-Aware Data and Task Placement in Edge Computing Environments", *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom 2019),* **2019**, 1-10
17. Brézillon, J. & Brézillon, P., "Context Modeling: Context as a Dressing of a Focus". In: Kokinov, B.; Richardson, D.; Roth-Berghofer, T. & Vieu, L. (Eds.), *Modeling and Using Context*, Springer, LNCS 4635, p. 136-149, 2007.
18. Brézillon, P., "Context-Based Development of Experience Bases". In: Brézillon, P.; Blackburn, P. & Dapoigny, R. (Eds.), *8th Int. and Interdisciplinary Conf. on Modeling and Using Context (CONTEXT)*, p. 87-100, 2013.
19. Brusilovsky, P. & Maybury, M. T., "From adaptive hypermedia to the adaptive Web", *Communication of ACM*, ACM, 45 (5), May **2002**, 30-33

20. Burkhart, T.; Weis, B.; Werth, D. & Loos, P., “Towards Process-Oriented Recommender Capabilities in Flexible Process Environments--State of the Art”, *45th Hawaii International Conference on System Sciences*, **2012**, 4386-4395
21. Carrillo-Ramos, A.; Kirsch Pinheiro, M.; Villanova-Oliver, M.; Gensel, J. & Berbers, Y., “Collaborating agents for adaptation to mobile users”, In: Chevalier, M.; Soule-Dupuy, C. & Julien, C. (Eds.), *Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling*, IGI Global, **2009**, 250-276
22. Cassales, G.W.; Charao, A.; Kirsch-Pinheiro, M.; Souveyet, C. & Steffenel, L.A., "Context-Aware Scheduling for Apache Hadoop over Pervasive Environments", *The 6th International Conference on Ambient Systems, Networks and Technologies (ANT 2015)*, London, UK, June 2 - 5, 2015. *Procedia Computer Science*, vol. 52, Jun **2015**, Elsevier, 202–209. doi: 10.1016/j.procs.2015.05.058.
23. Cassales, G.W.; Charão, A.S.; Kirsch-Pinheiro, M.; Souveyet, C. & Steffenel, L.A. “Bringing Context to Apache Hadoop”, In: Jaime Lloret Mauri, Christoph Steup & Sönke Knoch (Eds.), *8th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2014)*, August 24 - 28, **2014,** Rome, Italy, ISBN: 978-1-61208-353-7, IARIA, 252-258.
24. Cassales, G.W.; Charão, A.S.; Kirsch-Pinheiro, M.; Souveyet, C. & Steffenel, L.-A. “Improving the performance of Apache Hadoop on pervasive environments through context-aware scheduling”, *Journal of Ambient Intelligence and Humanized Computing*, 7(3), **2016**, 333-345.
25. Castro-Leon, E., “Consumerization in the IT service ecosystem”, *IEEE IT Professional*, 16(5), September/October **2014**, 20–27
26. Chaari, T.; Laforest, F. & Celentano, A., “Adaptation in context-aware pervasive information systems: the SECAS project”, *Journal of Pervasive Computing and Communications*, vol. 3-4, 2007.
27. Chang, J.M.; Ho, P.-C. & Chang, T.-C., “Securing BYOD”, *IEEE IT Professional*, 16(5), September/October **2014**, 9-11
28. Chen, N.; Yang, Y.; Zhang, T.; Zhou, M.; Luo, X. & Zao, J. K. "Fog as a Service Technology", *IEEE Communications Magazine*, 56(11), November **2018**, 95-101
29. Chen, S.; Zhang, T. & Shi, W., “Fog computing”, *IEEE Internet Computing*, 21(2), March **2017**, 4 – 6
30. Colman, A.; Hussein, M.; Han J. & Kapuruge, M., “Context Aware and Adaptive Systems”. In: Brézillon, P. & Gonzalez, A. J. (Eds.), *Context in Computing*, Springer, **2014**, 63-82.
31. Coronato, A. & Pietro, G. D., “MiPeG: A middleware infrastructure for pervasive grids”, *Future Generation Computer Systems*, 24 (1), **2008**, 17-29
32. Coutaz, J.; Crowley, J.; Dobson, S. & Garlan, D., “Context is the key”, *Communications of the ACM*, 48 (3), **2005**, 49-53.
33. Da, K. « Plateforme d'adaptation autonomique contextuelle à base de connaissances », PhD thesis, Université de Pau et des Pays de l’Adour, Septembre **2014**
34. Da, K.; Dalmau, M. & Roose, « Kalimucho : Plateforme de supervision d'applications sensibles au contexte », Exposito, E. & Zouari, M. (Eds.), *6ème Conférence francophone sur les Architectures Logicielles (CAL 2012), Revue des Nouvelles Technologies de l'Information RNTI* L-7, **2012**.
35. Degas, A.; Arcangeli, J.-P.; Trouilhet, S.; Calvary, G.; Coutaz, J.; Lavirotte, S. & Tigli, J.-Y. “Opportunistic Composition of Human-Computer Interactions in Ambient Spaces”, *Workshop on Smart and Sustainable City, The Smart World Congress*, **2016**.
36. Devlic, A.; Reichle, R.; Wagner, M.; Kirsch Pinheiro, M.; Vanrompay, Y.; Berbers, Y. & Valla, M., “Context inference of users' social relationships and distributed policy management”, *6th IEEE Workshop on Context Modeling and Reasoning (CoMoRea), 7th IEEE International Conference on Pervasive Computing and Communication (PerCom'09*), Galveston, Texas, 13 March **2009**. DOI : 10.1109/PERCOM.2009.4912890
37. Dey, A. K., “Understanding and using context”, *Personal and Ubiquitous Computing*, 5(1), **2001**, 4-7.
38. Dey, A.K., “Intelligibility in ubiquitous computing systems”. In : Ferscha, A., *Pervasive Adaptation: Next generation pervasive computing research agenda*, 68-69, **2011**. Disponible sur https://www.pervasive.jku.at/Conferences/fet11/RAB.pdf (dernière visite: août 2020).

39. Dorn, C.; Burkhart, T.; Werth, D. & Dustdar S., “Self-adjusting Recommendations for People-Driven Ad-Hoc Processes”. In: Hull, R., Mendling, J. & Tai, S. (eds), *Business Process Management. BPM 2010*. Lecture Notes in Computer Science, vol 6336, **2010**, Springer, 327-342
40. Dourish, P. & Bellotti, V., “Awareness and Coordination in Shared Workspaces”, *Proceedings of ACM Conference on Computer-Supported Cooperative Work (CSCW 92)*, ACM Press, **1992**, 107-114
41. Earley, S.; Harmon, R.; Lee, M.R. & Mithas, S., “From BYOD to BYOA, Phishing, and Botnets”. *IEEE IT Professional*, 16(5), September/October **2014**, 16–18
42. Edinger, J.; Schäfer, D.; Krupitzer, C.; Raychoudhury, V. & Becker, C., “Fault-avoidance strategies for context-aware schedulers in pervasive computing systems”, *IEEE International Conference on Pervasive Computing and Communications (PerCom),* **2017**, 79-88
43. Elazhary, H. “Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions”, *Journal of Network and Computer Applications*, 128, Elsevier, **2019**, 105-140.
44. Engel, T.A.; Charao, A.; Kirsch-Pinheiro, M. & Steffenel, L.A. "Performance Improvement of Data Mining in Weka through Multi-core and GPU Acceleration: opportunities and pitfalls", *Journal of Ambient Intelligence and Humanized Computing*, Springer, June **2015**. doi:10.1007/s12652-015-0292-9.
45. Engel, T.A.; Charao, A.; Kirsch-Pinheiro, M. & Steffenel, L.A., "Performance Improvement of Data Mining in Weka through GPU Acceleration", *5th International Conference on Ambient Systems, Networks and Technologies (ANT 2014)*, Hasselt, Belgium, June 2 - 5, 2014. Procedia Computer Science, 32, **2014**, Elsevier, 93–100.
46. Fensel, D.; Facca, F.M.; Simperl, E. & Toma, I., “*Semantic Web Services*”, Springer, **2011**, p. 350
47. Ferguson-Boucher, K., "Cloud Computing: A Records and Information Management Perspective", *IEEE Security & Privacy*, 9(6), Nov.-Dec. **2011**, 63-66. doi: 10.1109/MSP.2011.159
48. Garcia Lopez, P.; Montresor, A.; Epema, D.; Datta, A.; Higashino, T.; Iamnitchi, A.; Barcellos, M.; Felber, P. & Riviere, E. “Edge-centric Computing: Vision and Challenges”, *ACM SIGCOMM Computer Communication Review*, 45 (5), **2015**, 37-42.
49. García, K., “Ascertaining the Availability of Shared Resources in Ubiquitous Collaborative Environments”, PhD Thesis, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, **2013**.
50. Geihs, K.; Reichle, R.; Wagner, M. & Khan, M.U., “Modeling of Context-Aware Self-Adaptive Applications in Ubiquitous and Service-Oriented Environments”, In: Cheng, B.H.C., de Lemos, R., Giese, H., Inverardi, P. & Magee, J. (Eds.), *Software Engineering for Self-Adaptive Systems, Lecture Notes in Computer Science,* 5525, **2009**, 146-163
51. Gensel, J. ; Villanova-Oliver, M. & Kirsch-Pinheiro, M., « Modèles de contexte pour l'adaptation à l'utilisateur dans des Systèmes d'Information Web collaboratifs », *8èmes Journées Francophones d'Extraction et Gestion des Connaissances (EGC'08), Atelier sur la Modélisation Utilisateur et Personnalisation d'Interfaces Web,* **2008**, 5-15
52. Ghobaei-Arani, M. ; Souri, A. & Rahmanian, A. A., "Resource Management Approaches in Fog Computing: a Comprehensive Review”, *Journal of Grid Computing*, vol. 18, **2019**, Springer, 1-42. DOI 10.1007/s10723-019-09491-1
53. Greenberg, S., “Context as a dynamic construct”, *Human-Computing Interaction*, vol. 16, n° 2-4, **2001**, 257-268
54. Grudin, J., “Desituating action: digital representation of context”, *Human-Computing Interaction*, vol. 16, n° 2-4, **2001**, 269-286
55. Gutwin, C. & Greenberg, S., “A Descriptive Framework of Workspace Awareness for Real-Time Groupware”, *Computer Supported Cooperative Work (CSCW)*, 11(3-4), sept. **2002**, Kluwer Academic Publishers, 411 – 446
56. Hagras, H. “Intelligent pervasive adaptation in shared spaces”, In : Ferscha, A. *(Ed.)*, *Pervasive Adaptation: Next generation pervasive computing research agenda,* **2011**, 16. Disponible sur https://www.pervasive.jku.at/rab/pdf/PerAda%20Research%20Agenda.pdf (Last visit: Sept. 2019).

57. Hao, Z.; Novak, E. ; Yi, S. & Li, Q., “Challenges and software architecture for fog computing”, *IEEE Internet Computing*, 21(2), **2017**, 44–53. DOI:doi.ieeecomputersociety.org/ 10.1109/MIC.2017.26
58. Hauser, M.; Günther, S. A.; Flath, C. M. & Thiesse, F., “Designing Pervasive Information Systems: A Fashion Retail Case Study”, *Proceedings of the 38th International Conference on Information Systems (ICIS 2017),* Seul, South Korea, Dec. **2017**. Disponible sur https://www.researchgate.net/publication/319998755_Designing_Pervasive_Information_Systems_A_Fashion_Retail_Case_Study (Last visit: août 2020)
59. Hermosillo, G.; Seinturier, L. & Duchien, L., “Using Complex Event Processing for Dynamic Business Process Adaptation”, *7th IEEE 2010 International Conference on Services Computing (SCC 2010)*, Jul 2010, Miami, Florida, United States. IEEE Computer Society, **2010**, 466-473
60. Hofmann, P.; Woods, D., “Cloud computing: The limits of public clouds for business applications”, *IEEE Internet Computing*, 14 (6), Nov. **2010**, 90–93.
61. Hong, C.-H. & Varghese, B., “Resource Management in Fog/Edge Computing: A Survey on Architectures, Infrastructure, and Algorithms”, *ACM Computing Survey*, 52(5), **2019**, article n° 97. DOI:10.1145/3326066
62. Hong, J.; Suh, E.-H.; Kim, J. & Kim, S., “Context-aware system for proactive personalized service based on context history”, *Expert Systems with Applications*, 36(4), **2009**, 7448 – 7457
63. Hu, X.; Ding, Y.; Paspallis, N.; Bratskas, P.; Papadopoulos, G.A.; Vanrompay, Y.; Kirsch Pinheiro, M. & Berbers, Y., “A Hybrid Peer-to-Peer Solution for Context Distribution in Mobile and Ubiquitous Environments”, In: Papadopoulos G., Wojtkowski W., Wojtkowski G., Wrycza S., Zupancic J. (eds), *17th International Conference on Information Systems Development (ISD2008)*, *Information Systems Development: Towards a Service Provision Society*, **2008**, Springer, 501-510. DOI : 10.1007/b137171_52
64. Huang, D. & Wu, H. “Mobile Cloud Computing: foundations and service models”. Morgan Kaufmann, **2017**, p. 307. ISBN: 978-0-12-809641-3
65. Hughes, D.; Greenwood, P.; Blair, G.; Coulson, G.; Grace, P.; Pappenberger, F.; Smith, P. & Beven, K., “An Experiment with Reflective Middleware to Support Grid-based Flood Monitoring”, *Concurr. Comput.: Pract. Exper.,* John Wiley and Sons Ltd., 20(11), **2008**, 1303-1316.
66. IBM, “What is edge computing?”. Disponible sur : https://www.ibm.com/cloud/what-is-edge-computing (Last visit: avril **2019**).
67. Issarny, V.; Caporuscio, M. & Georgantas, N., “A Perspective on the Future of Middleware-based Software Engineering”. *Future of Software Engineerin (FOSE)*, **2007**, 244–258
68. Jackson, M., “*Software requirements specifications: a lexicon of practice, principles and prejudices*”, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, **1995**, p. 228
69. Jaffal, A. ; Grand, B. L. & Kirsch-Pinheiro, M., “Refinement Strategies for Correlating Context and User Behavior in Pervasive Information Systems”, *1st Workshop on Big Data and Data Mining Challenges on IoT and Pervasive (Big2DM), 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015)*, *Procedia Computer Science*, vol. 52, Jun **2015**, 1040-1046. DOI : http://dx.doi.org/10.1016/j.procs.2015.05.103
70. Jaffal, A. ; Kirsch-Pinheiro, M. & Le Grand, B., “Unified and Conceptual Context Analysis in Ubiquitous Environments”, In : Jaime Lloret Mauri, Christoph Steup & Sönke Knoch (Eds.), *8th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2014)*, August 24 - 28, 2014, ISBN 978-1-61208-353-7, IARIA, **2014**, 48-55
71. Jaffal, A., « *Aide à l'utilisation et à l'exploitation de l'Analyse de Concepts Formels pour des non-spécialistes de l'analyse des données* », Thèse de doctorat, Université Paris 1 Panthéon Sorbonne, **2019**. Disponible sur : https://tel.archives-ouvertes.fr/tel-02526323/ (Last visit: juillet 2020)
72. Jaffal, A., « *Analyse formelle de concepts pour la gestion du contexte* », Dissertation de master, Master Recherche Systèmes d'Information et de Décision, Université Paris 1 Panthéon Sorbonne, **2013**. p. 47
73. Jaffal, A.; Grand, B. L. & Kirsch-Pinheiro, M., « Extraction de connaissances dans les Systèmes d'Information Pervasifs par l'Analyse Formelle de Concepts », *Extraction et Gestion des*

Connaissances (EGC 2016), Revue des Nouvelles Technologies de l'Information, RNTI-E-30, **2016**, 291-296

74. Kaabi, R.S. & Souveyet, C., "Capturing Intentional services with Business Process Maps". *1rst Int. Conference on Research Challenges in Information Science (RCIS)*, **2007**, 309- 318
75. Khan, M.U., "Unanticipated Dynamic Adaptation of Mobile Applications", PhD thesis (Doktor der Ingenieurwissenschaften), University of Kassel, German, 31 March **2010**
76. Kirsch Pinheiro, M. ; Carrillo-Ramos, A. ; Villanova-Oliver, M. ; Gensel, J. & Berbers, Y., "Context-Aware Adaptation in Web-based Groupware Systems", In: JingTao Yao (Ed.), *Web-based Support Systems, Series: Advanced Information and Knowledge Processing*, Springer, mars **2010**, 3-31
77. Kirsch Pinheiro, M. « Adaptation Contextuelle et Personnalisée de l'Information de Conscience de Groupe au sein des Systèmes d'Information Coopératifs ». Thèse de Doctorat, Université Joseph-Fourier - Grenoble I, **2006**. p. 267
78. Kirsch-Pinheiro, M.; de Lima, J.V. & Borges, M.R.S., "A framework for awareness support in groupware systems", *Computers in Industry*, 52(1), **2003**, 47-57
79. Kirsch Pinheiro, M. & Souveyet, C., "Is Group-Awareness Context-Awareness?", In: Rodrigues, A.; Fonseca, B. & Preguiça, N. (Eds.), *24th International Conference on Collaboration and Technology (CRIWG 2018), LNCS 11001*, Springer, **2018**, 198-206
80. Kirsch Pinheiro, M. & Souveyet, C., "Is Group-Awareness Context-Awareness?: Converging Context-Awareness and Group Awareness Support", *International Journal of e-Collaboration (IJeC)*, Special Issue of Revised and Extended Papers from the 24th International Conference on Collaboration and Technology, 15(3), **2019**, p. 19. DOI: 10.4018/IJeC.2019070101
81. Kirsch Pinheiro, M.; Le Grand, B.; Souveyet, C. & Najar, S., « Espace de Services : Vers une formalisation des Systèmes d'Information Pervasifs », *XXXIème Congrès INFORSID 2013 : Informatique des Organisations et Systèmes d'Information et de Décision,* **2013**, 215-223
82. Kirsch-Pinheiro, M. & Souveyet, C. "Supporting context on software applications: a survey on context engineering" (« Le support applicatif à la notion de contexte : revue de la littérature en ingénierie de contexte »), *Modélisation et utilisation du contexte*, 2(1), **2018**, ISTE OpenScience. Disponible sur : https://www.openscience.fr/Le-support-applicatif-a-la-notion-de-contexte-revue-de-la-litterature-en/ (Last visit: août 2020)
83. Kirsch-Pinheiro, M. & Souveyet, C., « Le Rôle des Ressources dans l'Évolution des Systèmes d'Information », *Actes du XXXVIIème Congrès INFORSID (INFORSID 2019),* Paris, France, Juin 11-14 **2019**, 85-97
84. Kirsch-Pinheiro, M., Mazo, R., Souveyet, C. & Sprovieri, D., "Requirements Analysis for Context-oriented Systems", *7th Int. Conf.on Ambient Systems, Networks and Technologies (ANT 2016), Procedia Computer Science*, 83, **2016**, 253-261
85. Kirsch-Pinheiro, M. & Souveyet, C., "Quality on Context Engineering". In: Brézillon P., Turner R., Penco C. (Eds.) *Modeling and Using Context. CONTEXT 2017. Lecture Notes in Computer Science*, 10257, **2017**, 432-439
86. Kirsch-Pinheiro, M.; Vanrompay, Y.; Victor, K.; Berbers, Y.; Valla, M.; Frà, C.; Mamelli, A.; Barone, P.; Hu, X.; Devlic, A. & Panagiotou, G., "Context Grouping Mechanism for Context Distribution in Ubiquitous Environments", In: Robert Meersman, Zahir Tari *et al.* (Eds.), *10th International Symposium on Distributed Objects, Middleware, and Applications (DOA'08), OTM 2008 Conferences, Lecture Notes in Computer Science*, 5331, **2008**, 571-588
87. Kirsch-Pinheiro, M.; Vanrompay, Y. & Berbers, Y., "Context-aware service selection using graph matching". In: Paoli, F. D.; Toma, I.; Maurino, A.; Tilly, M. & Dobson, G. (Eds.), *2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC'08), at ECOWS 2008*, CEUR Workshop proceedings, 411, **2008**. Disponible sur : http://ceur-ws.org/Vol-411/ (Last visit: août 2020)
88. Kirsch-Pinheiro, M. & Rychkova, I., "Dynamic Context Modeling for Agile Case Management", In: Y.T. Demey & H. Panetto (Eds.), *OTM 2013 Workshops, Lecture Notes in Computer Science,* 8186, Springer-Verlag, **2013**, 144–154

89. Kirsch-Pinheiro, M.; Gensel, J. & Martin, “Representing Context for an Adaptative Awareness Mechanism”. In: H. Vreede, G.-J.; Guerrero, L. & Marin Raventós, G. (Eds.), *Groupware: Design, Implementation, and Use, X International Workshop on Groupware (CRIWG 2004), Lecture Notes in Computer Science,* 3198, Springer Berlin Heidelberg, **2004**, 339-348
90. Kirsch-Pinheiro, M.; Gensel, J. & Martin, H., “Awareness on Mobile Groupware Systems”. In: Ahmed Karmouch, Larry Korba, Edmundo Roberto Mauro Madeira (Eds.), *1st International Workshop on Mobility Aware Technologies and Applications (MATA 2004)*, *Lecture Notes in Computer Science*, 3284, **2004**, 78-87
91. Kirsch-Pinheiro, M.; Villanova-Oliver, M.; Gensel, J. & Martin, H., "Context-aware filtering for collaborative web systems: adapting the awareness information to the user's context”. In: Hisham Haddad, Lorie M. Liebrock, Andrea Omicini, Roger L. Wainwright (Eds.), *Proceedings of the 2005 ACM Symposium on Applied Computing (SAC 2005)*, **2005**, 1668-1673
92. Kirsch-Pinheiro, M.; Villanova-Oliver, M.; Gensel, J. & Martin, H., “BW-M: a framework for awareness support in Web-based groupware systems”. In: Weiming Shen, Anne E. James, Kuo-Ming Chao, Muhammad Younas, Zongkai Lin, Jean-Paul A. Barthès (Eds.), *Proceedings of the 9th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2005)*, Volume 1, **2005**, 240-246
93. Kirsch-Pinheiro, M.; Villanova-Oliver, M.; Gensel, J. & Martin, H., “A Personalized and Context-Aware Adaptation Process for Web-Based Groupware Systems”. In: Moira C. Norrie, Schahram Dustdar, Harald Gall (Eds.), *Proceedings of the CAISE 06, Workshop on Ubiquitous Mobile Information and Collaboration Systems* (UMICS 2006), **2006**, 884-898. Disponible sur http://ceur-ws.org/Vol-242/ (Last visit: août 2020).
94. Kirsch Pinheiro, M.; Villanova-Oliver, M.; Gensel, J.; Berbers, Y. & Martin, H., “Personalizing Web-Based Information Systems through Context-Aware User Profiles”, *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (Ubicomm 2008),* **2008**, 231-238. DOI: 10.1109/UBICOMM.2008.18
95. König, I.; Voigtmann, C.; Klein, B. & David, K., “Enhancing Alignment Based Context Prediction by Using Multiple Context Sources: Experiment and Analysis”. In: Beigl, M.; Christiansen, H.; Roth-Berghofer, T.; Kofod-Petersen, A.; Coventry, K. & Schmidtke, H. (Eds.), *Modeling and Using Context*, LNCS 6967, **2011**, 159-172
96. Kourouthanassis, P. E. & Giaglis, G. M., “A Design Theory for Pervasive Information Systems”. *3rd Int. Workshop on Ubiquitous Computing (IWUC 2006), in conjunction with ICEIS 2006*, May **2006**, Paphos, Cyprus, 62-70. DOI: 10.5220/0002503700620070. Disponible sur : https://www.scitepress.org/ Papers/2006/25037/25037.pdf (Last visit: Aout 2020)
97. Kourouthanassis, P. E.; Giaglis, G. M. & Karaiskos, D. C., “Delineating `pervasiveness' in pervasive information systems: a taxonomical framework and design implications”. *Journal of Information Technology*, 25(3), **2010**, 273-287
98. Krogstie, J.; Lyytinen, K.; Opdahl, A.L.; Pernici, B., Siau, K. & Smolander, K., "Research areas and challenges for mobile information systems”. *International Journal of Mobile Communications*, 2(3), September **2004**, 220–234. DOI: 10.1504/IJMC.2004.005161
99. Lopez Peña, M. A. & Muñoz Fernández, I., “SAT-IoT: An Architectural Model for a High-Performance Fog/Edge/Cloud IoT Platform”, *IEEE 5th World Forum on Internet of Things (WF-IoT),* **2019**, 633-638
100. Lu, Y., “Industry 4.0: A survey on technologies, applications and open research issues”, *Journal of Industrial Information Integration*, 6, June **2017**, 1 – 10
101. Maass, W. & Varshney, U., “Design and evaluation of Ubiquitous Information Systems and use in healthcare”, *Decision Support Systems*, 54(1), **2012**, 597-609. DOI: 10.1016/j.dss.2012.08.007
102. Mayrhofer, R.; Harald, R. & Alois, F., “Recognizing and predicting context by learning from user behavior”. In. W. Schreiner, G. Kotsis, A. Ferscha, & K. Ibrahim (Eds.), *Int. Conf. on Advances in Mobile Multimedia (MoMM2003)*, **2003**, 25–35

103. Mirbel, I. & Crescenzo, P., “From End-User’s Requirements to Web Services Retrieval: A Semantic and Intention-Driven Approach”. In: J.-H. Morin, J. Ralyté & M. Snene (eds.), *Exploring Services Science*, Springer Berlin Heidelberg, **2010**, 30–44
104. Moran T. & Dourish, P., “Introduction to this special issue on context-aware computing”, *Human-Computer Interaction*, 16 (2-3), **2001**, 87–95.
105. Mulfari, D.; Fazio, M.; Celesti, A.; Villari, M. & Puliafito, A., “Design of an IoT cloud system for container virtualization on smart objects”. *3rd Workshop on CLoud for IoT (CLIoT 2015), European Conference on Service-Oriented and Cloud Computing (ESOCC 2015), Communications in Computer and Information Science (CCIS)*, vol. 567, **2015**, Springer, 33-47
106. Najar, S. ; Kirsch-Pinheiro, M. & Souveyet, C., “Enriched Semantic Service Description for Service Discovery: Bringing Context to Intentional Services”, *International Journal On Advances in Intelligent Systems*, volume 5, numbers 1 & 2, June **2012**, 159-174, IARIA Journals / ThinkMind, ISSN: 1942-2679
107. Najar, S., « Adaptation des services sensibles au contexte selon une approche intentionnelle », Phd Thesis, Université Paris 1 Panthéon Sorbonne, Paris, France, **2014**
108. Najar, S.; Saidani, O.; Kirsch-Pinheiro, M.; Souveyet, C. & Nurcan, S., “Semantic representation of context models: a framework for analyzing and understanding” In: J. M. Gomez-Perez, P. Haase, M. Tilly, and P. Warren (Eds), *Proceedings of the 1st Workshop on Context, information and ontologies (CIAO 09), European Semantic Web Conference (ESWC'2009)*, **2009,** 1-10
109. Najar, S.; Kirsch Pinheiro, M. & Souveyet, C., “The influence of context on intentional service”. *5th Int. IEEE Workshop on Requirements Engineerings for Services (REFS’11), IEEE Conference on Computers, Software, and Applications (COMPSAC’11)*, Munich, Germany, 470-475, **2011**
110. Najar, S.; Kirsch Pinheiro, M. & Souveyet, C., “Bringing context to intentional services”. *3rd Int. Conference on Advanced Service Computing, Service Computation’11*, Rome, Italy, **2011**, 118-123. Best Paper Awards
111. Najar, S.; Kirsch Pinheiro, M. & Souveyet, C., “Towards Semantic Modeling of intentional pervaisve System”, *6th International Workshop on Enhanced Web Service Technologies (WEWST’11), European Conference on Web Services (ECOWS’11)*, Lugano, Switzerland, **2011**, 30-34
112. Najar, S.; Kirsch Pinheiro, M.; Le Grand, B. & Souveyet, C., “A user-centric vision of service-oriented Pervasive Information Systems”, *8th International Conference on Research Challenges in Information Science (RCIS 2014)*, IEEE, **2014**, 359-370
113. Najar, S.; Kirsch Pinheiro, M.; Le Grand, B. & Souveyet, C., « Systèmes d'Information Pervasifs et Espaces de Services : Définition d'un cadre conceptuel ». *UbiMob 2013 : 9èmes journées francophones Mobilité et Ubiquité*, Jun **2013**, Nancy, France. Disponible sur https://ubimob2013.sciencesconf.org/19119.html (Last visit: août 2020)
114. Najar, S.; Kirsch-Pinheiro, M. & Souveyet, C. "Service discovery and prediction on Pervasive Information System", *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, issue 4, June **2015**, Springer, 407-423. ISSN: 1868-5137. DOI: 10.1007/s12652-015-0288-5
115. Najar, S.; Kirsch-Pinheiro, M. & Souveyet, C., “A context-aware intentional service prediction mechanism in PIS”, In: David De Roure, Bhavani Thuraisingham & Jia Zhang (Eds.), *IEEE 21st International Conference on Web Services (ICWS 2014)*, 27 June - 2 July **2014**, Anchorage, Alaska, USA, IEEE CS, 662-669. DOI : 10.1109/ICWS.2014.97
116. Najar, S. ; Kirsch-Pinheiro, M. & Souveyet, C., “A new approach for service discovery and prediction on Pervasive Information System”, *5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014), Procedia Computer Science*, v32, **2014**, Elsevier, 421–428. DOI : http://dx.doi.org/10.1016/j.procs.2014.05.443
117. Najar, S.; Kirsch Pinheiro, M.; Vanrompay, Y.; Steffenel, L.A. & Souveyet, C., “Intention Prediction Mechanism in an Intentional Pervasive Information System”, In : Kolomvatsos, K., Anagnostopoulos, C., Hadjiefthymiades, S. (Eds.), *Intelligent Technologies and Techniques for Pervasive Computing*, IGI Global, **2013**, pp. 251-275. DOI: 10.4018/978-1-4666-4038-2.ch014, ISBN : 978-1-4666-4040-5

118. Najar, S.; Kirsch-Pinheiro, M.; Souveyet, C. & Steffenel, L. A., "Service Discovery Mechanisms for an Intentional Pervasive Information System". *Proceedings of 19th IEEE International Conference on Web Services (ICWS 2012)*, Honolulu, Hawaii, 24-29 June **2012**, 520-527
119. Najar, S. ; Kirsch-Pinheiro, M. ; Steffenel, L. A. & Souveyet, C., « Analyse des mécanismes de découverte de services avec prise en charge du contexte et de l'intention ». In : Philippe Roose & Nadine Rouillon-Couture (dir.), *8èmes Journées Francophones Mobilité et Ubiquité (Ubimob 2012)*, June 4-6, **2012**, Anglet, France. Cépaduès Editions, 210-221. ISBN 978.2.36493.018.6
120. Najar, S. ; Kirsch-Pinheiro, M. & Souveyet, C., « Mécanisme de prédiction dans un système d'information pervasif et intentionnel », In : Philippe Roose & Nadine Rouillon-Couture (dir.), *8èmes Journées Francophones Mobilité et Ubiquité (Ubimob 2012)*, June 4-6, **2012**, Anglet, France. Cépaduès Editions, pp. 146-157. ISBN: 978.2.36493.018.6
121. Neumann, G.; Sobernig, S. & Aram, M., "Evolutionary Business Information Systems", *Business & Information Systems Engineering*, 6, **2014**, 33-38. DOI: 10.1007/s12599-013-0305-1
122. Olaniyan, R.; Fadahunsi, O.; Maheswaran, M. & Zhani, M.F. "Opportunistic edge computing: Concepts, opportunities and research challenges", *Future Generation Computer Systems*, 89, **2018**, 633 – 645.
123. Papazoglou, M.P. & Georgakopoulos, D., "Introduction: Service-oriented computing". *Communication of the ACM*, 46(10), **2003**, 24–28
124. Papazoglou, M.P., "Service-oriented computing: concepts, characteristics and directions". In *Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE 2003)*, **2003**, 3–12
125. Papazoglou, M.P.; Traverso, P.; Dustdar, S. & Leymann, F., "Service-Oriented Computing: A Research Roadmap". *Int. Journal of Cooperative Information Systems*, 17(2), **2008**, 223-255
126. Parashar, M. & Pierson, J.-M., "Pervasive grids: Challenges and opportunities", In: K. Li, C. Hsu, L. Yang, J. Dongarra & H. Zima (Eds.), *Handbook of Research on Scalable Computing Technologies*, IGI Global, **2010**, 14–30.
127. Paspallis, N.; Rouvoy, R.; Barone, P.; Papadopoulos, G.A.; Eliassen, F. & Mamelli, A., "A Pluggable and Reconfigurable Architecture for a Context-Aware Enabling Middleware System". In: Meersman, R. & Tari, Z. (Eds.), *On the Move to Meaningful Internet Systems: Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Lecture Notes in Computer Science,* 5331, **2008**, Springer, 553–570
128. Petit, M.; Ray, C. & Claramunt, C., « Algorithme de recommandation adaptable pour la personnalisation d'un système mobile », *6èmes Journées Francophones Ubiquité et Mobilité (UBIMOB 2010)*, **2010**, 59-62
129. Ploesser, K.; Recker, J. & Rosemann, M., "Supporting Context-Aware Process Design: Learnings from a Design Science Study", In: zur Muehlen, M. & Su, J. (Eds.), *Business Process Management Workshops (BPM 2010)*, *Lecture Notes in Business Information Processing*, 66., Springer, **2010**, 97-104
130. Preuveneers, D.; Victor, K.; Vanrompay, Y.; Rigole, P.; Kirsch Pinheiro, M. & Berbers, Y. "Context-Aware Adaptation in an Ecology of Applications". In: Dragan Stojanovic (Ed.), *Context-Aware Mobile and Ubiquitous Computing for Enhanced Usability: Adaptive Technologies and Applications*, IGI Global, **2009**.
131. Priss, U., "Formal Concept Analysis in Information Science". In: Blaise, C. (eds.), *Annual Review of Information Science and Technology*, ASIST, vol. 40, **2006**, 521-543
132. Ramakrishnan, A.; Preuveneers, D. & Berbers, Y., "Enabling self-learning in dynamic and open IoT environments", In: Shakshuki, E. & Yasar, A. (Eds.), *The 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014)*, Procedia Computer Science, 32, **2014**, Elsevier, 207-214
133. Ramakrishnan, A.K., "Support for Data-driven Context Awareness in Smart Mobile and IoT Applications: Resource Efficient Probabilistic Models and a Quality-aware Middleware Architecture" (Ondersteuning voor data-gedreven context-bewustzijn in intelligente mobiele en

IoT applicaties: Hulpbronnenefficiënte probabilistische modellen en een kwaliteit-aware middleware architectuur), PhD thesis, Katholieke Universiteit Leuven, Belgium **2016**

134. Rychkova, I. ; Kirsch Pinheiro M. & Le Grand B. “Context-Aware Agile Business Process Engine: Foundations and Architecture”, In : Nurcan, S., Proper, H., Soffer, P., Krogstie, J., Schmidt, R., Halpin, T. & Bider, I. (Eds.), *Enterprise, Business-Process and Information Systems Modeling, Proceedings of the 14th Working Conference on Business Process Modeling, Development, and Support (BPMDS 2013)*, *Lecture Notes in Business Information Processing*, 146, Valence, Spain, **2013**, 32-47
135. Rychkova I. ; Kirsch-Pinheiro M. & Le Grand B., “Automated Guidance for Case Management: Science or Fiction?”, In : Ficher, L. (Ed.), *Empowering Knowledge Workers: New Ways to Leverage Case Management*, Series BPM and Workflow Handbook Series, Future Strategies Inc., **2014**, 67-78. ISBN : 978-0-984976478
136. Rolland, C., “Capturing System Intentionality with Maps”. In: J. Krogstie, A.L. Opdahl & S. Brinkkemper (Eds.), *Conceptual Modelling in Information Systems Engineering*, Springer Berlin Heidelberg, **2007**, 141–158
137. Rolland, C.; Kirsch-Pinheiro, M. & Souveyet, C., “An Intentional Approach to Service Engineering”, *IEEE Transactions on Services Computing*, 3(4), Oct.-Dec. **2010**, 292-305
138. Römer, K. & Mattern, F., “Adaptation without antecipation”, In : Ferscha, A., *Pervasive Adaptation: Next generation pervasive computing research agenda*, 28-29, **2011**. Disponible sur https://www.pervasive.jku.at/Conferences/fet11/RAB.pdf (dernière visite: août 2020)
139. Rottenberg, S.; Leriche, S.; Taconet, C.; Lecocq, C. & Desprats, T. “MuSCa: A multiscale characterization framework for complex distributed systems”, *2014 Federated Conference on Computer Science and Information Systems*, **2014**, 1657-1665
140. Saidani, O.; Rolland, C. & Nurcan, S., “Towards a Generic Context Model for BPM”. *48th Annual Hawaii International Conference on System Sciences (HICSS’2015)*, Jan **2015**
141. Santos, L.O.B. da S.; Guizzardi, G.; Pires, L.F. & Sinderen, M., “From User Goals to Service Discovery and Composition”. In: Heuser C.A. & Pernul G. (Eds), *Advances in Conceptual Modeling - Challenging Perspectives (ER 2009)*, Lecture Notes in Computer Science, 5833, **2009**, 265-274
142. Sawyer, P., Mazo, R., Diaz, D., Salinesi, C., Hughes, D., “Using Constraint Programming to Manage Configurations in Self-Adaptive Systems”. *Computer*, 45(10), **2012**, 56-63
143. Schadt, E.E.; Linderman, M.D.; Sorenson, J.; Lee, L. & Nolan, G.P., “Computational solutions to large-scale data management and analysis”. *Nature Reviews Genetics*, 11(9), Sept. **2010**, 647–657
144. Schmidt, K., “The problem with ‘awareness’: introductory remarks on ‘Awareness in CSCW’”, *Computer Supported Cooperative Work*, 11(3-4), sept. **2002**, Kluwer Academic Publishers, 285-298
145. Schreiber, F. A.; Camplani, R.; Fortunato, M.; Marelli, M. & Rota, G., "PerLa: A Language and Middleware Architecture for Data Management and Integration in Pervasive Information Systems", *IEEE Transactions on Software Engineering*, 38 (2), March-April **2012**, 478-496, DOI: 10.1109/TSE.2011.25
146. Shekhar, S. & Gokhale, A., “Dynamic Resource Management Across Cloud-Edge Resources for Performance-Sensitive Applications”, *17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID),* **2017**, 707-710
147. Shoaib, M.; Bosch, S.; Incel, O.D.; Scholten, H. & Havinga, P. J.M. “A survey of online activity recognition using mobile phones”. *Sensors*, 15(1), **2015**, 2059–2085
148. Sigg, S.; Haseloff, S. & David, K. “An Alignment Approach for Context Prediction Tasks in UbiComp Environments”. *IEEE Pervasive Computing*, vol. 9, issue 4, **2010**, 90–97
149. Singh, S.& Chana, I., “A Survey on Resource Scheduling in Cloud Computing: Issues and Challenges”, *Journal of Grid Computing*, 14(2), **2016**, 217-264
150. Souveyet, C.; Villari, M.; Steffenel, L. A. & Kirsch-Pinheiro, M., « Une approche basée sur les MicroÉléments pour l'Évolution des Systèmes d'Information », *Atelier Évolution des SI : vers des SI Pervasifs ?, INFORSID 2019,* **2019.** Disponible sur https://evolution-si.sciencesconf.org/data/book_evolution_si_fr.pdf (Last visit: août 2020)

151. Steffenel, L. & Kirsch-Pinheiro, M. “CloudFIT, a PaaS platform for IoT applications over Pervasive Networks”, In: Celesti A., Leitner P. (Eds). *3rd Workshop on CLoud for IoT (CLIoT 2015)*. Advances in Service-Oriented and Cloud Computing (ESOCC 2015). Communications in Computer and Information Science, 567, **2015**, 20-32.
152. Steffenel, L. A.; Flauzac, O.; Charao, A. S.; Barcelos, P.; Stein, B.; Cassales, G.; Nesmachnow, S.; Rey, J.; Cogorno, M.; Kirsch-Pinheiro, M. & Souveyet, C., "Mapreduce challenges on pervasive grids", *Journal of Computer Science*, 10(11), July **2014**, 2194-2210.
153. Steffenel, L. A.; Flauzac, O.; Charao, A. S.; Barcelos, P. P.; Stein, B.; Nesmachnow, S.; Kirsch Pinheiro, M. & Diaz, D., “PER-MARE: Adaptive Deployment of MapReduce over Pervasive Grids”, *8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC'13)*, **2013**, 17-24.
154. Steffenel, L.A. & Kirsch Pinheiro, M. “When the cloud goes pervasive: approaches for IoT PaaS on a mobiquitous world”. In: Mandler B. *et al.* (Eds), *EAI International Conference on Cloud, Networking for IoT systems (CN4IoT 2015), Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST)*, 169, **2015**, 347–356.
155. Steffenel, L.A. & Kirsch-Pinheiro, M., "Leveraging Data Intensive Applications on a Pervasive Computing Platform: the case of MapReduce", *1st Workshop on Big Data and Data Mining Challenges on IoT and Pervasive (Big2DM),* London, UK, June 2 - 5, 2015. Procedia Computer Science, 52, Jun **2015**, Elsevier, 1034–1039. doi: 10.1016/j.procs.2015.05.102.
156. Steffenel, L.A. & Kirsch-Pinheiro, M., « Accès aux Données dans le Fog Computing : le cas des dispositifs de proximité », *Conférence d’informatique en Parallélisme, Architecture et Système (ComPAS'19)*, 25-28 July **2019**, Anglet, France. Disponible sur https://hal.univ-reims.fr/hal-02174708 (Last visit: aout 2020)
157. Steffenel, L.A. & Kirsch-Pinheiro, M., "Improving Data Locality in P2P-based Fog Computing Platforms", *9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2018), Procedia Computer Science*, 141, Elsevier, **2018**, 72-79. DOI : 10.1016/j.procs.2018.10.151
158. Steffenel, L.A.; Kirsch-Pinheiro, M.; Vaz Peres, L. & Kirsch Pinheiro, D. "Strategies to implement Edge Computing in a P2P Pervasive Grid", *International Journal of Information Technologies and Systems Approach (IJITSA)*, IGI Global, 11(1), **2018**, 1-15. DOI: doi:10.4018/IJITSA/2018010101
159. Steffenel, L.A. & Kirsch-Pinheiro, M., « Stratégies Multi-Échelle pour les Environnements Pervasifs et l’Internet des Objets ». *11èmes Journées Francophones Mobilité et Ubiquité (Ubimob 2016)*, 5 juillet **2016**, Lorient, France. Paper n°6. Disponible sur https://ubimob2016.telecom-sudparis.eu/files/2016/07/Ubimob_2016_paper_6.pdf (Dernière visite: aout 2020)
160. Sundmaeker, H.; Guillemin, P.; Friess, P. & Woelfflé, S., “Vision and Challenges for realizing the internet of things”. *Cluster of European Research projects on the Internet of Things (CERP-IoT)*, **2010**. DOI: 10. 2759/26127. Disponible sur : http://www.theinternetofthings.eu/sites/default/files/Rob%20van%20Kranenburg/Clusterbook%202009_0.pdf (Last visit: novembre 2014)
161. TechTarget, “Cutting edge: IT's guide to edge data centers”, SearchDataCenter, June **2016**. Disponible sur : https://searchdatacenter.techtarget.com/guide/Cutting-edge-ITs-guide-to-edge-data-centers (Last visit: avril 2019).
162. Tyagi, R. & Gupta, S.K., “A survey on scheduling algorithms for parallel and distributed systems”. In: Mishra, A.; Basu, A. & Tyagi V. (eds), *Silicon Photonics & High Performance Computing*, *Advances in Intelligent Systems and Computing*, vol. 718, **2018**, 51 – 64. DOI: 10.1007/978-981-10-7656-5_7
163. Van Lamsweerde A., “Requirements Engineering in the Year 00: A Research Perspective”. *22nd International Conference on Software Engineering (ICSE’2000)*, Limerick, Ireland, **2000**, 5-19
164. Vanrompay, Y.; Kirsch Pinheiro, M.; Ben Mustapha, N.; Aufaure, M.-A., “Context-Based Grouping and Recommendation in MANETs”, In : Kolomvatsos, K., Anagnostopoulos, C., Hadjiefthymiades, S. (Eds.), *Intelligent Technologies and Techniques for Pervasive Computing*, IGI Global, **2013**, 157-178.

165. Vanrompay, Y.; Kirsch-Pinheiro, M. & Berbers, Y., “Context-Aware Service Selection with Uncertain Context Information”, *Context-Aware Adaptation Mechanism for Pervasive and Ubiquitous Services 2009 (CAMPUS 2009), Electronic Communications of the EASST, vol. 19*, **2009**
166. Vanrompay, Y.; Kirsch-Pinheiro, M. & Berbers, Y., “Service Selection with Uncertain Context Information”, In: Stephan Reiff-Marganiec and Marcel Tilly (Eds.), *Handbook of Research on Service-Oriented Systems and Non-Functional Properties: Future Directions*, IGI Global, 192-215, **2011**
167. Vanrompay, Y.; Mehlhase, S. & Berbers, Y. “An effective quality measure for prediction of context information”, *8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, **2010**, 13 -17
168. Velasquez, K. ; Abreu, D.P. ; Assis, M. *et al.* “Fog orchestration for the Internet of Everything: state-of-the-art and research challenges”. *Journal of Internet Services and Applications*, 9, article 14, **2018**. DOI: https://doi.org/10.1186/s13174-018-0086-3
169. Villari, M.; Fazio, M.; Dustdar, S.; Rana, O. & Ranjan R., "Osmotic Computing: A New Paradigm for Edge/Cloud Integration”, *IEEE Cloud Computing*, 3(6), **2016**, 76-83
170. Wagner, M.; Reichle, R.; Khan, M.U.; Geihs, K.; Lorenzo, J.; Valla, M.; Frà, C.; Paspallis, N. & Papadopoulos, G.A., “A Comprehensive Context Modeling Framework for Pervasive Computing Systems”, *8th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS 2008)*, Lecture Notes on Computer Science, vol. 5053, **2008**, 281-295
171. Weiser, M., “The computer for the 21st century”, *Scientific American*, vol. 265, No 3, September **1991**, 94-104
172. White, Tom, “Hadoop: The definitive guide”, 2nd edition, O’Reilly, **2010**, ISBN 978-1-449-38973-4, p. 626.
173. Wille, R., “Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies”. In: Ganter, B. *et al.* (Eds.), *Formal Concept Analysis*, Springer-Verlag, **2005**, 1-33
174. Yu, E.S.K. & Mylopoulos, J., “Why Goal-Oriented Requirements Engineering”. *Int. Working Conference on Requirement Engineering: Foundation for Software Quality*, **1998**, 15-22
175. Zaidenberg, S., « Apprentissage par renforcement de modèles de contexte pour l’informatique ambiante ». PhD Thesis, Institut National Polytechnique de Grenoble - INPG, **2009**

Annexes

Annex I

Paper

Kirsch-Pinheiro, M. & Souveyet, C. "Supporting context on software applications: a survey on context engineering", Modélisation et utilisation du contexte, 2(1), 2018, ISTE OpenScience.

Supporting context on software applications: a survey on context engineering

Le support applicatif à la notion de contexte : revue de la littérature en ingénierie de contexte

Manuele Kirsch Pinheiro[1], Carine Souveyet[2]

[1] Centre de Recherche en Informatique, Université Paris 1 Panthéon Sorbonne, France, mkirschpin@univ-paris1.fr
[2] Centre de Recherche en Informatique, Université Paris 1 Panthéon Sorbonne, France, souveyet@univ-paris1.fr

ABSTRACT. Engineering context-aware applications, i.e. applications that are able to adapt their behavior according to context information, is a complex task. Not only is context a large and complex notion, but its support on software applications involves tackling multiple challenges and issues. These challenges involve not only technical challenges, but also quality concerns. Indeed, with the growing development of context-aware applications, it is becoming essential to start considering the quality of context on every step of the application development. The goal of this paper is to provoke discussion on the issues related to the support of the notion of context and its quality concerns on software applications. We present here a roadmap on context management considering different dimensions of supporting context and quality of context (QoC) on software applications, and a literature review of solutions and issues related to these dimensions. Through these, we aim at sharing with non-expert designers the necessary expertise on context management allowing them to better understand the notion of context and QoC and their challenges.

RÉSUMÉ. La conception d'applications sensibles au contexte, i.e. applications capables d'adapter leur comportement au contexte d'exécution, est une tâche complexe. Non seulement la notion de contexte correspond à un concept large et complexe, mais également son support au sein d'un logiciel implique la prise en compte de plusieurs challenges. Ceux-ci ne se limitent pas aux challenges techniques, incluant aussi le support à la qualité de contexte (QoC). Avec le développement croissant de ces applications, il devient essentiel de considérer la notion de qualité à chaque étape de leur développement. L'objectif ici est ainsi d'inciter la discussion et la prise de conscience sur ces différents aspects liés à la gestion de contexte et de ses paramètres de qualité. Nous présentons une roadmap tenant compte des différentes dimensions nécessaires à la gestion de contexte, ainsi qu'une révision de littérature discutant les solutions et les problèmes liés à ces dimensions. A travers ces éléments, nous voulons partager une connaissance nécessaire à la compréhension de la notion de contexte et de QoC, et à la conception d'applications sensibles au contexte par de concepteurs non-experts.

KEYWORDS. Context-aware computing, context engineering, Quality of Context.
MOTS-CLÉS. Informatique sensible au contexte, ingénierie de contexte, qualité de contexte.

1. Introduction

Observing the environment using software applications is now possible. The development of low cost sensors, actuators, nano-computers and other IoT-related technologies is allowing software developers to easily propose applications that observe and interact with the physical environment. Applications may now observe the execution context and integrate such information into their own behavior. The growing interest for IoT applications demonstrates this tendency quite well.

The capability of sensing enables the design of new intelligent systems that are aware of their context and able to adapt their behavior accordingly [2]. In other words, thanks to this growing development of technology, one may also expect to observe in the next few years a growing development of context-aware applications, i.e. applications that are able to adapt their own behavior according their execution context [1][21]. We are already seeing this phenomenon, with an increasing number of applications that are able to observe elements from the environment (e.g. the user's location, physical activity, etc.) and to adapt the content proposed to the user accordingly. This kind of application is already part of our everyday life. However, in most of cases, its development is still

performed in an ad-hoc way, despite all the research that has been done about context-aware applications. Undeniably, supporting context information on software applications involves several technical challenges, with multiple impacts on the application architecture. Currently, the main challenge is no longer on the development itself, but mainly on understanding the issues involved and on exploring the opportunities that arise through these new technologies. Indeed, in order to go further with the technology itself and the development of simple ad-hoc solutions, it is necessary to better understand the notion of context and its challenges, since this notion is central for the design and conception of new solutions.

Understanding the notion of context and its support is a complex but necessary task. It is complex because the notion of context is itself a complex and ambiguous notion, whose support on software systems involves several technical issues. It is necessary because it is only by understanding this notion and its support that we will be able to explore the full potential of it and all the opportunities it opens for business models and Information Systems. It is only through a better understanding of this notion that a real "context engineering" process can be achieved, allowing the production of new complex and extensible context-aware applications. More than ever, it is becoming necessary to form a new generation of software engineers capable of "thinking" about context in the same way they are able to think about components and about object-oriented solutions.

In our opinion, it is important to supply non-expert designers with the necessary knowledge about the issues and challenges related to context support and management on software systems. It is only through this knowledge that the above-mentioned understanding will be developed. In the past, we have identified a set of dimensions, which we consider to be necessary for such support [31] and analyzed the impact of quality considerations on such dimensions [32]. These dimensions act as guidelines in a requirement analysis process, helping non-expert users to identify necessary issues on context support for new software applications. This support can be greatly affected by quality concerns (for instance, precision and uncertainty issues affecting the information reliability), making the quality support a transversal concern affecting all dimensions of context support.

Together, all these aspects, analyzed separately in [31][32], offer a global view of the challenges involved with software support for the notion of context. In this paper, we discuss this global view, thanks to a literature review pointing out existing solutions and open issues related to context support and management. The goal of this paper is to build a survey on context engineering for non-expert software developers and designers. This survey is intended as a basis for training new "context engineers", capable of understanding and building new context-aware applications for tomorrow's Information Systems.

The paper is organized as follows: Section 2 introduces our motivations and illustrative scenarios; Section 3 discusses the context engineering dimensions and their quality support, extending what we have proposed in [31] and [32]; Section 4 introduces a literature review, pointing out challenges and solutions for each context support dimension; finally, Section 5 discusses conclusions and future work.

2. Motivation & illustrative examples

2.1. *Illustrative scenarios*

Today, it is undeniable that computation is embedded into our everyday life; we continually use computational devices without thinking of them as computational in any way [4]. Indeed, we live surrounded by multiple computing devices, forming a truly pervasive environment, as envisioned by [63]. The dynamic and ad-hoc nature of such environments leads intrinsically to important adaptation needs: the environment has to adapt to changing operating conditions and changing user preferences and behaviors in order to enable more efficient and effective operation, while avoiding system failure [26]. The user acceptation of such environments depends on these adaptation capabilities.

Context-aware systems can be seen as applications that are able to respond to these changes. They are defined as applications capable of observing context changes and adapting their behavior accordingly [1] [21]. Compared with traditional software applications, context-aware applications can be considered as more complex since they must cope with heterogeneous and dynamic environments. They have to run, often continuously, under changing conditions. They must observe different elements from the environment and react to their changes accordingly, often using very constrained computing platforms (for instance, nano-computers or smartphones with battery and connectivity limitations). Such a dynamic and constrained execution environment has a significant impact on the software architecture and development, notably in terms of modularity, integration, interoperability and increasing number of non-functional constraints (e.g., robustness, scalability). Under these conditions, traditional software qualities, such as flexibility, dynamicity, modularity and extensibility, become difficult to satisfy, notably with ad-hoc development processes, which are still frequently adopted when developing context-aware applications, as observed in [2][3].

A central aspect that makes developing context-aware applications complex is the notion of context itself. The notion of context corresponds to a large and ambiguous concept that has been analyzed several different ways in Computer Science and other domains [6] [7] [8] [42]. Supporting this notion on a software application involves different challenges, from identifying relevant context information, acquiring and modeling it up to its interpretation and exploitation for different purposes [1] [31] [32]. It quickly becomes arduous for non-expert designers to design and build new applications using this notion.

Before considering the challenges of developing new context-aware applications, let us consider some scenarios for such applications. Several application domains may benefit from context-aware systems, including the smart cities and smart agriculture domains. In order to demonstrate the interest of such applications, let us consider three illustrative scenarios.

The first scenario we would like to mention is a flood warning scenario, proposed by [28] [55]. This scenario, called GridStix, considers a Wireless Sensor Network (WSN) deployed on the Rivers Ribble and Dee in England and Wales. Each GridStix node (illustrated in Figure 2.1a) consists of depth and flow sensors in which power is supplied by batteries, replenished by solar panels. Nodes are equipped with both 802.11b (Wi-Fi) and Bluetooth communications for inter-node data transmission and with a GSM uplink node. In addition to the different transmission and data collecting modes, each node can be activated or deactivated according to the power level of the corresponding battery and the state of the river. Based on the information collected from GridStix nodes, a flood warning application considers a stochastic model for predicting flooding situations. It may also perform adaptation actions, such as activating or deactivating nodes for battery saving according the node's power level and neighborhood nodes' status (preventing low quality observation on some portions of the river). In order to support such a dynamic adaptation of the configuration of the WSNs, the system has to be aware of changes in the nodes' context and to respond in a reactive and proactive manner. This implies not only data acquisition, by collecting data from GridStix sensors, but also transferring this data and making it available for processing stochastic models. It also implies specifying adaptation policies that state the actions required to adapt the running system to a configuration that better fits its current context (e.g. change a node with a low battery to a neighboring node with a full of battery and turning off a node to save battery power).

Another scenario that we would highlight is the one considered by the project CC-Sem[1], whose goal is to develop an integrated platform for smart monitoring, controlling, and planning of the energy consumption and generation in urban scenarios. This project considers that the capabilities of monitoring/controlling/managing the energy consumption and generation are very important when

[1] https://www.fing.edu.uy/inco/grupos/cecal/hpc/cc-sem/

implementing the smart city paradigm. In this scenario, one may consider the use of smart electricity meters for collecting consumption information from homes, as well as other sensors for collecting information such as temperature, humidity and weather conditions. Collected data could then be analyzed in order to identify patterns of energy consumption. Such patterns may be used as the basis for recommendation purposes on smart home controllers, suggesting economy actions for final users (e.g. reducing air conditioning or heating intensity, based on weather information), but also for preventive actions such as turning off water heaters and air conditioning systems in case of system overload. They could also be used by electric grid administrators and energy providers in order to better predict consumption and anticipate preventive actions for preventing problems due to over consumption. Carrying out such a scenario demands not only the deployment of intelligent energy meters and temperature/humidity sensors, but also the deployment of an appropriate infrastructure. Such an infrastructure is necessary for collecting and transferring raw data, as well as for analyzing it using Big Data techniques. It also implies dealing with privacy and security issues necessary for keeping personal consumption data safe and secure.

(a)  (b) 

Figure 2.1. *(a) A GridStix node taken from [55]. (b) A prototype of hydric stress monitoring system.*

Finally, a third scenario we would like to highlight is the application of IoT to the smart agriculture domain. Indeed, the use of sensors and actuators opens new perspectives for the agriculture. By using different sensors, such as humidity, hydric stress, luminosity and temperature sensors, it is possible to better monitor the overall state of health of plants and production. Such monitoring activity can be used as the basis not only for decision-making actions, but it can also trigger preventive actions automatically. Small cultures, like flowers, tomatoes, strawberries or spices, often deployed over greenhouse structures, may benefit from a constant surveillance of temperature and hydric conditions. Data observed from sensors directly located in the field can be used for decision making: producers may receive daily reports about their crops and decide preventive actions for saving or improving production. Such data may also be used for taking actions automatically, controlling, for instance, the water supply of a given position of the crops according to the plants' hydric stress. Figure 2.1b shows a very small prototype of a hydric stress monitoring system in which an Arduino nano-computer is used to monitor hydric stress, thanks to a soil moisture sensor, and to control a water pump accordingly. Commercial systems similar to this prototype are already available for domestic users, like the Daisy system[2]. Nevertheless, deploying this kind of equipment for a professional crop demands considering not only the infrastructure and energy supply, but also tackling challenges such as choosing the most suitable sensors and data to be monitored, choosing the necessary frequency among data collection

[2] http://daisy.si/

according the crop and environment needs, choosing the appropriate thresholds for triggering actions, or choosing how to represent/store collected data for better reporting and analyzing. The quality of decision making (automatic or not) depends on these issues, from the selection of data to be observed up to analysis.

This scenario also illustrates the ambiguity that may characterize context-aware applications. One may easily wonder if this scenario corresponds to a context-aware application or to a self-adapting one [16]. What is the difference, if there is a difference, between those? Should all the collected data be considered to be context data or just application data? For non-expert designers these questions appear to be hard to answer and even to understood, notably because the notion of context and context-awareness are complex and hard to understand without the necessary knowledge. Promoting this knowledge to non-expert designers is necessary if we want to stimulate the development of such new context-aware applications.

2.2. *Illustrative target group*

The number of applications exploring the notion of context is constantly increasing thanks to the growing development of sensor technologies now often embedded into smartphones, tablets or associated with IoT devices. Before being considered as context-aware, applications such as those in Section 2.1 can be seen by consumers and non-expert designers as "smart" applications, since they propose what can be perceived as an "intelligent" behavior with data collected from the environment. Indeed, being able to observe the physical environment, to dynamically adapt its behavior without any human intervention are behaviors commonly perceived as "intelligent" (or "smart") by consumers, and consequently, more and more developers are considering integrating such behavior into their new applications.

Even if these applications are becoming more and more popular, their design and implementation is often poorly understood or has not been mastered sufficiently by non-expert designers. In order to illustrate this situation, we have invited two groups of master's degree students in computer science to participate in a survey. In this survey, students are invited to answer a set of 50 questions about their practices on designing/engineering "intelligent" software applications. These questions consider different aspects concerning the project, the design and the development of such smart applications. Both groups, containing about 20 students each, were composed, in their majority, by students in apprenticeship, with 1 to 3 years of experience, and for the others about 4-6 months of internship, both on software (mainly Web) development or Information Systems.

Before submitting the students to the survey, we gave them a small experiment using a RaspberryPi nano-computer (see Figure 2.2). Students had to build a small application that observed temperature in the room and reacted to the observed temperature through a LED (red LED if temperature is greater than a threshold, yellow if it is lower, and green otherwise). Students were organized into small groups. Each group received a kit containing one RaspberryPi ZeroW, a temperature sensor I2C BMP 280 and a triple LED (see Figure 2.2), as well as a SD card containing the OS Raspbian Jessie and two examples of code for handling the sensor and the LED. It is only after connecting the sensors and building the application that students received the survey for completion.

Although all the students concerned had a computer science background and an academic background in software development, only half of them (about 48.83%) had declared having already participated in a project developing a smart application. Among those that had not yet participated, about 63.6% were planning (or would have liked) to participate in such kind of project. Among those with previous experience, mobile and Web applications appeared to be the most commonly concerned, counting for respectively about 66.67% and 42.85% of those students, followed by location-aware applications and smart home applications (concerning each 28,57%)[3]. Independently of their previous

[3] Students could select more than one category of application.

experience, almost all students declared to know platforms commonly used for context-aware applications: 100% declared to know RaspberryPi and Android Phone, 86% for iPhone, 83% and 81% for iPad and iPad Pro, 81% for Android tablets and 48.83% for Arduino. A similar tendency appeared when considering platforms they had already used or wanted to use: 93% declared to use/want to use the Android phone as a target platform, 72% for RaspberryPi, 67.44% for Android tablets and iPhone, 30.23% for Arduino and even 11.62% for SmartTV platforms. These numbers illustrate quite well the growing interest these young developers have for creating applications for mobile and IoT platforms, and for exploring the possibilities of these platforms.

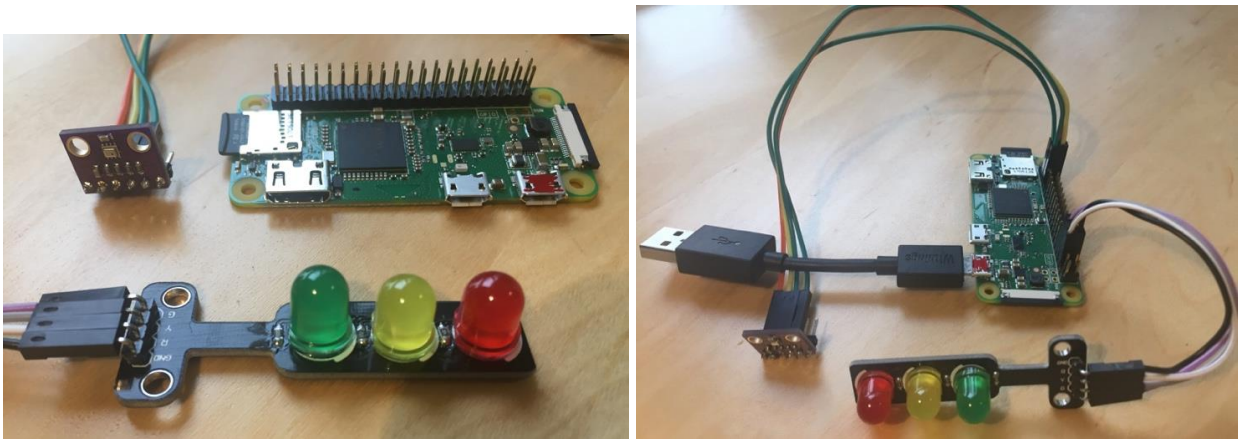


Figure 2.2. *Kit distributed to the students containing a RaspberryPi ZeroW, a I2C BMP 280 temperature sensor and a triple LED.*

It is worth noting that all these students had already been briefly introduced to the notion of context and to Pervasive Computing during their scholarship. Although the notion of context was not totally unknown to these students, they still could not be considered as "experts" in the domain, being mostly novices in this area. When asked about programming platforms (libraries, APIs, framework or middleware), almost no particular technology for context-aware and IoT application was cited. Only 4 students cited Pi4J, the library proposed in the examples used in the experiment they performed with the RaspberryPi. Among those with previous development experience, 36.36% declared using an ad-hoc direct access to the devices and 31.81% declared using the OS/programming language calls/API. The multiple platforms and technologies proposed in the literature (e.g. [18] [13] [22] [53] [61]) seem to remain unknown and unexplored for these students.

Similarly, when we asked students with previous development experience in "smart applications" when, during the previous project(s), the technologies were chosen, about 22.72% couldn't answer ("I don't know") as much as "at the very beginning, it was predefined in the project specifications". When asked about the difficulty of using these technologies, about 31.81% of the students evaluated the difficulty as "medium", pointing out that some elements were unknown and that they had to learn how to handle these technologies. It is interesting to note that no student evaluated the difficulty as "easy" and only about 9% declared it as "hard", pointing out that they had never used such technologies before. In any case, about 90% of students with previous experience declared that having previous knowledge about these technologies is necessary (50% considered that "it helps a lot" and 40% chose "it could help, but it is not mandatory"). Better dissemination of the knowledge about these technologies seems to be an interesting means for the progress of context-aware applications.

Finally, to the question "have you already heard about context-aware computing?", almost 52.27% of the students answered "yes, I have some notions on it" and 22.72% answered "yes, very slightly", which corresponded to the expected values since both groups of students already had some academic background in this topic. Still, when asked about whether the application they had built or wished to build was "context-aware", about 36.36% said "I don't know". Similarly, when asked if they were

familiar with the notion of "context", about 43.18% answered "yes, slightly", 25% declared to overcome the concept and 25% assumed to have some difficulty with it. Nevertheless, when asked how this notion could correspond to their past/future application(s), only 34.09% of the students could answer this question. Through these elements we may observe that the understanding of the notion of context and its use on software applications seems to remain a challenge for these computer science students, even for those with some previous experience on smart applications.

Students like those who participated in this survey illustrate the public we are focusing on here: software designers who, despite some experience, are not necessarily experts on the design and on the development of context-aware applications, but who could be led to participate in this kind of project in the near future.

3. Context engineering roadmap

As one may observe from the scenarios in Section 2.1, engineering context-aware applications can become a difficult task due to the complex nature of the notion of context as well as the different aspects that should be taken into account for considering this notion. Engineering such applications implies taking into account different aspects involving the notion of context and its support, which include collecting, transferring and analyzing context data in dynamic environments. Non-expert designers, such as the students cited in Section 2.2, were left alone to understand and identify the necessary concepts and components for building such applications. Acquiring the necessary knowledge for developing software applications exploring this notion thus represents a challenge for non-expert designers.

In [31], we tackled this question by considering multiple dimensions necessary for supporting context on software applications in a context engineering roadmap. This roadmap represents, for us, a first step towards a global approach, allowing us to better grasp the different aspects involved in the management of context information. It considers different challenges related to context management, organized along multiple dimensions. Each dimension focuses on different aspects and tackles different issues necessary to context management.

Additionally, in [32] we extended this roadmap in order to take into account quality aspects on context management. Indeed, the roadmap proposed in [31] does not consider the influence of quality, limiting its analysis to a few dimensions, such as the acquisition of context data. Still, managing the quality of context information demands a deeper reflection about the consequences of quality on the application behavior, and its influence over every aspect of context management. With the growing development of context-aware applications in multiple domains (healthcare, smart homes, transport, etc.), the importance of managing Quality of Context (QoC) is also growing, since the consequences of low-quality observations on the application behavior might be dramatic, or at least, may seriously affect the application reliably. These consequences can be illustrated when considering the smart agriculture scenario (cf. Section 2.1): low quality observations from malfunctioning or non-functioning soil moisture sensors may lead to erroneous decisions considering irrigation, exposing the production to the consequences of a too abundant or too sparse irrigation. According to [58], context information is naturally dynamic and uncertain: it may contain errors, be out-of-date or even incomplete. The quality of the information collected by a given sensor may vary according to several different and possible unpredictable factors (e.g. weather and wind conditions, failures of energy supply, communication interferences, etc.), leading to erroneous, incomplete or missing information. Uncertainty being indissociable from context information, handling quality of context becomes a central concern for reaching the reliability that is mandatory for the development of context-aware applications in the near future.

Since the quality of context information may potentially affect all dimensions represented by the roadmap, it should be considered as a transversal concern affecting all aspects of context management.

Again, the aforementioned scenario on smart agriculture illustrates this point quite well: errors in the information collected, as well as in the threshold definition may affect the water supply to crops and lead to an excessive or an insufficient water provision, which will in its turn influence the production.

In this paper, we propose a unified view of [31] and [32], which is enriched with a literature review (cf. Section 4), covering challenges and solutions related to every dimension pointed out by the roadmap. In the next few sections, we describe the context engineering roadmap, with its proposed dimensions, as well as the quality dimension, proposed as a transversal plan affecting all previous dimensions.

3.1. *Context support dimensions*

As briefly illustrated by the scenarios in Section 2, several kinds of software application may use the notion of context. Most known applications are probably context-aware applications, which use this notion for adaptation purposes, adapting the behavior of the application accordingly. Nevertheless, adaptation is not the only possible purpose of using context information on a software application. This notion may be explored in several ways, with different implications on the application design and behavior. Whatever the purpose of using context information, handling such information means dealing with different aspects related to its management, from its observation up to its use on a software application. For instance, deciding modalities and means for data collection and transfer on Grid Stix and CC-Sem scenarios (cf. Section 2.1), choosing threshold on smart agriculture one, defining adaptation mechanisms on GridStix or data analysis methods for CC-Sem, are just a few examples of issues that should be considered when developing these scenarios. Identifying these challenges and issues related to the management of the notion of context on software applications becomes mandatory for supporting the growing development of new intelligent applications using this notion. We believe that considering context information through its multiples challenges may contribute to a global approach for designing such applications, by allowing a better understanding of the different aspects involved in the management of context information.

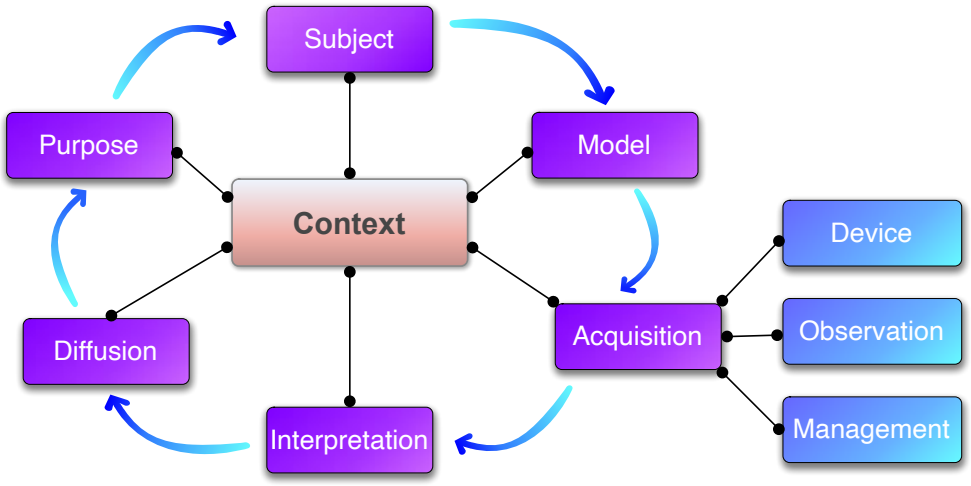


Figure 3.1. *Context engineering roadmap [31].*

Indeed, engineering context-aware applications involves tackling several challenges involving the notion of context and its support. By analyzing several existing works applying this notion on software applications, we could identify in [31] the most relevant characteristics of context management required by context-aware systems and organize them according to six dimensions, represented in Figure 3.1: purpose, subject, model, acquisition, interpretation and diffusion. Each dimension identifies challenges and issues, leading to the identification of functional and non-functional goals that should be considered and satisfied (at least partially) by these applications. These dimensions do not necessarily follow a particular order. As demonstrated by [2], projects developing intelligent software

applications using the notion of context do not follow a singular process; the adopted process may change according to the team and the project itself. Nonetheless, in order to simplify the presentation of the roadmap, we will discuss the proposed dimensions respecting the order represented in Figure 3.1.

The first dimension we consider is the Purpose dimension, which focuses on the purpose of using context information in a given application, and on the meanings and mechanisms for reaching it. This dimension considers why a given application needs context information. Such a purpose has a significant impact on how this information is exploited and consequently on what information will be considered, how it will be acquired, represented and analyzed, which are issues considered by subsequent dimensions of the roadmap. For instance, considering the scenarios presented in Section 2, two main purposes can be highlighted: the adaptation purpose, such as in the third scenario, in which the water supply is automatically adapted according to humidity conditions; but also, decision making, like in CC-Sem scenario, in which information collected about energy consumption can be used by power grid administrators for decision making.

The second dimension in this roadmap is the Subject dimension. It focuses on what information could be considered as context and how to identify relevant elements. This is not a trivial question, since the notion of context corresponds to a large and often ambiguous concept [42][7][25], potentially referring to very different elements, whose relevance depends on the use we will make of it. The subject dimension is directly connected to the first dimension, since the purpose of using context information in an application will influence the relevance (or not) of a given information for that application.

Multiple definitions have been proposed for the notion of context [7] [42] [25]. One of the most commonly accepted considers context as any information that can be used to characterize the situation of an entity (a person, place, or object) that is considered relevant to the interaction between a user and a system [21]. This definition points out both an observed entity (e.g. the user) and a piece of information that is observed about this entity. The entity delimitates the observation: the aim is to observe a given entity, but when looking at this entity, different elements can be observed. For instance, when considering a user (i.e. an entity), it is possible to observe his location, his mood, his level of expertise, etc.; when considering a device (i.e. another entity), it is possible to observe its available memory, network connection, etc. Thus, the entity corresponds to the subject of the observation. It plays a central role in context modeling, as pointed out by [17] [9], since it is precisely the context of this subject that is currently been observed. Everything we observe is related to this subject [34]. He we call the information observed (location, memory, etc.) about a given entity/subject the context element. For [34], when observing such context elements, we obtain *values* corresponding to their current status that will probably evolve over the time. For instance, by observing the *context element* ‘location’ for a *subject* ‘user’, we may obtain *values* for latitude and longitude, corresponding to the current user’s location. Similarly, in the smart agriculture scenario, the *context element* ‘temperature’ of a land parcel, representing our *entity*, can be estimated through multiple *values* obtained from a temperature sensor. Figure 3.2 represents a meta-model proposed by [34] in which context information corresponds to a set of context elements that are observed for a given subject (entity) and for which multiple values can be dynamically observed.

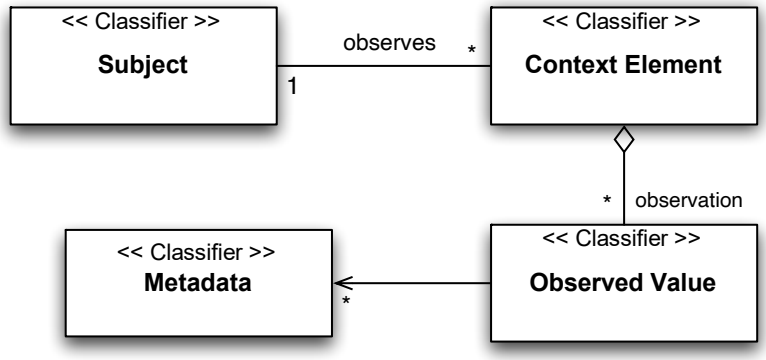


Figure 3.2. *Context meta-model from [34].*

Identifying what information should be considered as a context element in a given application means identifying what entities and context elements should be considered as relevant. This relevance depends on the identified purpose. Context information is observed in order to satisfy this purpose. Its relevance depends on its contribution to this purpose. For instance, in GridStix scenario (cf. Section 2), considering that the goal of this system is the flooding prediction, the main entity to be considered is the river and more precisely its state. The latter is determined by observing the depth and the flow rate of the river at different points. Nevertheless, the GridStix nodes themselves can also be considered as a possible entity since it is necessary to observe their battery level and communication capabilities (Bluetooth and Wi-Fi connections) for self-adapting the overall infrastructure in order to reduce the energy consumption.

This identification process remains a significant challenge, as identified by [21] [25] [2], notably because any information that can be used to characterize something (an entity or the subject in the metamodel in Figure 3.2) can be potentially considered as context. The main aspect to be taken into account remains its relevance, if it is relevant for characterizing a given entity, considering the system purpose. For instance, information related to the battery level can be considered as relevant in the GridStix scenario because it supposes adapting node behavior according to this. In the CC-Sem scenario the same information can be ignored since its domestic use allows us to suppose a continuous power supply. According to [2], since context is a crucial element that defines the functionality of a context-aware system and shapes its behavior, context selection becomes a significant task in the design of these systems. For these authors, system designers need to anticipate the relevant combinations and characteristics of context before implementing the system, and to decide which context to include in their design.

However, just identifying relevant context information is not enough, it is also necessary to consider how to represent this information in an appropriate manner. The Model dimension focuses precisely on context modeling issues. Its main goal is to determine the most appropriate representation for context information on a given application according to its identified purpose. Context information must indeed be represented, internally, in a software application, in such a way that it becomes practical and possible for this application to explore it and to realize its purpose. As pointed out by [34], a context model ensures the definition of independent adaptation processes and isolates this process from context acquiring techniques. An inappropriate model may compromise, or at least make more complex, the implementation of a given application. For instance, in the GridStix scenario, the adoption of an appropriate context model (e.g. an object-oriented model as suggested in [31]) allows, on the one hand, the definition of adaptation rules (notably for turning on and off a given node) independently of precise sensors or APIs used for obtaining the data. On the other hand, an inappropriate model (e.g. a too complex model) may negatively impact the performance of the flooding prediction system since data analysis by stochastic models may demand extra processing of the available data. Several approaches of context modeling exist, from key-value sets and object-oriented models up to complex ontologies

[42] [6] [12] [33] [54] [24]. Simple models, such as key-value ones, will be easy to implement but will offer no particular reasoning mechanism. On the contrary, ontology-based models will be more complex to implement, but they will allow complex reasoning mechanisms.

Representing context information is a challenging issue due to the nature of this information. Firstly, context information can be heterogeneous. Since different kinds of context element can be observed, the information obtained may vary from numeric information, like GPS coordinates or a percentage (e.g. CPU load), to symbolic values (e.g. the role of a user in a group). Such heterogeneity can be observed on both content and data structure. For instance, in the GridStix scenario, multiple elements can be considered (as pointed out in [31]), such as the flow rate and depth of the river (the observed entity), and the battery level and Bluetooth state from the nodes. Most of these consider numeric values, followed by a timestamp (meta-data describing the observation), except the Bluetooth state, which can be represented by a symbolic value ('on' or 'off'). However, in other scenarios, context elements with a more complex structure can be observed. For instance, when considering location information, multiple representations are possible, such as GPS coordinates or postal address. Both representations are composed of multiple values (at least latitude and longitude for the former, street name and number, locality, zip code, country, etc. for the latter). It is the context model that organizes and structures data obtained from sensors and other data sources into valuable information that can be explored by the application.

Furthermore, context information is naturally dynamic, varying among observations. For instance, in our hydric stress scenario (cf. Section 2), observed values for humidity levels will vary between observations according the plants' consumption and weather conditions. This dynamicity must be supported by the context model, which should keep values associated with context elements and assume that these values will vary over time. Indeed, we may consider that, by definition, context is about characterizing the situation of an entity that is (or may) be constantly evolving.

Finally, context information is also uncertain and often incomplete or presenting errors and imprecisions [58] [14] [13] [37], mainly due to problems during the acquisition of data (connection problems, interferences, etc.), resulting in erroneous or missing data. For instance, in the CC-Sem scenario, the weather conditions and notably exposing temperature sensors to direct sunlight may adulterate the quality of the measurements. Similarly, in the GridStix scenario, the river conditions (e.g. an important flow in a flood period) may damage GridStix nodes, causing missing data on portions of the river. This uncertainty represents an important issue since this data influences the behavior of a context-aware application, making the quality of context a very important issue when considering context support on software applications. The influence of quality concerns on the context management is not limited to only context modeling, representing a transversal concern. We will discuss this concern and its influence on all dimensions of the roadmap in Section 3.2.

It is worth noting that these three aspects (heterogeneity, dynamicity and uncertainty) profoundly characterize context information. Handling these aspects is a key factor for successfully exploiting context information in a software application. More than in traditional applications, managing context data implies handling these aspects as a priority.

Context information should be acquired by observing the environment around a given entity. The Acquisition dimension highlights the challenges of acquiring context information from the environment, which implies considering the capture devices, the observation process and the management of context sources. Indeed, in order to correctly capture context information from the environment, one has to observe this environment, using an appropriate acquisition device. The main challenges here are the heterogeneity and the interoperability of these devices since their nature can be quite variable. For instance, a user's location can be acquired using a GPS device or calculated from a Wi-Fi connection; temperature data, necessary for CC-Sem and smart agriculture scenarios, can be observed using very different kinds of sensors, which are not necessarily interoperable. Such

heterogeneity makes it more complex to support different context elements in a given application, as well as the evolution of existing applications (e.g. observing new context elements or modifying the acquisition device), since changing the observation device may seriously affect the application code or design. This is particularly true in large deployment scenarios, such as CC-Sem and smart agriculture ones. In these cases, in which a large number of sensors are to be deployed, it is particularly important to support multiple models of sensors (e.g. temperature sensors on CC-Sem scenario) and to easily replace one sensor with another similar one (e.g. a soil moisture sensor by another humidity sensor). Thus, when considering the acquisition dimension, it becomes imperative to consider how to isolate the software application and its behavior from the precise technology used for acquiring context information, as underlined by authors such as in [21] [6].

Moreover, using a given acquisition device for observing the environment means fulfilling the context model with observed values. The Observation process should consider not only the device used for this, but also the observation frequency, according to the expected dynamicity of the observed information. For instance, location information will demand an active observation in order to guarantee some accuracy, while the user's role can be acquired on-demand. Once observed, this information also has to be kept updated in order to represent the current context of the observed entities. For instance, in the smart agriculture scenario (cf. Section 2), the information obtained from a soil moisture sensor must be regularly updated in order to keep track of changes on the plants' hydric stress level. These updates should be frequent enough to satisfy plants' hydric requirements, but too frequent observations will probably be inefficient or even useless since the plants' current hydric situation will not change drastically over a very short period of time (e.g. several seconds). The acquisition dimension also involves managing the environment. The environment itself being dynamic, the availability of devices used for observation is not guaranteed. Some devices may disappear (e.g. being switched off) and others may join the environment, becoming available for capturing the context of a given entity. The management of this dynamic environment is also a challenge, considering the evolution of the environment and the availability of the acquisition devices in it. The GridStix scenario is a good example of this management, since GridStix nodes can be switched on and off, coming in and out the system, according to battery conditions.

Data collected during the acquisition process corresponds to raw data that often have to be aggregated or interpreted in order to be better exploited by context-aware applications. The Interpretation dimension focuses on this issue, considering the challenges related to the interpretation of context information in its different forms (interpretation rules, context mining, etc.). It considers how to transform raw context data on useful knowledge for a given application. For instance, when considering the smart agriculture scenario, a soil moisture sensor has been used in the prototype illustrated in Figure 2.1b for evaluating the humidity level. Raw data offered by this sensor corresponds in fact to impedance values observed on the soil parcel around the sensor. This raw data is compared to predefined thresholds in order to deduce the parcel humidity level. The goal of this dimension is then to specify appropriate interpretation mechanisms and to consider necessary reasoning and aggregation mechanisms that can be applied according to the capabilities of the context model. Different interpretation mechanisms can be considered, from ad-hoc reasoning up to complex rule-based systems [54][23]. Those mechanisms cannot be dissociated from the context model. Not only will the context model limit the possibilities of interpretation (i.e. a key-value structure will offer fewer reasoning opportunities than an object-oriented or an ontology-based model), but the information that will also be deduced from the interpretation mechanism will also feed the context model, similar to an acquisition mechanism, building for instance high level information from raw data.

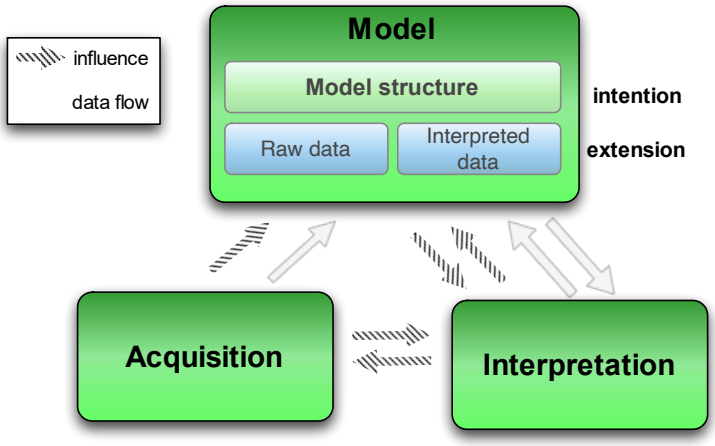


Figure 3.3. *Influence and data flow among Model, Acquisition and Interpretation dimensions.*

The three dimensions, *Model*, *Acquisition* and *Interpretation*, are then intrinsically connected and cannot be dissociated, as illustrated by Figure 3.3. Firstly, they are connected because the mechanisms on these dimensions exchange data, represented by the data flow in Figure 3.3: acquisition mechanisms feed context model with raw data, which is also consumed by the interpretation mechanism, whose results will again feed the model extension (i.e. instances). Secondly, these three dimensions influence choices about each other (as illustrated by the influence arrow in Figure 3.3): acquisition devices may influence the interpretation mechanisms that can be applied (for example, gyroscope and accelerometer data that can be interpreted on a user's movement information thanks to statistical methods such as in [51]); conversely, the interpretation mechanism can influence the selection of acquisition methods (for instance, triangulation methods can be used to deduce a location from GSM-based data instead of GPS, as in [48]). Similarly, decisions about interpretation and acquisition may influence the model, both its intention (i.e. structure) and extension (i.e. instances), and the model will guide and limit interpretation possibilities (for instance, rule-based mechanisms, such as [23], will be hard to apply without an ontology-based model, and statistical methods such as Bayesian networks applied on [51], often require a numeric representation of data).

Finally, the Diffusion dimension explores the issues related to the transmission of context information among multiple nodes. Indeed, context-aware applications often behave as distributed applications, in which multiple nodes communicate and exchange information about their current state. In some cases, the context information should be distributed from the node in which it is observed to a different node, in which it will be processed, interpreted or stored. For instance, in the GridStix scenario (cf. Section 2), context information about the river and the nodes' conditions is transmitted for remote processing, for a flooding warning application. Similarly, in the CC-Sem scenario (cf. Section 2), context information should be transferred from acquiring devices to an appropriate computational infrastructure allowing the data analysis of the energy consumption. Several challenges arise from this distribution, above all, the stability of the context information (how long does a given piece of information remain valid and useful after being transferred from a different node?) and the coherence of the collected data, since contradictory data can be reported from multiple nodes observing a given entity (e.g. multiple temperature sensors observing different values for a given room according to external influences such as sunlight or heating).

In addition to the dimensions considered by the roadmap in Figure 3.1, another concern must be considered: the quality of context (QoC). As one may observe, the roadmap presented here does not consider the influence of quality in depth. In our opinion, managing the quality of context is not only a matter of correctly representing the context information or its meta-data. It demands a deeper reflection on the influence of QoC on all aspects of context management. For us, quality should be a transversal

concern affecting all dimensions of the context management. In the next section, we discuss this influence, extending what we introduced in [32].

3.2. *Quality on context support*

As introduced earlier, Quality of Context (QoC) is a transversal concern that influences all aspects of the context management. According to [5], Quality of Context is usually defined as the set of parameters that express quality requirements and properties for context data (e.g., precision, freshness, trustworthiness...). Being able to observe and handle these properties requires the consideration of their influence on every aspect of context management, transforming QoC concerns on a transversal plan, affecting all other dimensions of the context engineering roadmap, as represented in Figure 3.4. In every dimension, particular challenges are considered and somehow influenced by the notion of quality. When considering each dimension, we should consider the influence of taking QoC into account on supporting the dimension challenges. This influence is materialized in Figure 3.4 by the verbs attached to each dimension of the roadmap, representing a guideline when considering QoC on the dimension. Thus, for each dimension, we tried to identify in [32] questions, represented through the verbs in Figure 3.4, that should be considered when thinking about the influence of QoC. Similar to the context engineering roadmap presented earlier (cf. Section 3.1), the main goal of this proposal is not necessarily to give solutions to these questions, but mainly to raise discussion on the impact of QoC on context management.

The first dimension, Purpose, considers the purposes for which the notion of context is used in an application. According its purpose (e.g. adaption, decision making, etc.), an application can be more or less sensitive to errors or low-quality context information, since errors on context information may lead to erroneous decisions from the application, which may be more or less important, according to the application domain. For example, let us consider a healthcare application that proposes to automatically adapt insulin levels according to a patient's blood sugar levels or to call the emergency services if a patient falls over. Reliability of this kind of application depends on the quality of context observed since erroneous information may lead to a wrong decision with significant consequences for the patient's health. The same can be assumed in the smart agriculture scenario (cf. Section 2), in which errors on context information could directly affect water supply and consequently production.

The question raised by the Purpose dimension is essentially whether to and how to *follow* QoC in a given application. The management of QoC should consider the consequences of a poor-quality context information and the consequences of having no information about it. These consequences should be considered but also the costs of managing QoC. For example, still considering the smart agriculture scenario, errors on humidity data may lead to an excessive water provision. Similarly, a problem affecting the humidity data provisioning (i.e. missing context information) may lead to an insufficient water provision. Both cases can be very harmful for the plants and affect productivity. Furthermore, reaching application purposes implies several mechanisms that are potentially affected by QoC. Including QoC in these may represent a cost that should be considered. Will the cost of observing QoC be more or less important than the risk of not observing it? For instance, in a healthcare application, considering QoC in the adaptation process implies using different algorithms for detecting and eliminating suspicious measures. Such algorithms will consume processing and battery power in the hosting device. These represent an execution cost, in addition to design and development costs. Even if these can be significant compared to overall application costs, the risks and the possible consequences of not considering QoC justify these costs. The application developer should then consider the risks and the costs that will follow QoC observation on the application purpose. However, one question arises from this dimension: how can we measure such risks and costs? This point remains an open question.

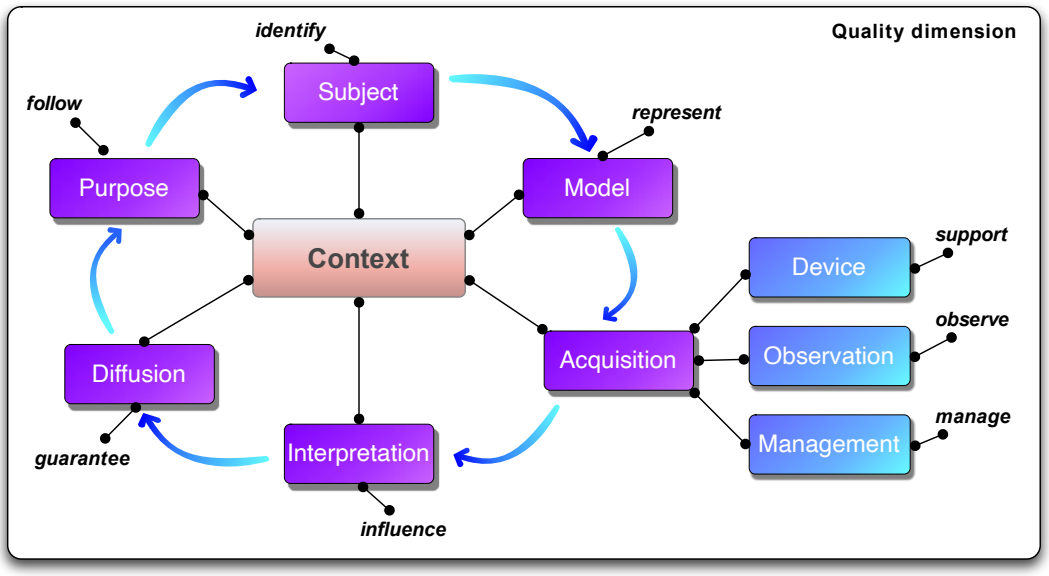


Figure 3.4. *Quality plan on the context engineering roadmap [32].*

The remaining dimensions may give us some insights about the costs and risks of observing (or not) QoC. The Subject dimension considers what kind of information can be observed as context information. When identifying relevant context information, one should also *identify* possible QoC indicators that can be associated with it. Often, QoC consists of several elements such as precision, up-to-dateness, freshness or probability of correctness [37][5]. Identifying what context information will be observed allows application developers to identify either what quality information is relevant to be associated with it for reaching the application's purpose. For instance, when considering location information, different quality indicators can be considered, such as estimated error or precision, freshness (which can be obtained regarding production time) or even the number of available satellites, when considering GPS data. Indeed, several QoC criteria are possible, as illustrated by [37]. These authors have identified and compared different QoC indicators proposed in the literature, highlighting the variation in terms and in meaning of these criteria. Similar to context information itself, the relevance of a given indicator often depends on the use that will be made of it. Again, the purpose of a system highly influences the relevance of context and QoC information. Both are observed considering this purpose, their contribution with its satisfaction determines their relevance. For instance, considering the CC-Sem and hydric stress scenarios presented in Section 2, both may consider accuracy as a QoC indicator, but their needs considering this indicator will not be the same, errors would be better tolerated on the first scenario than on the latter.

All identified information should be represented in an appropriate context model. Considering QoC on the Model dimension implies considering how to *represent* QoC information, how it will be associated with observed context. Several research works have been carried out on context models [6][7][8] [42], and multiple proposes have already considered the question of QoC [14] [13] [37] [27], often through meta-data representing QoC indicators. As summarized by [6], a good context modeling approach must include modeling of context information quality to support reasoning about context.

It is impossible to consider context information without considering its acquisition. The same can be said about QoC information. The Acquisition dimension considers this question through three different points of view (or sub-dimensions): device, observation and management. QoC will influence the analysis of each of these sub-dimensions. First, when considering the necessary devices for acquiring context information, it is also important to consider if these devices are able to *support* acquiring QoC indicators. Similarly to context information, QoC indicators are calculated based on information supplied by acquiring devices. The possibility of obtaining a given QoC indicator depends on the capability of these devices offering basis information. For example, when considering GPS data, the

number of satellites can be used for supposing data precision. If this information cannot be obtained for any reason, this criterion will be unavailable. In the hydric stress scenario presented in Section 2, one may consider using precision as a valuable QoC indicator. However, it is also necessary to consider how to obtain such an indicator, since most soil moisture sensors are unable to calculate this by themselves. A calibration phase is necessary, which may offer only an estimation of the sensor precision.

Similarly, data confidence can also be considered as a possible quality indicator. Considering, for example, a team application that constantly informs users about project context (and progress), information about task progression might heavily depend on the information supplied by the users themselves. In this case, it seems difficult to estimate data confidence and thus trustworthiness of this information. Furthermore, assuring quality of context information leads to choosing appropriate acquisition devices, and consequently, to the costs associated with such devices. For instance, on a smart home application, one may consider using ground sensors for detecting a resident's fall, since these devices may offer a better accuracy for fall detection than simple accelerometers. The costs associated with these devices are not the same, but they can be justified according to the application purposes (e.g. if the application is designed for supporting medical care or special needs residents). Software developers must be aware of these issues when considering their QoC indicators and the devices used to obtain them.

Furthermore, context observing policies are directly influenced by QoC considerations. For instance, considering if a given context information needs very frequent observation probably implies that freshness is a relevant QoC indicator for this information, and vice versa: high levels of freshness demand very frequent observations. This is often the case of location information on transport applications: when considering moving vehicles, the freshness of location information will indicate if this information can still be used or if new measures are necessary. Application developers must then consider how to *observe* QoC indicators during context observation process and how often this observation process should be realized. Finally, the management of acquiring infrastructure is also influenced by QoC. Observed environment being more and more dynamic, it is necessary to *manage* acquiring devices on this environment. This management can be influenced by QoC indicators (for example, deactivating a given device if precision offered by it is too low or reactivating it in order to increase overall system precision). This example is visible on GridStix scenario mentioned in Section 2, in which GridStix nodes containing flood and depth sensors are turned off in order to save battery and on again in order to guarantee that every portion of the river has enough sensors observing it, improving the quality of overall observation system. It is then important to consider not only how to *manage* QoC information, but also how to manage acquisition environment according to QoC information.

Similar to previous dimensions, the Interpretation dimension is also influenced by QoC considerations. Quality of context information may affect interpretation and reasoning mechanisms and influence the reliability of the target application. An illustration of this influence is given by [60]. In this work, the authors discuss a set of metrics evaluating QoC and propose using such metrics on context prediction, in order to prevent low quality information affecting prediction mechanism. Works such as [60] demonstrate the importance of considering how QoC *influences* context reasoning and interpretation, and how these reasoning mechanisms can explore QoC information for better results. Indeed, it is worth noting that interpretation mechanisms may consume both context data and QoC data. Context being dynamic and uncertain, context data and their quality indicators will evolve over time, making possible different historical and statistical analysis (and then interpretation). For instance, the analysis of outliers and weak signals on context values and on QoC indicators (e.g. means, accuracy, precision, etc.) may contribute to predicting new tendencies and anticipating actions. Weak signals are usually seen as abnormal values or "information on potential change of a system to an unknown direction" [39]. The analysis of such signals can be particularly interesting in some scenarios, revealing undiscovered events or phenomena. In the CC-Sem scenario (cf. Section 2), the analysis of

weak signals and outliers may contribute to the energy consumption prevision, revealing new tendencies on energy consumption (for example the presence of a new device) or forecasting variation due to seasonal climate changes. Similarly, in the smart agriculture scenario, the analysis of weak signals on mean values of temperature and soil moisture sensors may help in identifying possible misfunctioning sensors.

Finally, with the growing distribution of software applications, the distribution of context information over multiple nodes is becoming a necessary. For instance, in IoT scenarios, context information is often transferred to distant servers or cloud platforms for data analysis. In these cases, quality indicators such as latency or packet loss can significantly affect the information reliability. The Diffusion dimension considers the challenges related to the distribution of the context information. When considering QoC on such distributed environments it is important to consider whether this transmission may affect the quality of transmitted context information. According to [5], it is necessary to consider the quality of both the exchanged context data and the distribution process to ensure user satisfaction. Indeed, if the context data distribution is not aware of the data quality, possible service reconfigurations could be misled by low quality data. For instance, real time context information may be affected by network latency and become out-of-date. When considering, for example, an application such as [18] that deploys its components on remote nodes according to available resources, if information about these resources is outdated because of network latency, deployment decision may lead to user dissatisfaction and performance loss. It is then necessary for application developers to consider not only how to guarantee QoC information transfer, but also how to *guarantee* that this diffusion of context information will not affect QoC?

As illustrated in Figure 3.4, quality concerns affect all aspects of context management and consequently all kinds of software applications using this notion. Both the context engineering roadmap represented in Figure 3.1 (cf. Section 3.1) and its quality concerns illustrated in Figure 3.4 offer a multi-dimensional view of the multiple aspects of context management in software applications. It is worth noting that, as the variety of examples given in this section leads us to suppose, every dimension proposed in the roadmap in Figures 3.1 and 3.4 will not equally influence all kinds of applications. The relevance of each dimension depends on the purpose of the application itself and on the considered context information. It is then essential to consider and discuss each dimension, considering its possible relevance for an application and the influence of the quality concern on it. In this section, we raise questions about the support of context information on software applications and about the influence of quality concerns on it. More than solutions, the main goal here is to initiate discussion and to point out challenges and issues of context management and QoC on software applications.

4. Literature review: challenges and solutions

The main goal of the context engineering roadmap presented in Section 3 is to help non-expert designers to better understand challenges and issues related to the support of the notion of context on software applications. Some of the challenges suggested here have already be highlighted by previous works in the literature. A good example of this is the context lifecycle proposed by [46] (see Figure 4.1), which considers the movement of context data on software applications, focusing on context modeling, acquisition, reasoning and dissemination. All these steps are considered in the roadmap as dimensions (respectively modeling, acquisition, interpretation and diffusion), to which we have added the purpose and the subject dimensions. Indeed, the context engineering roadmap intends to represent a summary of the main challenges highlighted in the literature, such as [1][42][7][6][25][14][5][37][46]. In this section, we introduce a brief summary of the literature review in which the roadmap is based. The main goal here is to discuss solutions and open issues proposed in the literature for each dimension of the roadmap. This summary is a necessary complement to the roadmap for non-expert designers, giving concrete examples of the challenges considered on each dimension.

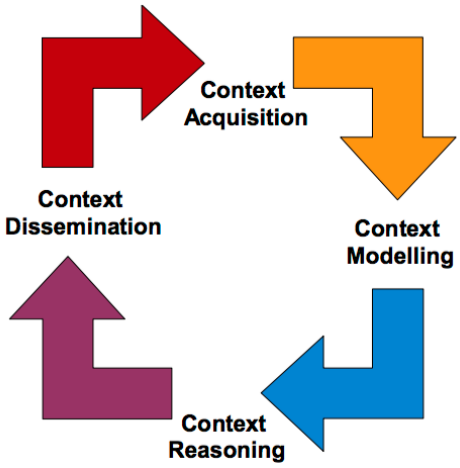


Figure 4.1. *Context life cycle proposed by [46]*

When analyzing the literature, a first class of applications advocating using the notion of context is represented by context-aware applications, which use context information for adaptation purposes. This adaptation may affect different aspects of the application behavior: one may adapt the content supplied to the user [56][15], the services offered by the application [58] [12], or even the application composition itself [49][18][22] according to the execution and the user's context. Still, as mentioned earlier, adaption is not the only purpose for using context information on software applications. In [14], the authors underline six common uses for context information: (i) *context display* (i.e. presenting context information to the user); (ii) *contextual augmentation* (i.e. annotating data with context information); (iii) *context-aware configuration* (i.e. configuring a service using context information); (iv) *context triggered actions* (i.e. triggering actions according to context information); (v) *contextual mediation* (i.e. modifying services and content to best match context of use); and (vi) *context-aware presentation* (i.e. adapting user interface or content presentation).

As one may observe, most of these uses refer to adapting application behavior (content, actions, services or interface) according to the context of use, but for [14], context information can also be used for annotation or simply displayed. Indeed, annotating data or objects using context information allows applications, services or even users to better characterize information or data, while displaying context may contribute to decision-making processes. For instance, [35] uses context information for characterizing fragments of methods on Method Engineering, while [47] [54] use context information for characterizing tasks on workflow models, and [33] associates context and group awareness information on Groupware Systems for helping users to better coordinate their actions. More recently, IoT applications are being considered for massively observing information from the environment, opening new perspectives for the use of this context information on many different applications. Works on IoT and smart cities [46][40][50][64], as well as projects such as CC-Sem mentioned in Section 2, illustrate this tendency quite well. They demonstrate the interest of using context information as a support for decision making as well as for adaptation.

When considering subject dimension, the literature illustrates quite well the variability of the notion of context. From initial works on context-aware computing, which mostly consider the user's location and device as the main context elements [56] [10], up to works considering complex situations on their behavior [58] [18], multiple visions of what can be considered as context are given. First of all, multiple definitions can be found in the literature [42]. For instance, [56] defines context as the location, the identity of nearby people and objects, and changes in those objects. Through this definition, these authors focus on three main questions: where, who is around and the surrounding resources. For [41], context refers to physical and social situation in which computational devices are embedded. This definition focuses on computational devices, delimitating this notion to its software perception, without considering precisely possible elements, keeping yet its generality. Currently, one

of the most accepted definitions is given by [21], which considers context as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves". Similar to [41], [21] examines the notion of context from a software background, by focusing on the interaction between a user and a system.

As one may observe from these definitions, the notion of context may cover several different elements. For instance, in [56][15][10], location is mainly considered, while in [22] [24] information about the execution environment, such as available memoire and network connection, is considered. In [49] [18], information about the executing device but also about surrounding devices is considered as context and used for better identifying the user's situation and deploying application over these devices consequently. On [51], data obtained from a smartphone accelerometer is used for activity detection, while in [48], location detected using GPS and GSM data, as well as connectivity information, is used for adapting content delivery on a health record application.

A similar question arises when considering QoC. Considering quality of context on a software application means considering what indicators or properties to use for measuring such quality. Several authors have highlighted different indicators for context information, as demonstrated by [37]. Among common indicators, we may cite precision, up-dateness, accuracy or freshness.

There is a consensus in the literature about the arduousness of defining what context elements or QoC indicators can be considered as relevant for a given application (e.g. [21][2][25][37]). This question of choosing elements remains an open issue. Even if cited works do not tackle this issue of selecting context information and QoC indicators directly, some of them [24][22][27][13][37] propose interesting solutions for supporting these elements during the design phases. For instance, [24] and [13] propose a MDE (Model Driven Engineering) approach for developing context-aware applications. Both properties corresponding to the observed context elements (for the first) or to the QoC indicators (for the later) are modeled using UML and other high-level model representations at the very beginning of the design process. By focusing on this modeling, these approaches help designers to better conceptualize the necessary elements for their software applications. On the one hand, freeing designers from implementation details at the very beginning of the project allows those to focus on concepts and then to consider more easily the necessary information for their applications. On the other hand, by producing code from these model specifications, such approaches help designers in the development task itself.

Furthermore, the literature also reveals other issues that follow from identifying relevant context information. The first one concerns the relationship among the observed elements. Context elements are not necessarily independent, and their relationship can also be relevant. This is notably the case of cooperative applications, in which group members, tasks and objects can be related in different ways. For instance, [33] considers that knowing that an activity is related to a group can be as important as knowing the group itself. These authors explore information about these relationships in order to characterize the relevance of given information to a user. In [23], these relationships are explored through rules, allowing reasoning about access rights on shared resources according to the user's context. Another issue is the granularity of the observed information. Some context elements can be broken down into lower-level elements or regrouped forming higher-level elements. Managing different levels of abstraction can be required by complex applications. For instance, in [18], the authors propose to aggregate low level information in order to describe a complex user's situations. Similarly, in [45], the authors propose a pluggable architecture that allows new context information to be composed based on lower level data. In both cases, this composition of more complex information based on lower-level context data can also be assimilated into the interpretation of raw data for producing new context information.

Another common topic in the literature concerns context modeling. Several works have tackled this question, proposing many different context models [42][6][7]. All these works highlight several challenges related to context modeling, and notably how to deal with the heterogeneity and dynamicity of context information, as well as with the uncertainty that characterizes context information. Indeed, QoC concerns strongly affect context modeling, since those models should include quality information in order to handle QoC concerns on application behavior. As suggested by [14] or [27], quality information should be part of the context model and cannot be dissociated from it, in a holistic view of context modeling. Furthermore, context modeling plays an important role in the application extensibility and evolution. Context models contribute to isolating application behavior from the acquisition technologies. By doing so, these models contribute to the possibility of evolution, allowing new context information to be easily considered in the application without demanding large recoding efforts from the developers.

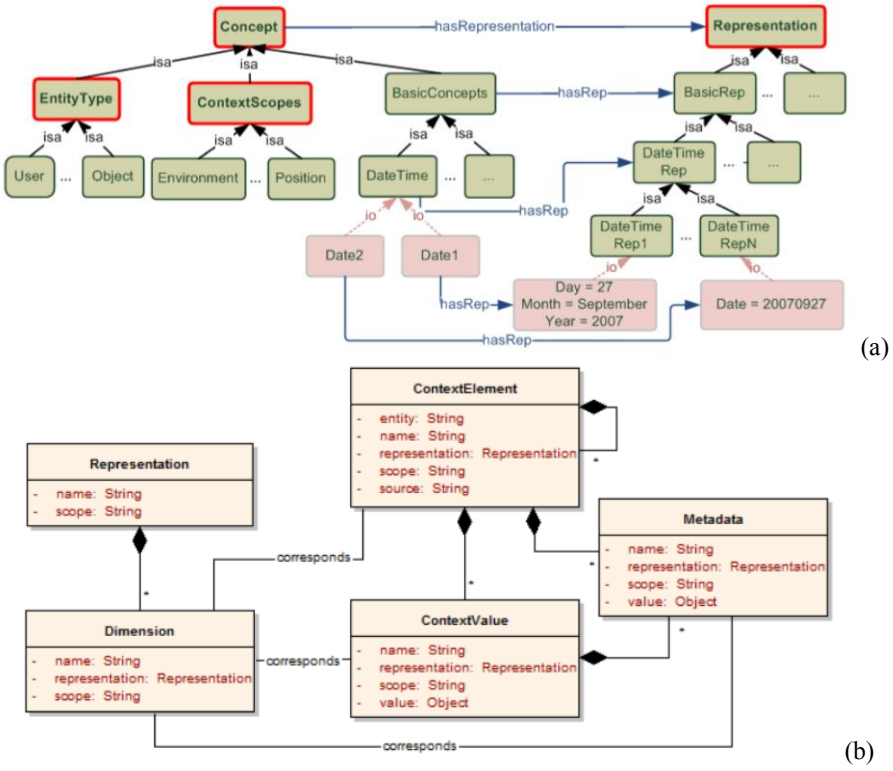


Figure 4.2. *Object oriented context meta-model (a) and context ontology (b) proposed on [62].*

As underlined by [42][6], different kinds of context models have been proposed in the literature. They vary from simple "key-value" models up to complex ontologies (e.g. [54]), passing by structured models (XML or RDF based, for instance [36]) and object-oriented models (e.g. [33]). These multiple modeling paradigms involve different degrees of complexity, both in implementation and execution. They also offer different reasoning capabilities, which may include ad-hoc processing, statistical and data analysis techniques or complex rule-based reasoning. For instance, in [33] an object-oriented model allows the representation of different information from a user's physical and organizational context. Organizational elements are also considered by [54], which use an ontology-based model for representing context information (time, location, availability, etc.) of entities related to a business process (actors, resources, etc.). In [62], different paradigms are combined. First, a context ontology, illustrated in Figure 4.2a, allows significant concepts related to context information to be described, including context entities (the user, an object, etc.), information elements about it (e.g. location) and their representation. Then, an object-oriented meta-model, resuming the same concepts, is used mainly for implementation purposes (see Figure 4.2b). Finally, an XML schema representing these concepts is used for exchanging context information among different computing nodes.

Modeling is also an important concern when considering QoC. In [14], a context model is proposed considering in particular the uncertainty of context information. This is performed notably through the definition of relations allowing context information to be compared under uncertainty. In [27], authors analyze the effects of quality on a context model. They consider a MDE approach for context modeling, proposing a DSL (Domain Specific Language) for creating context models. Through this language, it is possible to describe different context elements and its data sources, as well as more complex situations combining different context elements. Similarly, [13] [37] have also considered a MDE approach, focusing particularly on the modeling and support of QoC indicators. In [37], the authors have proposed a QoC meta-model in which quality indicators are associated with context information. Each QoC indicator has a set of associated values and it is defined by a QoC criterion containing a set of defined metrics, allowing a full definition of each QoC indicator, from the concept definition to its metrics.

As mentioned before, context models play an important role in isolating the application's behavior from the technology used for acquiring context information. By doing so, context models also contribute to hiding the heterogeneity of the acquisition devices. Indeed, many different devices can be used for acquiring a single context element. Maybe the most common example of this is location information, which can be observed using different devices and methods (GPS, GSM-based estimation, Wi-Fi triangulation, etc.), but the same can be said about other context elements. For instance, temperature can be observed using several different models of sensors. Not all these sensors necessarily offer the same support according to the programming language, libraries and platforms. As an illustration, the website HomeAutomation.org[4] compares about 5 different temperature sensors for Arduino, with different C programming codes for accessing each one. Code and process necessary to capture data from sensors is not necessarily the same according to the sensor model. This heterogeneity makes the acquisition process more complex and can seriously compromise the interoperability among acquisition devices.

Several research works have considered this issue, proposing mechanisms for isolating applications from the heterogeneity of the environment and consequently allowing a better interoperability among different acquisition devices. Among these, we may cite the Context Toolkit proposed by [21]. This toolkit isolates the application itself from the acquiring technology through different abstractions, and notably the notion of the context widget, which encapsulates the access to the physical device. The application only has to handle these abstractions, no direct knowledge about the physical device is necessary. This knowledge is concentrated inside each context widget, which offers a standard interface for the application, improving interoperability, from the application point of view. A similar approach is assumed in [45], which proposes a pluggable architecture in which context plugins are dynamically loaded according to the application needs. Again, applications are not directly faced with the real physical environment, keeping contact only with their context plugins, which provide a standard interface for accessing context information. All the interaction with the real environment is confined in the context plugin, reducing the application's complexity and improving interoperability and possibilities of code reuse.

Less discussed in the literature but still important, the acquisition of quality information suffers from a similar issue. Indeed, not all sensors offer the necessary information or metrics for quality indicators, and often software designers have to consider other means to calculate or estimate such indicators. For instance, considering temperature sensors like those mentioned by HomeAutomation.org[3], these sensors may easily be influenced by external factors (e.g. exposition to sunlight or to heating/cold source), which may lead to significant errors in acquired data. In the experiment we performed with the students (cf. Section 2.2), we used five temperature sensors of the model I2C BMP 280 in a single room (about 35 m^2) and obtained up to 2~3°C of difference in the perceived temperature among the

[4] http://www.homautomation.org/2014/02/18/arduino-temperature-sensor-comparison/

sensors. Unfortunately, sensors like BMP 280 do not offer natively meta-data or any quality information allowing an application to automatically calculate QoC indicators. Calculating QoC indicators such as precision demands, in this case, extra equipment or processes for calibration. This issue of how to obtain or estimate QoC indicators remains until now an open question left in charge of the application designer.

The same can be affirmed about the observation process. This process as well as the QoC considerations concerning it have not received the same attention in the literature as other dimensions considered in the roadmap. One of the reasons motivating this could be the dependency between this process and the application itself. Indeed, this process greatly depends on the application needs considering the context information and its freshness or up-dateness. Nonetheless, most of the middleware solutions for context management, such as [22][45][24], offer push (a.k.a. publish-subscribe) and pull mechanisms for capturing context data from the environment. Such mechanisms represent the basis for successful observation policies. Nevertheless, the definition of these policies is left under the responsibility of the application designers, according to the application needs.

Furthermore, one of the main challenges of context-aware applications is the dynamicity of pervasive environments in which these applications are supposed to execute. This is the main aspect analyzed by the management dimension. By its own nature, context data is in constant evolution. Context is a dynamic construct as viewed over a period of time, episodes of use, social interaction, internal goals, and local influences [25]. Context is not simply the state of a predefined environment with a fixed set of interaction resources. It is part of a process of interacting with an ever-changing environment composed of reconfigurable, migratory, distributed, and multiscale resources [17]. Dynamicity is intrinsic to the notion of context and it should be taken into account properly when considering it on a software application. It should be considered through appropriate models and acquisition mechanisms, but also in the management of the surrounding environment. Indeed, this environment cannot be supposed to be static. By definition, context-aware applications must consider a dynamic environment, in which execution conditions may vary, users and devices may move, resources may come in or disappear at any moment. Such a dynamic environment leads to multiple challenges and notably service and resource discovery. As resources move or change their current state, the composition of the surrounding environment and its available resources also change, making the ability of discovering and managing such surrounding resources a necessity. Multiple works have considered this issue, such as [52] which proposes a dynamic binding and component discovery mechanism for service component architectures. Moreover, this question also has be considered on other domains, such as the grid systems [43], proposing interesting solutions that could also be applied for handling the dynamicity of pervasive environments.

The dynamicity of pervasive environments also impacts the diffusion of context information. Diffusion of context information to other nodes is often necessary mainly when considering adapting applications to the surrounding available resources, such as [18], in which context information about surrounding nodes is used for adapting application deployment. As underlined by [5], multiple approaches for distributing context information have been considered, including centralized approaches, peer-to-peer and hybrid or hierarchical ones. We may cite, for instance, [59] and [19]. The former proposes to organize the distribution of context information on dynamic groups, which regroup nodes presenting a similar situation. Groups are established dynamically based on distribution policies, which indicate context information that can be shared among group members and the common context elements that define group membership. The latter propose a hierarchical solution based on a SIP communication protocol for sharing context information among members of a community.

Most of these context distribution approaches are now faced with an IoT environment, which imposes an important requirement: scalability. Indeed, in an IoT environment, the number of pieces of equipment may grow exponentially [44], demanding distribution mechanisms to be able to scale up. Many of the approaches underlined by [5] are unable to scale up and potentially handle hundreds of

nodes, such as considered in IoT scenarios. Hybrid architectures, using hierarchical approaches, such as [53] or architectures combining IoT and cloud infrastructures such as [40] [61], are increasingly being considered.

The use of cloud computing infrastructures leads us to consider the persistence of context information and its access control policies. Unfortunately, these questions remain marginal in the literature compared with their importance for a user's privacy. One possible reason explaining this is the fact that context information is often supposed to be consumed in "real time": it is the context of a given entity (user, object, service, etc.) at this particular moment. In this case, storage and historical analysis are not necessarily taken into account. However, this is about to change with the massive adoption of cloud-based solutions for storage and the application of data mining techniques for context analysis, such as in [51] [29]. The question of security and access control for context information will also become more and more relevant in the next few years. Nowadays the number of works considering these issues is not proportional to their relevance. Only a few works have, for instance, focused on how to control the access to context information produced by a given node. For instance, [20] proposes access policies based on the XACML standard using a FOAF (Friend Of A Friend) approach. We hope that, with the development of IoT technologies, these questions will in the near future receive the attention they need in the literature.

Finally, the interpretation dimension has been considered in the literature through several different approaches, according to the application purposes. Indeed, as mentioned in Section 3, context data is often acquired as raw data that must be interpreted to become information or knowledge. Different ways to proceed to such an interpretation are possible, such as the possibility to aggregate low level data on more complex context information. The previously mentioned context plugins proposed by [45] represent an interesting mechanism for aggregating context data in a transparent way: the application may handle aggregated context information in the same way as it handles raw data through these context plugins.

More sophisticated interpretation mechanisms are also possible. For instance, [33] and [58] consider similarity measures for analyzing and comparing context information, while [23] considers rule-based systems for deducing new information from context information. In all these cases we may observe the relationship between interpretation mechanisms and context models: none of these proposals would be applied without an appropriate context model. For instance, rules proposed by [23] are possible thanks to the ontology-based model adopted by these authors. Other complex mechanisms, not necessarily based on ontology-based models, can also be cited. In [18] a workflow mechanism is used in order to deduce complex situations from the context data, while in [55] context data is used with constraint programming in order to control application self-adaptation according to environment changes.

Furthermore, a new tendency towards interpretation can be observed: the mining of context information. The idea is to apply data mining techniques on context information for different purposes: to discover missing information [51][57], to anticipate context evolution [38], or to determine the relevance of a context element on a given system [29]. For instance, [51] considers different statistical methods, and notably Bayesian networks, for analyzing accelerometer and gyroscope data and identifying a user's movement related situation (e.g. walking, running, etc.). In [38], the authors propose using Markov chains in order to deduce the next context information and thus to anticipate a user's possible situation. Finally, [29] analyzes the relevance of context elements in the use of a given application by crossing context and application use data using Formal Concept Analysis (FCA). All these mining techniques may contribute to context-aware applications by allowing them to assume a more predictive behavior, anticipating pervasive environment evolution.

All the works cited in this section contribute somehow to the support and management of context information in software applications. They illustrate the challenges and possible solutions for different issues considered by the roadmap dimensions. Through this literature review we intend to contribute to

a better understanding of these challenges, complementing the roadmap dimensions by concrete examples of these dimensions in action. Table 4.1 summarizes what has been discussed here, associating dimensions identified in the context roadmap with their key concepts and the main questions that raise these dimensions, as well as examples from the literature review.

Dimension	Key concepts	Questions	Examples
Purpose	Adaptation Quality of Context (QoC)	Why use context information? What is the purpose of using this information in the application? How will this information be explored? Why should the application follow QoC indicators? Is QoC relevant enough for the application?	[11] [12] [15] [18] [22] [49] [56] [58] [33] [50] [35] [47] [54]
Subject	Context information QoC indicator	What information is considered context? How can we identify it? What information is needed for the application? What quality indicators can be used?	[2] [3] [10] [48] [51] [56] [37]
Model	Context model QoC metric & models	How can we internally represent context and QoC data? How can we structure this data? How can we handle heterogeneity, dynamicity and uncertainty in this representation?	[6] [24] [33] [36] [54] [62] [13] [14] [27] [37]
Acquisition	Sensor devices Acquiring platform QoC measurement	How can we acquire context and QoC data? What acquisition devices can be used? How and how often should data be collected? How can we manage the environment and its devices? How can we support devices' heterogeneity? Are QoC indicators supported by the devices?	[21] [18] [22] [45] [52]
Interpretation	Reasoning Context mining	How can we interpret context and QoC data? How can we produce new context and QoC data from low level data? How can we apply a reasoning/analysis method on available data? How can we take into account QoC during interpretation?	[14] [18] [23] [29] [38] [51] [55] [57]
Diffusion	Context distribution Scalability	How can we transfer context and QoC data among nodes? May this transfer affect QoC? How can we guarantee the reliability of these data during transfer? How can we ensure data validity and coherence? How can we manage scalability when the number of nodes grow? How can we ensure privacy and data access policies?	[5] [19] [20] [44] [53] [59]

Table 4.1. *Context management dimensions with their main questions and some related examples.*

5. Discussion & conclusions

The notion of context has been widely explored in different ways through software applications. This use is likely to grow in the next few years with the development of IoT technologies, which allow

applications to observe the physical environment using low cost devices. Nevertheless, the notion of context remains an obscure and ambiguous concept. What information can be considered as context, what information cannot, is a justified question for software developers. Information such as available memory, battery level or user's role can be considered as context for some [45][24][33], and as a simple application parameter for others [28][55]. This is also perceptible through some answers in the survey we performed (cf. Section 2.2). For example, a student that has declared having previous experience with "smart" applications and "mastering" the topic of context-aware computing assumed context in his/her application as "keywords" spoken by the user to "activate a feature". Clearly, an element that can be perceived as a simple input parameter and not necessarily as "context" by context-aware computing literature, is being considered as such in this case. Some authors, such as [11], tried to bring a distinction between context data and application data. For these authors, context data corresponds to a set of parameters, which are external to the application and that influence the behavior of the application. Despite the efforts to clarify this distinction, the boundary between context and application data remains blurred, as well as the notion of context itself, which remains often unclear for software developers.

There is an ambiguity not only on what can be considered as context, but also on what is a context-aware application and the possible distinctions between those and self-adapting applications. The smart agriculture and GridStix scenarios illustrate these ambiguities. In the former, context information can be easily seen as application data, while the latter was considered in [55] as a self-adapting application. According to [30], the concepts of context awareness and self-adaptation are often sources of confusion because self-adaptive applications often adapt their behavior based on the context stimuli and therefore it is often difficult to make a clear distinction between these two concepts. Indeed, both can be considered as adaptive systems, which, according to [16], aim to achieve a certain *goal* through the definition of some form of loop whereby the environment and/or the system itself is *monitored*, the information gathered is *analyzed*, a *decision* is taken as to what change is needed in response, and these changes are then *enacted* in the system. For these authors, 'self-awareness' means that changes can often be handled automatically compared with conventional systems that require off-line re-design, implementation and redeployment, which is also true for context-aware systems since they automatically adapt their behavior according to context changes without any particular human intervention. Some authors have tried to establish some distinction between these concepts [16] [30]. Nevertheless, the most important question here is not actually these potential differences (if they really exist), but the support of these systems. Both are very complex and poorly known concepts for non-expert designers. Both base their behavior on a very dynamic and complex kind of data. Whenever we call it context or not, the management of such dynamic data raise several challenges that should be considered when developing such a system. While these challenges remain misunderstood by non-expert designers, the possible distinctions between context-awareness and self-adapting, between context data and application data will remain irrelevant faced with these challenges.

The main question is not whether or not a given piece of information can be considered as context, but how to support and manage it in a software application. As pointed out by [17], it is commonly agreed that context is about evolving structured and shared information spaces, and that such spaces are designed to serve a particular purpose. Whatever information is considered as context profoundly depends on the application and on its purposes, and whatever this information could be, it is necessary to support it appropriately. This support requires understanding the challenges that it implies and the main characteristics of context information, such as its heterogeneity, its dynamicity and its uncertainty. The main goal of the roadmap discussed in this paper is precisely to contribute to this understanding.

The roadmap presented in this paper analyzed these different challenges of context management, organized through six dimensions. Each dimension has considered a precise aspect of context management and its quality concerns. Through this roadmap, we hope helping non-expert designers to better understand the challenges of supporting context information in software application. Indeed, in

our opinion, the main challenge now may not only be dealing with the remaining unsolved issues in this support, but perhaps it is about acquiring the necessary knowledge for developing new applications. Non-expert software developers, when developing context-aware applications, are faced with a very complex concept, the understanding and management of which is far from simple, as the multiple dimensions of the roadmap and the literature review we presented demonstrate. With the development of IoT and connected devices, and their integration into our everyday life, it is becoming essential to form a new generation of software developers that are able to reason about context support and its challenges, including quality concerns. More than just technical solutions (which are still necessary), we also need to go further towards context engineering approaches, offering a global approach to understand the context notion in software development. Raising questions and discussion about context management and the impact of Quality of Context on it is, for us, an important step towards a true context engineering approach.

Finally, we are strongly convinced that, as underlined by [17], context is key in the development of new services that will impact social inclusion for the emerging information society. More than ever it is important to encourage a better understanding of the notion of context and its support in young non-expert software designers in order to incite the development of new services and applications using this notion in a responsible way.

Acknowledgments

The authors would like to thank the students from the MIAGE Sorbonne master's degree program for accepting to participate to our survey. We would like to thank Dr. Raul Mazzo, from Université Paris 1 Panthéon Sorbonne, for his inestimable help with the GridStix scenario and testing the roadmap, as well as Dr. Luiz Angelo Steffenel, from Université Reims Champagne-Ardenne, for his also inestimable help on the CC-Sem project and on the setup of the students' experiment.

Bibliography

[1] BALDAULF M., DUSTDAR, S., ROSENBERG, F., « A survey on context-aware systems », *Int. J. of Ad Hoc and Ubiquitous Comp.,* vol. 2-4, 91-S46, p. 263-277, 2007.

[2] BAUER C., DEY A., "Considering context in the design of intelligent systems: Current practices and suggestions for improvement", *J. of Systems and Software*, vol. 112, p. 26-47, 2016, Elsevier.

[3] BAUER, J. S., NEWMAN, M. W., KIENTZ, J. A., "Thinking About Context: Design Practices for Information Architecture with Context-Aware Systems", *iConference 2014 Proceedings*, p. 398–411, 2014, doi:10.9776/14116.

[4] BELL G., DOURISH P., "Yesterday's tomorrows: notes on ubiquitous computing's dominant vision", *Personal and Ubiquitous Computing*, vol. 11, p. 133-143, 2007.

[5] BELLAVISTA, P., CORRADI, A., FANELLI, M., FOSCHINI, L., "A survey of context data distribution for mobile ubiquitous systems". *ACM Comput. Surv.*, vol. 45, p. 1–49, 2013.

[6] BETTINI, C., BRDICZKA, O., HENRICKSEN, K., INDULSKA, J., NICKLAS, D., RANGANATHAN, A., RI-BONI, D., "A survey of context modelling and reasoning techniques", *Pervasive and Mobile Computing*, vol. 6 issue 2, p. 161-180, 2010.

[7] BRÉZILLON, J., BRÉZILLON, P., "Context Modeling: Context as a Dressing of a Focus". In: Kokinov, B.; Richardson, D.; Roth-Berghofer, T. & Vieu, L. (Eds.), *Modeling and Using Context*, Springer, LNCS 4635, p. 136-149, 2007.

[8] BRÉZILLON, P., "Context-Based Development of Experience Bases". In: Brézillon, P.; Blackburn, P. & Dapoigny, R. (Eds.), *8th Int. and Interdisciplinary Conf. on Modeling and Using Context (CONTEXT)*, p. 87-100, 2013.

[9] BREZILLON, P., "Expliciter le contexte dans les objets communicants". In: Kintzig, C., Poulain, G., Privat, G., Favennec, P.-N. (Eds.), *Objets Communicants*, Hermes Science Publications, p. 293–303, 2002.

[10] BROWN, P.; BOVEY, J. & CHEN, X., "Context-aware applications: from the laboratory to the marketplace", *IEEE Personal Communications*, vol. 4 issue 5, p. 58-64, 1997.

[11] CHAARI, T., LAFOREST, F., CELENTANO, A., "Adaptation in context-aware pervasive information systems: the SECAS project", *Journal of Pervasive Computing and Communications*, vol. 3-4, 2007.

[12] CHAARI, T.; DEJENE, E.; LAFOREST, F.; SCUTURICI, V.-M. "Modeling and Using Context in Adapting Applications to Pervasive Environments". *IEEE Int. Conf. on Pervasive Services (ICPS'06)*, p. 111-120, 2006.

[13] CHABRIDON, S.; CONAN, D.; ABID, Z.; TACONET, C. "Building ubiquitous QoC-aware applications through model-driven software engineering". *Sci. Comput. Program.*, vol. 78, p. 1912–1929, 2013.

[14] CHALMERS, D.; DULAY, N.; SLOMAN, M. "Towards Reasoning About Context in the Presence of Uncertainty". *1st Int. workshop on advanced context modelling, reasoning and management*. Nottingham, UK, 2004.

[15] CHEVEREST, K.; MITCHELL, K.; DAVIES, N. "The role of adaptive hypermedia in a context-aware tourist guide". *Communication of ACM*, vol. 45, p.47–51, 2002.

[16] COLMAN, A., HUSSEIN, M., HAN J., KAPURUGE, M., "Context Aware and Adaptive Systems". In: Brézillon, P., Gonzalez, A. J. (Eds.), *Context in Computing*, Springer, p. 63-82, 2014.

[17] COUTAZ, J.; CROWLEY, J.; DOBSON, S., GARLAN, D., "Context is the key", *Communications of the ACM*, vol. 48 n° 3, p. 49-53, 2005.

[18] DA, K.; ROOSE, P.; DALMAU, M.; NEVADO, J.; KARCHOUD, R. "Kali2Much: a context middleware for autonomic adaptation-driven platform". *Proceedings of the 1st Workshop on Middleware for Context-Aware Applications in the IoT (M4IoT@Middleware 2014)*, p. 25–30, 2014.

[19] DEVLIC, A., & KLINTSKOG, E., "Context retrieval and distribution in a mobile distributed environment". In Proceedings of the *Third Workshop on Context Awareness for Proactive Systems (CAPS 2007)*. Guildford, UK. 2007.

[20] DEVLIC, A.; REICHLE, R.; WAGNER, M.; KIRSCH PINHEIRO, M.; VANROMPAY, Y.; BERBERS, Y., VALLA, M., "Context inference of users' social relationships and distributed policy management", *IEEE Int. Conf. on Pervasive Computing and Communications (PerCom 2009)*, p. 1-8, 2009.

[21] DEY, A., « Understanding and using context », *Personal and Ubiquitous Computing*, vol. 5 issue 1, p. 4-7, 2001.

[22] FLOCH, J., FRÀ, C., FRICKE, R., GEIHS, K., WAGNER, M., LORENZO, J., SOLADANA, E., MEHLHASE, S., PASPALLIS, N., RAHNAMA, H., RUIZ, P.A., SCHOLZ, U., "Playing MUSIC: building context-aware and self-adaptive mobile applications". *Softw.: Pract. Exp.*, vol. 43, p.359-388, 2013.

[23] GARCÍA, K.; KIRSCH-PINHEIRO, M.; MENDOZA, S.; DECOUCHANT, D. Ontology-Based Resource Discovery in Pervasive Collaborative Environments. In: Antunes, P., Gerosa, M.A., Sylvester, A., Vassileva, J., and de Vreede, G.-J. (eds.), *19th Int. Conf. on Collaboration and Technology (CRIWG 2013), LNCS 8224*. Springer, p. 233–240, 2013.

[24] GEIHS, K., REICHLE, R., WAGNER, M., KHAN, M.U., "Modeling of Context-Aware Self-Adaptive Applications in Ubiquitous and Service-Oriented Environments", In: B.H.C. Cheng, R. de Lemos, H. Giese, P. Inverardi, J. Magee (Eds.), *Software Engineering for Self-Adaptive Systems, Lecture Notes in Computer Science, 5525*, p. 146-163, 2009.

[25] GREENBERG, S. "Context as a Dynamic Construct". *Human-Computer Interact*., vol. 16 issue 2, p. 257–268, 2001.

[26] HAGRAS H., "Intelligent pervasive adaptation in shared spaces", In A. Ferscha (Ed.), *Pervasive Adaptation: Next generation pervasive computing research agenda*, p. 16-17, 2011.

[27] HOYOS J.R., PREUVENEERS D., GARCÍA-MOLINA J.J., "Quality Parameters as Modeling Language Abstractions for Context-Aware Applications: An AAL Case Study", In: Brézillon P., Turner R., Penco C. (eds), *Modeling and Using Context. CONTEXT 2017. Lecture Notes in Computer Science, vol. 10257*, p. 569-581, 2017.

[28] HUGHES, D.; GREENWOOD, P.; BLAIR, G.; COULSON, G.; GRACE, P.; PAPPENBERGER, F.; SMITH, P. & BEVEN, K., "An Experiment with Reflective Middleware to Support Grid-based Flood Monitoring", *Concurr. Comput.: Pract. Exper.,* John Wiley and Sons Ltd., vol. 20 issue 11, p. 1303-1316, 2008.

[29] JAFFAL, A., LE GRAND, B., KIRSCH PINHEIRO, M., "Refinement Strategies for Correlating Context and User Behavior in Pervasive Information Systems", *Int. Workshop on Big Data and Data Mining Challenges on IoT and Pervasive Systems (BigD2M 2015), 6th Int. Conf. on Ambient Systems, Networks and Technologies (ANT-2015)*, Procedia Computer Science, vol. 52, p.1040-1046, 2015.

[30] KHAN, M.U., "Unanticipated Dynamic Adaptation of Mobile Applications", PhD thesis (Doktor der Ingenieurwissenschaften), University of Kassel, German, 31 March 2010.

[31] KIRSCH-PINHEIRO, M., MAZO, R., SOUVEYET, C., SPROVIERI, D., "Requirements Analysis for Context-oriented Systems", *7th Int. Conf.on Ambient Systems, Networks and Technologies (ANT 2016), Procedia Computer Science*, vol. 83, p. 253-261, 2016.

[32] KIRSCH-PINHEIRO, M., SOUVEYET, C., "Quality on Context Engineering". In: Brézillon P., Turner R., Penco C. (Eds.) *Modeling and Using Context. CONTEXT 2017. Lecture Notes in Computer Science*, vol 10257, p. 432-439, 2017.

[33] KIRSCH-PINHEIRO, M., GENSEL, J., MARTIN, H. "Representing Context for an Adaptative Awareness Mechanism", In: de Vreede G.-J.; Guerrero L.A.; Raventos G.M. (Eds.), *X Int. Workshop on Groupware (CRIWG 2004), LNCS 3198*. Springer-Verlag, p. 339-34, 2004.

[34] KIRSCH-PINHEIRO, M., RYCHKOVA, I., "Dynamic Context Modeling for Agile Case Management", In: Y.T. Demey and H. Panetto (Eds.), *OTM 2013 Workshops, LNCS 8186*. Springer-Verlag, p. 144–154, 2013.

[35] KORNYSHOVA, E.; DENECKÈRE, R.; CLAUDEPIERRE, B. "Towards Method Component Contextualization". *IJISMD*, 2, p. 49–81, 2011.

[36] LEMLOUMA, T., LAYAÏDA, N., "Context-Aware Adaptation for Mobile Devices". *5th IEEE International Conference on Mobile Data Management (MDM 2004)*, p. 106-111, 2004.

[37] MARIE, P. DESPRATS, T., CHABRIDON, S., SIBILLA, M., "The QoCIM Framework: Concepts and Tools for Quality of Context Management". In: Brézillon, P. & Gonzalez, A. J. (Eds.), *Context in Computing: A Cross-Disciplinary Approach for Modeling the Real World*, Springer New York, p.155-172, 2014.

[38] MAYRHOFER, R., HARALD, R., & ALOIS, F., "Recognizing and predicting context by learning from user behavior". In. W. Schreiner, G. Kotsis, A. Ferscha, & K. Ibrahim (Ed.), *Int. Conf. on Advances in Mobile Multimedia (MoMM2003)*, p. 25–35, 2003.

[39] MENDONÇA, S., PINA E CUNHA, M., KAIVO-OJA, J., RUFF, F., "Wild Cards, Weak Signals and Organizational Improvisation". FEUNL Working Paper No. 432, 2003. Available at SSRN: https://ssrn.com/abstract=882123

[40] MIORANDI, D.; SICARI, S.; PELLEGRINI, F. D. & CHLAMTAC, I., "Internet of things: Vision, applications and research challenges", *Ad Hoc Networks,* vol. 10 issue 7, p. 1497-1516*,* 2012.

[41] MORAN, T., DOURISH, P., "Introduction to this special issue on context-aware computing", *Human-Computer Interaction*, vol. 16 issue 2-3, p. 87-95, 2001.

[42] NAJAR, S., SAIDANI, O., KIRSCH-PINHEIRO, M., SOUVEYET, C., NURCAN, S. "Semantic representation of context models: a framework for analyzing and understanding" In: J. M. Gomez-Perez, P. Haase, M. Tilly, and P. Warren (Eds), *Proceedings of the 1st Workshop on Context, information and ontologies (CIAO 09), European Semantic Web Conference (ESWC'2009)*, p. 1-10, 2009.

[43] NAVIMIPOUR, N.J., RAHMANI, A.M., NAVIN, A.H., HOSSEINZADEH, M. "Resource discovery mechanisms in grid systems: A survey". *J. Netw. Comput. Appl*., vol. 41, p. 389–410, 2014.

[44] PARIDEL, K., YASAR, A., VANROMPAY, Y., PREUVENEERS, D., & BERBERS, Y., "Teamwork on the road: Efficient collaboration in VANETs with context-based grouping". *Proceedings of The Int. Conf. on Ambient Systems, Networks and Technologies (ANT-2011)*, Procedia Computer Science, vol. 5, p. 48-57, Elsevier, 2011.

[45] PASPALLIS, N., ROUVOY, R., BARONE, P., PAPADOPOULOS, G.A., ELIASSEN, F., MAMELLI, A. A Pluggable and Reconfigurable Architecture for a Context-Aware Enabling Middleware System. In: Meersman, R. and Tari, Z. (eds.) *On the Move to Meaningful Internet Systems: Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, LNCS 5331*. Springer, p. 553–570, 2008

[46] PERERA, C., ZASLAVSKY, A.B., CHRISTEN, P., GEORGAKOPOULOS, D., "Context Aware Computing for The Internet of Things: A Survey". *IEEE Communications Surveys & Tutorials*, vol. 16 issue 1, p. 414-454, 2014.

[47] PLOESSER, K., RECKER, J., ROSEMANN, M., "Supporting Context-Aware Process Design: Learnings from a Design Science Study", In: zur Muehlen, M. & Su, J. (Eds.), *Business Process Management Workshops, BPM 2010. Lecture Notes in Business Information Processing, vol 66.* Springer, Berlin, Heidelberg, p. 97-104, 2010.

[48] PREUVENEERS, D., NAQVI, N.Z., RAMAKRISHNAN, A., BERBERS,Y., JOOSEN, W., "Adaptive Dissemination for Mobile Electronic Health Record Applications with Proactive Situational Awareness". *49th Hawaii Int. Conf. on System Sciences* (HICSS 2016), p. 3229-3238, 2016.

[49] PREUVENEERS, D.; BERBERS, Y. "Context-driven migration and diffusion of pervasive services on the OSGi framework". *Int. J. Auton. Adapt. Commun. Syst.*, vol. 3, p. 3–22, 2010.

[50] PUGLISI, S., MOREIRA, Á. T., TORREGROSA, G. M., IGARTUA, MÓ. A., FORNÉ, J., " MobilitApp: Analysing Mobility Data of Citizens in the Metropolitan Area of Barcelona", In: Mandler, B.; Marquez-Barja, J.; Mitre Campista, M. E.; Cagáňová, D.; Chaouchi, H.; Zeadally, S.; Badra, M.; Giordano, S.; Fazio, M.; Somov, A. & Vieriu, R.-L. (Eds.), *Internet of Things: IoT Infrastructures, 2nd Int. Summit, IoT 360° 2015, Part I*, Springer, p.245-250, 2016.

[51] RAMAKRISHNAN, A.K., "Support for Data-driven Context Awareness in Smart Mobile and IoT Applications: Resource Efficient Probabilistic Models and a Quality-aware Middleware Architecture" (Ondersteuning voor data-gedreven context-bewustzijn in intelligente mobiele en IoT applicaties: Hulpbronnenefficiënte probabilistische modellen en een kwaliteit-aware middleware architectuur), PhD thesis, Katholieke Universiteit Leuven, Belgium 2016.

[52] ROMERO, D., ROUVOY, R., SEINTURIER, L., CARTON, P., "Service Discovery in Ubiquitous Feedback Control Loops". In: Eliassen, F., Kapitza, R. (Eds.), *10th IFIP WG 6.1 Int. Conf. on Distributed Applications and Interoperable Systems (DAIS) / Int. Federated Conf. on Distributed Computing Techniques (DisCoTec)*, Lecture Notes in Computer Science, vol. 6115, p.112-125, Springer, 2010.

[53] ROTTENBERG, S., LERICHE, S., TACONET, C., LECOCQ, C., DESPRATS, T., "MuSCa: A multiscale characterization framework for complex distributed systems", *2014 Federated Conference on Computer Science and Information Systems*, Warsaw, p. 1657-1665, 2014.

[54] SAIDANI, O., ROLLAND, C., NURCAN, S., "Towards a Generic Context Model for BPM". *48th Annual Hawaii International Conference on System Sciences (HICSS'2015)*, Jan 2015.

[55] SAWYER, P., MAZO, R., DIAZ, D., SALINESI, C., HUGHES, D., "Using Constraint Programming to Manage Configurations in Self-Adaptive Systems". *Computer*, vol. 45 issue 10, p. 56-63, 2012.

[56] SCHILIT, B.N., THEIMER, M.M. "Disseminating Active Map Information to Mobile Hosts", *Network*, vol. 8, IEEE, p. 22-32, 1994

[57] SIGG, S., HASELOFF, S., DAVID, K., "An alignment approach for context prediction tasks in ubicomp environments". *IEEE Pervasive Computing*, vol. 9, issue 4, p. 90–97, 2010.

[58] VANROMPAY, Y., KIRSCH-PINHEIRO, M., BERBERS, Y., "Service Selection with Uncertain Context Information". Reiff-Marganiec, S. & Tilly, M. (Eds.), *Handbook of Research on Service-Oriented Systems and Non-Functional Properties: Future Directions*, p. 192-215, 2011.

[59] VANROMPAY, Y., KIRSCH PINHEIRO, M., BEN MUSTAPHA, N., AUFAURE, M.-A., "Context-Based Grouping and Recommendation in MANETs", In: Kolomvatsos, K.; Anagnostopoulos, C. & Hadjiefthymiades, S. (Eds.), *Intelligent Technologies and Techniques for Pervasive Computing*, Chapter 8, IGI Global, p. 157-178, 2013.

[60] VANROMPAY, Y., "Efficient Prediction of Future Context for Proactive Smart Systems", PhD thesis, Katholieke Universiteit Leuven, Belgium, 2011.

[61] VILLALBA, L., PREZ, J.L., CARRERA, D., PEDRINACI, C., PANZIERA, L., "Servioticy and iserve: A scalable platform for mining the IoT". *6th Int. Conf. on Ambient Systems, Networks and Technologies (ANT-2015)*, Procedia Computer Science, vol. 52, p.1022-1027, 2015.

[62] WAGNER, M., REICHLE, R., KHAN, M.U., GEIHS, K., LORENZO, J., VALLA, M., FRÀ, C., PASPALLIS, N., PAPADOPOULOS, G.A., "A Comprehensive Context Modeling Framework for Pervasive Computing Systems", *8th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS 2008), Lecture Notes on Computer Science, 5053*, p. 281-295, 2008.

[63] WEISER M., "The computer for the 21st century", *Scientific American*, vol. 265 issue 3, p. 66-75, 1991.

[64] ZAPPATORE, M., LONGO, A., BOCHICCHIO, M. A., ZAPPATORE, D., MORRONE, A. A., DE MITRI, G., "Towards Urban Mobile Sensing as a Service: An Experience from Southern Italy", In: Mandler, B.; Marquez-Barja, J.; Mitre Campista, M. E.; Cagáňová, D.; Chaouchi, H.; Zeadally, S.; Badra, M.; Giordano, S.; Fazio, M.; Somov, A. & Vieriu, R.-L. (Eds.), *Internet of Things: IoT Infrastructures, 2nd Int. Summit, IoT 360° 2015, Part I*, Springer, p. 377-387, 2016.

Annex II

Paper

Kirsch-Pinheiro, M.; Gensel, J. & Martin, H., “Representing Context for an Adaptative Awareness Mechanism”. In: Gert-Jan de Vreede, Luis A. Guerrero, Gabriela Marín Raventós (eds.), 10th International Workshop Groupware: Design, Implementation and Use, CRIWG 2004, LNCS 3198, 339-348 (2004)

Representing Context for an Adaptative Awareness Mechanism

Manuele Kirsch-Pinheiro, Jérôme Gensel, and Hervé Martin

Laboratoire LSR – IMAG
BP 72 – 38402 Saint Martin d'Hères Cedex, France
{Manuele.Kirsch-Pinheiro,Jerome.Gensel,Herve.Martin}@imag.fr

Abstract. The application of mobile computing technologies to Groupware Systems has enforced the necessity of adapting the content of information by considering the user's physical and organizational contexts. In general, context-aware computing is based on the handling of features such as location and device characteristics. We propose to describe also the user's organizational context for awareness purposes. Our objective is to permit Groupware Systems to better select the information and then to provide mobile users with some adapted awareness information. This paper presents a representation of this notion of context to be used by awareness mechanisms embedded in Groupware Systems. Then, we show how this representation is exploited for filtering the content of information inside the awareness mechanism.

Keywords: adaptability, awareness, mobile computing, cooperative work.

1 Introduction

In recent years, Groupware Systems have been using the Web to propose a world wide access to their users. With the massive introduction of web-enable mobile devices, such as laptops, PDAs and cellular phones, these users can access the system virtually everywhere. In fact, nowadays, workers may interact with their colleagues and accomplish some tasks (such as to compose messages or to exchange meeting notes), even when they are not at their office, and through a large variety of devices.

However, the use of this kind of mobile device introduces several technical challenges on system design and development, mainly due to the heterogeneity and the physical constraints (limited display size, power and memory capacity…) of these devices. These challenges make adaptation a necessary technique when building mobile systems [9]. In fact, a mobile user, whilst moving or alternating between different devices, often changes the context in which she/he accesses the system. Therefore, the ability of detecting the context of use seems to be particularly relevant for mobile computing systems, which may be used in different locations, by different users (who may access them from different devices), and/or for different purposes [3]. This detection of the context characterizes what is called context-aware systems. In fact, one of the core premises of context-aware computing is that the computing device should be aware of the user's circumstances and should be able to interpret any interaction in a manner appropriate to these circumstances [16].

Considering the notion of context, it is worth noting that there is no unique definition for this concept (see [2] as an illustration). For instance, according to Kirsh [11], "context is a highly structured amalgam of informational, physical, and conceptual

G.-J. de Vreede et al. (Eds.): CRIWG 2004, LNCS 3198, pp. 339–348, 2004.

resources that go beyond the simple facts of who or what is where and when to include the state of digital resources, people's concepts and mental state, task state, social relations, and the local work culture, to name a few ingredients". This notion of context can be related to the notion of awareness on Groupware Systems, which refers to the knowledge a user has and her/his understanding about the group itself and her/his colleagues' activities, providing a shared context for individual activities in the group (see, for example, [6] and [14]).

Nevertheless, when considering context-aware systems, we notice that the notion of context usually adopted by those systems is limited to some physical aspects, such as the user's location or device (see [3] as an illustration). There are only a few systems that associate the notion of awareness on Groupware Systems to this idea of context-aware computing (for instance [8]). However, this association appears evident to us, once a mobile user is also involved in some cooperative process. And, as any other user in a cooperative environment, a mobile user needs to be aware of what is going on inside the group in order to build a sense of community [18]. This means that Groupware Systems, in order to cater for these mobile users, should provide them with an awareness support adapted to their situation. Consequently, we should consider the activities and the status of the group as parts of the notion of context handled by the system.

In this paper, we explore the hypothesis assuming that the awareness mechanism should exploit the notion of context in order to adapt the information delivered to the user. We propose a context based awareness mechanism which filters the information delivered to the user according a context description. This context description takes into account the concepts related to the notion of awareness (group and role definition, activities and work process, etc.). We use an object-oriented knowledge representation, where these concepts are represented as classes and associations. This paper represents the first part of a work in progress. Such work considers, when defining this context representation, an awareness mechanism embedded on Web-based Groupware Systems, which is accessed through mobile devices. We focus on Groupware Systems that support asynchronous work, such as systems managing group calendar, messages and shared repository (a shared workspace). We assume those systems as composed by many components (such as components for access control policy, for communication facilities, etc.) which are connected and communicate with each other (and eventually with other instances on remote sites). Hence, we assume awareness mechanism as one of such components, and we propose a filtering process that uses the context representation mentioned above to better select the awareness information delivered to mobile users.

This paper is organized as following: first, we present some works related to adaptation for mobile devices (Section 2). Second, we discuss the notion of awareness, our main focus area (Section 3). Then, we present the context description used to represent the extended notion of context (Section 4), and the filtering mechanism based on this description (Section 5), before we conclude (Section 6).

2 Adaptation and Context

The development of software applications for mobile environments involves several technical challenges, which makes adaptation a necessity for the usability of such

systems. Many researches consider this need of adaptation. In their majority, these works deal with the adaptation of multimedia and web-based information content. They usually take into account the technical capabilities of the client device, and try to adapt the content by transforming the original content in such a way that it can be handled by the device (see, for example, [20] and [13]).

Additionally, some works adapt the content by selecting or filtering the content delivered to the user, according to the physical context of the client device. In the latter, the adopted notion of context includes some aspects such as location, time and, of course, the device itself. Some examples of this approach include [3], [16], or [15].

We adopt this approach of selecting the content delivered to the user considering the context where this user finds herself/himself. Moreover, we consider that the organizational context, as well as the physical context, should be taken into account to evaluate what is relevant to a user, and thus, to select the available information for her/him. In fact, according Dourish [5], "context – the organizational and the cultural context, as much as the physical context, plays a critical role in shaping action, and also in providing people with the means to interpret and understand action".

Indeed, this organizational context is important to determine what information is relevant to users. These users are engaged in a cooperative process. Because of this process, such users are particularly interested in information related to their belonging group. More precisely, users, in most circumstances, are interested only in events related to their work context, i.e. events that can lead them to better decisions and/or increase their capacity to decide [4]. In our approach, we represent and explore this context in order to better cater for the user the awareness information to be delivered.

3 Awareness and Adaptation

It is worth noting that the term *awareness* represents a large concept, which can be used in very different situations, as shown by Liechti [14] and Schmidt [21]. The term awareness refers to actors' taking heed of the context of their joint effort, to a person being or becoming aware of something [21]. However, this definition is too vast and we adopt a more concise defined by Dourish [6]: "an understanding of the activities of others, which provides a context for your own activity. This context is used to ensure that individual contributions are relevant to the group's activity as a whole and to evaluate individual actions with respect to the group goals and progress".

There is, in the CSCW community, a consensus about the importance of the awareness support for cooperative work [21][7]. Awareness represents the knowledge about the group, its activities, status and evolution. This knowledge refers to the organizational context where the cooperative work takes place. Thanks to the awareness support, users may coordinate and evaluate their own contributions considering the group evolution. Indeed, awareness support can be used as an implicit coordination mechanism, since whether the members of a team are kept aware of their project status and activities, then they will be able to communicate with each other and coordinate themselves [19].

However, providing an awareness support presents some risks. Espinosa et al. [7], for instance, have obtained encouraging evidences about the benefits of awareness tool use, but they stress the fact that the availability of such tools can turn out to a distraction when not properly used. Awareness tools should fit with the tasks performed by the users.

For users who access Groupware Systems through mobile devices, such as PDAs or cellular phones, the delivered information should also match the constraints of the device as well as the mobile situation of such users. Typically, people using such mobile devices are interested only in information that can help them in their current context. In order to adapt the awareness information to such users, the notion of context should be represented to be exploited for adaptation purposes. In the next section, we present a representation of this.

4 Context Representation

In order to adapt the delivered awareness content to the user's context at a given moment, we have to represent this context in such way that it could be exploited by the awareness mechanism. In order to be useful, we believe that this representation should be limited to relevant aspects of the notion of context. As stated before, we address an awareness mechanism embedded on Groupware System supporting asynchronous work. Thus, the representation of the context described below focus only on concepts relevant to users who access the system through mobile devices.

There are, in the CSCW literature, several propositions of user's context representation. For instance, Leiva-Lobos and Covarrubias [12] propose that the context where cooperating users are situated is tripartite: spatial, temporal and cultural. The spatial context addresses artifacts populating physical or electronic space, while the temporal context refers to the history of performed cooperative processes and to the expected future one. The cultural context gathers users' shared view and practices (i.e., the community practices). Similarly, Allarcón and Fuller [1], define, as principal entities for the work context, the content (tools, shared objects, etc.), the process (activities and their calendar) and the users themselves. In addition, these authors emphasize the importance of the user's electronic location and integrate this concept into the user's context.

Considering these works, we identify five viewpoints containing main entities for a context representation: space, tool, time, community and process. The *space* viewpoint refers to the concept of physical *location*, while the *tool* viewpoint refers to the concepts of physical *device* and *application*. The *time* viewpoint refers to the group *calendar* idea. The *community* viewpoint refers to the composition of the community, including the concepts of *group*, *roles* and *user*. Finally, the *process* viewpoint refers to the *process* (workflow) performed by the group, including the concepts of *activities* (tasks) and *shared objects* (objects handled by the group).

We consider these concepts as the most relevant ones when defining a user's context for a cooperative mobile environment. Using these concepts as a starting point, we define our representation of the notion of context. We represent this notion through an object-oriented representation, using the UML notation. In this representation, the concepts above (user, group, role, location, etc.) become classes (member, group, role…) and the relationships among them, associations.

This representation of context relies, then, on the concepts identified by the five viewpoints above. We consider these concepts as the basic entities of the context description, which we see as a composition of such entities. Thus, our representation starts by the definition of a *context description* class, which is composed of a set of basic elements and is defined for a user that is currently accessing the Groupware System (see the UML class diagram in Fig. 1).

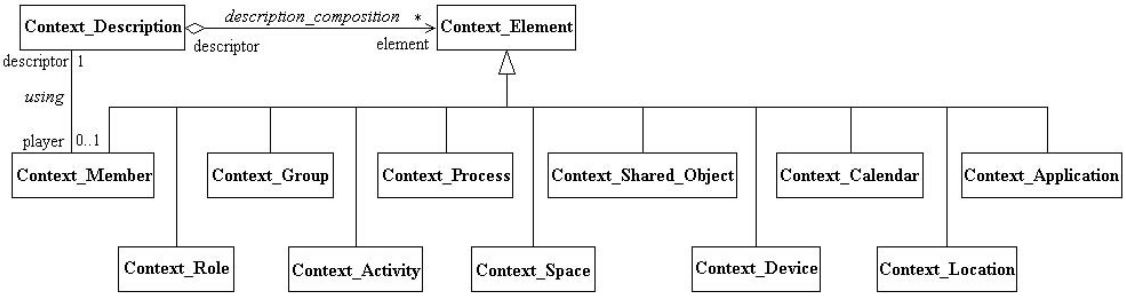


Fig. 1. The context description and the basic elements of the context representation. The prefix "Context" is used in this paper to distinguish the elements of the context representation from those of the awareness mechanism.

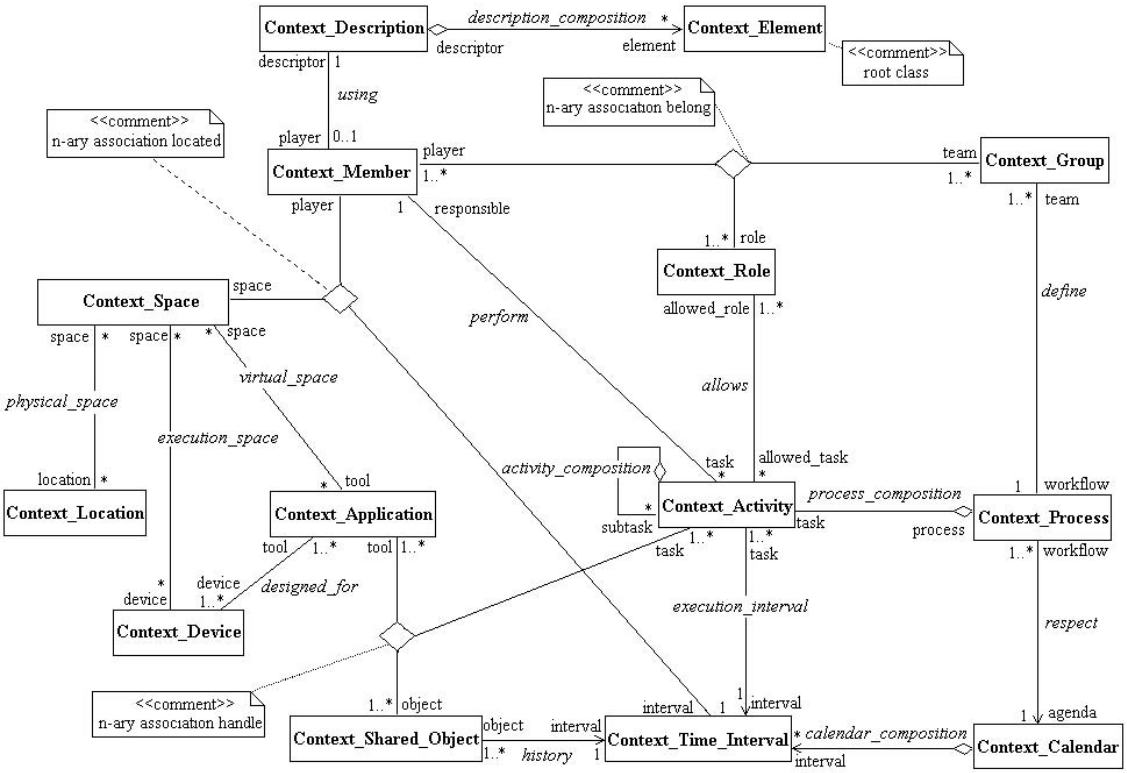


Fig. 2. The context representation with the relationships among all represented concepts.

These basic elements are related to each other, defining relationships between the corresponding concepts, as represented in the Fig. 2. Thus, we define relationships such as the user (called here 'member') that belongs to the group through the roles she/he plays in this group, represented in the model by the ternary association '*belong'* among the classes '*member'*, '*role*', '*group*' (see Fig. 2). We also consider that each group *defines* a *process*, which should *respect* a given *calendar* and is *composed* by a set of *activities* (or tasks, also composed by subtasks). The *roles allow* the execution of an activity, which is effectively *performed* by a member. Each activity *handles* a set of *shared objects* through a set of applications, which are *designed* for specific devices. This *member* is located, during a certain *time interval*, into a certain *space*. This space is composed by a physical space (the member's physical *location*), by a virtual space, comporting the *application* that the member is accessing, and by an

execution space, including the *device* used by the member. The Fig. 2 presents the complete context representation proposed.

These elements, combined through the context description, are able to describe the context of a user that is accessing the Groupware System through a mobile device. These elements, together with the context description, are instances of a knowledge base, which can be exploited by the awareness mechanism. It exploits this context representation through the context description.

As an illustration, let's us introduce a simple scenario of a group member (the team coordinator) who is participating to a business meeting in the company central office. During a break, she/he may access the Groupware System, using her/his PDA, in order to consult the last information about a report that her/his group is writing. For this scenario, the context description will contain the following basic objects: a "member" object describing the mobile user (her/his name, email…); a "role" object, describing the coordination role (rights, etc.); an "activity" object describing the report writing (status, components, deadline…); a "location" object pointing out to the central office; a "device" object defining her/his PDA; an "application" object indicating what application she/he is using; and a "process" object describing the group's main process. Finally, all these objects will compose the context description object associated with this mobile user. The Fig. 3 shows the relationships among these basic objects.

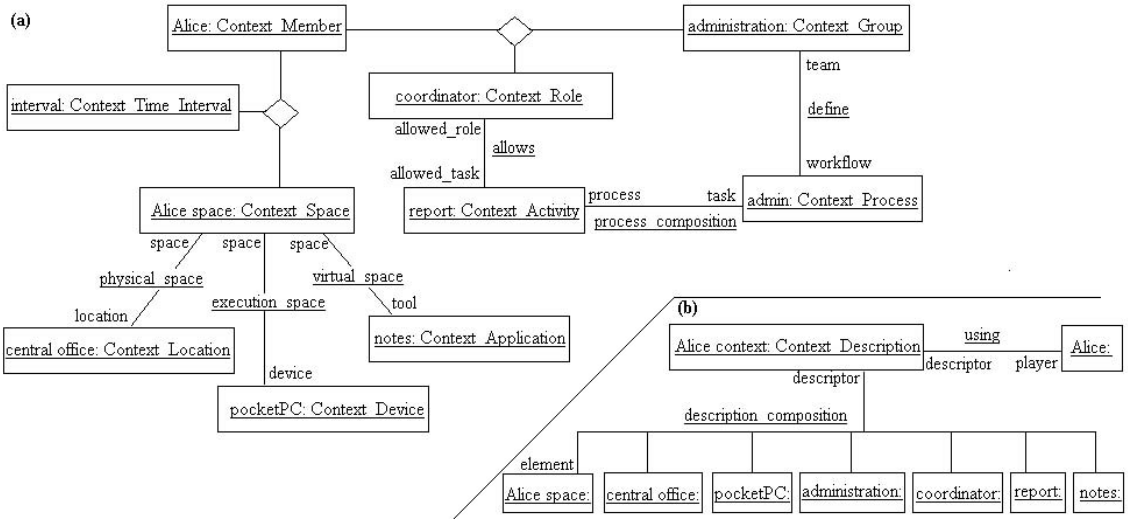


Fig. 3. A scenario of context description. In (a), the relationships among the context elements. In (b), the context description object composed by these context elements.

In addition, since this description is defined as a composition of basic elements, the mechanism can handle partial representations of the context. For instance, a *context description* object may refer, through this composition relationship, only to objects describing the member, her/his roles and the activities she/he is performing, and omit some information about the application or location, or it may include only objects referring to the notion of space (location, device, and application), and ignore all information about the activities and the group process. This omission means that the system does not have enough knowledge to represent these elements, and then it can assume nothing about them. This feature (the omission of some context elements in the context description) is very interesting for awareness mechanism, since often the

system is not able to determine all elements of this context representation. Back to our scenario, if the system cannot determine the user's location and device, context description will be defined without such objects ("central office" and "pocketPC").

We implemented this representation using the AROM system [17] that is an object-based knowledge representation system, which adopts classes/objects and associations/tuples as main entities. This representation allows queries such as if a user is currently using a given device or stands in a specific location. This corresponds to query whether the objects representing such device or location belong to the current user's context description (for instance, if the object "central office" belongs to our mobile user's context description). We now describe how an awareness mechanism can exploit this context representation.

5 Filtering Awareness Information According Context

As we stated before, we consider the awareness mechanism as a component of a web-based Groupware System. This awareness mechanism should be able to analyze the users' activities, as well as those performed by others components, in order to collect information that could be relevant to the performance of the group members. However, the amount of information collected by this mechanism can be very important. Hence, we consider that the awareness information delivered to a group member should be subject of a carefully selection, in order to avoid problems such as an information overload and to better cater for the user's needs.

In this work, we intent to exploit the context representation presented above to perform this selection. We consider an event-based awareness mechanism and we assume that all information that can be delivered to a group member is carried by events. Events are defined by the Groupware developer and each event selects useful information about a specific subject. In order to select the events that should be delivered, the awareness mechanism defines the concept of "*general profile*". This concept represents the preferences and the constraints that the system should apply for a given element (group member, role, device…). For group members, this concept is specialized on "*preferences*", describing the preferences of the user concerning the awareness information delivery, and for devices, it is specialized on "*characteristics*", describing the capabilities of the referred device. These profiles may define what types of events and information should be delivered, as well as its quantity (maximum number of events or Kbytes supported). For devices, the "characteristics" profiles can take the form of a CC/PP description, whereas the "preferences" profiles may indicate a priority order for the events, the time interval that is suitable for the user, and other conditions related to the context description (for instance, if the current device accepts a given media type, or if a given activity has been concluded).

In order to perform the context based filtering process, we associate those events and general profiles with context description objects (see Fig. 4). In fact, these events should be produced in a certain context, which can be represented by a *context description* object. Further, once a user accesses the system, she/he is doing so through a specific context, which is identified by the Groupware System and represented by a context description. We also associate with the profiles, at least, one context description object describing the circumstances where it can be applied. The Fig. 4 presents the associations among events, profiles and the classes defined by the context representation (Section 4).

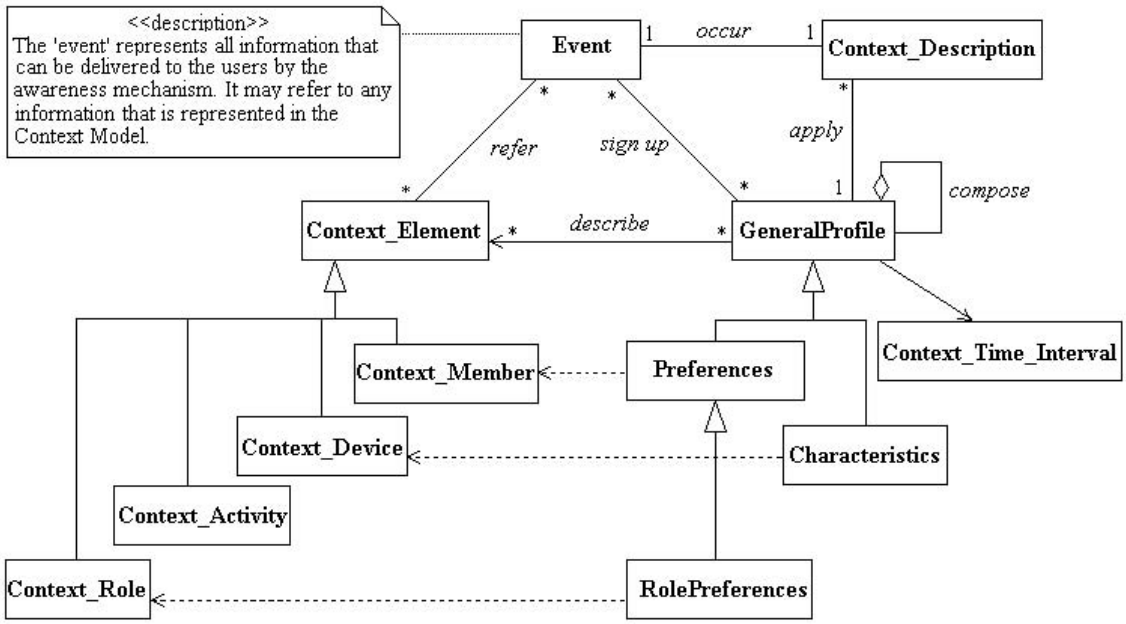


Fig. 4. The application of the context representation on the awareness mechanism.

Then, the proposed filtering process uses these context description objects associated with the group member and with the general profiles to perform the selection of the suitable events for the current context. This filtering process is performed in two steps. First, the awareness mechanism selects the profiles (preferences or characteristics) that are applicable to current user's context description. In fact, we consider that each group member can define for herself/himself a set of profiles and the circumstances, through a context description object, in which each profile can be applied. The selection is performed by comparing the content of the context description objects of both, user and profile: if the context description object of the profile has the same content or is a subset of the current user's context description object, then this profile can be applied. For example; a preference profile for which the context description object is composed by the objects "coordinator" (for the *role*) and "report" (for *activity*) will be selected in the previous scenario (cf. Section 4). In addition, individual elements of the user's context may have its own profiles (e.g. a device which has its own "*characteristics*" profile).

Once all applicable profiles are selected, awareness mechanism can apply them over the available set of events, performing the filtering process. We suggest a gradual application, first applying the selected preferences of the user and then applying the other selected profiles. This can be made respecting a selection order (one profile at a time) or performing first a merge of the content of each selected profiles. This latter is especially interesting for "preferences" objects, which union can form a complete set of the group member's preferences. It is worth noting that, in this case, the awareness mechanism should be able to handle eventual incompatibilities among the selected profiles

The result of such filtering process should be a limited set of events that will be delivered by the Groupware System. However, our approach presents some limitations. The first one relies on the profiles definition that may be a hard and boring task. In addition, the definition of profiles with strict sets of application circumstances (profiles with context description composed by many context objects) may lead to the

selection of no profile at all. If the context description object associated with the profile is larger than the one related to the user, it will never be a subset of the latter and such profile will never be applied. This will block the filtering process, leading to the presentation of all the available events.

Other limitation of our approach concerns, similarly to other context-aware systems, the context detection as all filtering process depends on the user's context description object. Hence, the definition of this object by the system (the context detection process) is very critical.

6 Conclusion

This paper presents a proposition of context base filtering process for awareness mechanisms, using a context representation that uses an object-based knowledge model. This context representation represents the main contribution of this paper. It was successfully implemented on a knowledge base, using the AROM system, and we are now implementing the filtering process, using a framework for awareness support called BW [10]. We expect to perform practical tests using a test application, a cooperative game, specially designed for this purpose. In this game, small teams will use PDAs and laptops to look for (and describe) targets (sights and objects) geographically distributed. Users will then use the system mainly to communicate and build the descriptions of the targets, and the awareness mechanism to coordinate their actions. Through these tests, we expect to evaluate the effective impact of our proposition in a context based awareness mechanism, and the user's acceptation, specially concerning the limitations related to the profiles (cf. Section 5).

Acknowledgements

This work received grants from CAPES-Brazil (BEX 2296/02-0). Authors would like to thanks the referees by their valorous comments.

References

1. Alarcón, R., Fuller, D.: Intelligent awareness in support of collaborative virtual work groups. In Haake, J.M., Pino, J.A. (eds.), CRIWG 2002, LNCS 2440. Springer-Verlag (2002) 168-188
2. Brézillon, P.: Modeling and using context: past, present and future. Rapport de Recherche LIP6 2002/010 (2002) http://www.lip6.fr/reports/lip6.2002.010.html (Jan., 2004, in French)
3. Burrell, J., Gray, G.K., Kubo, K., Farina, N.: Context-aware computing: a text case. In: Borriello, G., Holmquist, L.E. (eds.), Ubicomp 2002, LNCS 2498. Springer-Verlag (2002) 1-15
4. David, J.M.N., Borges, M.R.S.: Improving the selectivity of awareness information in groupware applications. In: International Conference on Computer Supported Cooperative Work in Design (CSCWD'2001). IEEE Computer Society (2001) 41-46
5. Dourish, P.: Seeking a foundation for context-aware computing. Human Computer Interaction, vol. 13, n° 2-4. Hillsdale (2001) 229-241

6. Dourish, P., Bellotti, V.: Awareness and Coordination in Shared Workspaces. Proceedings of ACM Conference on Computer-Supported Cooperative Work (CSCW'92). ACM Press (1992) 107-114
7. Espinosa, A., Cadiz, J., Rico-Gutierrez, L., Kraut, R., Scherlis, W., Lautenbacher, G.: Coming to the wrong decision quickly: why awareness tools must be matched with appropriate tasks. CHI Letters, CHI 2000, vol. 2, n° 1. ACM Press (2000) 392-399
8. Hibino, S., Mockus, A.: handiMessenger: awareness-enhanced universal communication for mobile users. In: Paternò, F. (ed.), Mobile HCI 2002, LNCS 2411. Springer-Verlag (2002) 170-183
9. Jing, J., Helal, A., Elmagarmid, A.: Client-server computing in mobile environment. ACM Computing Surveys, vol. 31, n° 2. ACM Press (1999) 117-157
10. Kirsch-Pinheiro, M., de Lima, J.V., Borges, M.R.S.: A Framework for Awareness support in Groupware Systems. Computers in Industry, vol. 52, n° 3, Elsevier (2003) 47-57
11. Kirsh, D.: The context of work. Human Computer Interaction, vol. 13, n° 2-4. Hillsdale (2001) 305-322
12. Leiva-Lobos, E.P., Covarrubias, E.: The 3-ontology: a framework to place cooperative awareness. In Haake, J.M., Pino, J.A. (eds.), CRIWG 2002, LNCS 2440. Springer-Verlag, (2002) 189-199
13. Lemlouma, T.; Layaïda, N.: Context-aware adaptation for mobile devices. In: IEEE International Conference on Mobile Data Management (2004)
14. Liechti, O.: Awareness and the WWW: an overview. In: ACM Conference on Computer-Supported Cooperative Work (CSCW'00) - Workshop on Awareness and the WWW. ACM Press (2000) http://www2.mic.atr.co.jp/dept2/awareness/fProgram.html (Dec., 2002)
15. Muñoz, M.A., Gonzalez, V.M., Rodriguez, M., and Favela, J.: Supporting context-aware collaboration in a hospital: an ethnographic informed design. In: Favela, J., Decouchant, D. (eds.), CRIWG 2003, LNCS 2806. Springer-Verlag (2003) 330-344
16. O'Hare, G., O'Grady, M.: Addressing mobile HCI needs through agents. In: Paternò, F. (ed.), Mobile HCI 2002, LNCS 2411. Springer-Verlag. (2002) 311-314
17. Page, M., Gensel, J., Capponi, C., Bruley, C., Genoud, P., Ziébelin, D., Bardou, D., Dupierris, V.: A New Approach in Object-Based Knowledge Representation: the AROM System. In: 14th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, (IEA/AIE2001), LNAI 2070, Springer-Verlag (2001) 113-118
18. Perry, M., O'Hara, K., Sellen, A., Brown, B., Harper, R.: Dealing with mobility: understanding access anytime, anywhere. ACM Transactions on Computer-Human Interaction, vol. 8, n.4. ACM Press
19. Projet ECOO: Projet ECOO: environnements et coopération. Rapport d'activité, INRIA (2001) http://www.inria.fr/rapportsactivite/RA2001/ecoo/ecoo.pdf (Nov. 2002, in French)
20. Schilit, B.N., Trevor, J., Hilbert, D.M. and Koh, T.K.: Web interaction using very small Internet devices. Computer, vol. 35, n°10. IEEE Computer Society (2002) 37-45
21. Schmidt, K.: The problem with 'awareness': introductory remarks on 'Awareness in CSCW'. Computer Supported Cooperative Work, vol. 11, n° 3-4. Kluwer Academic Publishers (2002)

Annex III

Paper

Kirsch-Pinheiro, M.; Vanrompay, Y.; Victor, K.; Berbers, Y.; Valla, M.; Frà, C.; Mamelli, A.; Barone, P.; Hu, X.; Devlic, A.; Panagiotou, G., “Context Grouping Mechanism for Context Distribution in Ubiquitous Environments”, In: Robert Meersman, Zahir Tari et al.(eds.), *10th International Symposium on Distributed Objects, Middleware, and Applications (DOA'08), OTM 2008 Conferences, Lecture Notes in Computer Science*, 5331, 2008, 571-588.

Context Grouping Mechanism for Context Distribution in Ubiquitous Environments

M. Kirsch-Pinheiro[1], Y. Vanrompay[1], K.Victor[1], Y. Berbers[1], M. Valla[2], C. Frà[2], A. Mamelli[3], P. Barone[3], X. Hu[4], A. Devlic[5,6], and G. Panagiotou[5,6]

[1] Katholieke Universiteit Leuven, Leuven, Belgium
[2] Telecom Italia Lab, Milan, Italy
[3] Hewlett-Packard Italiana - Italy Innovation Center, Milan, Italy
[4] European Media Laboratory, Heidelberg, Germany
[5] Appear Networks, Kista, Sweden
[6] Royal Institute of Technology (KTH), Stockholm, Sweden
{manuele.kirschPinheiro,yves.vanrompay,koen.victor, yolande.berbers}@cs.kuleuven,be, {massimo.valla,cristina.fra}@telecomitalia.it, {alessandro.mamelli,paolo.barone}@hp.com, xiaoming.hu@eml-d.villa-bosch.de, ade@appearnetworks.com, gpan@kth.se

Abstract. Context distribution is a key aspect for successful applications within mobile and ubiquitous computing environments. In such environments, context information is acquired by several and multiple context sensors distributed over the environment. Applications collect and react to these data, according to predefined adaptation mechanisms. The success of these mechanisms depends on the availability of context information, which is disseminated over the network. However, in practice, only a fraction of the observable context information is required by the adaptation mechanisms. Moreover, for privacy reasons, it is important to delimitate a scope for context dissemination. In this work we address these issues by proposing a context grouping mechanism which allows the definition of groups based on the context characteristics. Each group is defined by these characteristics and delimitate a given context information set that can be distributed among group members. This approach of context grouping acts as a two-fold mechanism. On the one hand, it controls and organizes context distribution over a peer-to-peer network. On the other hand, it proposes a primary and low-level privacy mechanism for context distribution, which is an important aspect influencing context distribution.

Keywords: Context awareness, context distribution, peer-to-peer systems, context grouping, JXTA.

1 Introduction

Context distribution is a key aspect for successful applications within mobile and ubiquitous computing environments. In such environments, context information is acquired by

R. Meersman and Z. Tari (Eds.): OTM 2008, Part I, LNCS 5331, pp. 571–588, 2008.

several and multiple context sensors distributed over the environment. Context-aware applications [1] collect and react to this information, exploiting it through predefined adaptation mechanisms. The success of such mechanisms depends on the availability of context information, which is disseminated over the network. However, in practice, only a fraction of the observable context information is required by the adaptation mechanisms. For instance, in a metro station[1], a wide variety of information can be available: temperature and humidity, available computing infrastructure, network status, etc. Each context-aware application running in such an environment will use only a subset of this information: a traveling guide can use, for adaptation purposes, available computing infrastructure, network status and user profile information, and ignore temperature and humidity information, which is exploited by maintenance applications. In such ubiquitous environments, a specific context distribution mechanism is necessary in order to control and organize this information distribution appropriately.

Moreover, for privacy reasons, it is important to define a scope for context dissemination. Context information can be sensitive information, whose unlimited distribution can be perceived as inappropriate by the users. For instance, in the scenario presented above, information related to the location of technical workers in the metro can be considered as sensitive information. This kind of private context information can be exploited by context-aware applications, but its distribution should thus be limited to the allowed entities.

In this work we address these issues by proposing a context grouping mechanism which allows the definition of groups based on context characteristics. Each group is defined by these characteristics and delimitates a given context information set that can be distributed among group members. This approach of context grouping acts as a two-fold mechanism. On the one hand, it controls and organizes context distribution over a peer-to-peer network. On the other hand, it proposes a primary and low-level privacy mechanism for context distribution. We consider here highly dynamic environments, in which entities may appear and disappear at any moment, and in which context-aware and self-adapting applications exploit information available in the environment for adapting the supplied content and services, as well as their internal structure. Moreover, this grouping mechanism belongs to a larger initiative, the MUSIC Project, which focuses on the development of context-aware self-adapting applications [12]. Applications run on the top of a middleware, like the MUSIC middleware, executing in ubiquitous environments. In this paper, we propose a context grouping mechanism that addresses such dynamic environments.

The rest of this paper is organized as follows: In Section 2, we review related work on context distribution and P2P grouping. Section 3 introduces a hybrid P2P architecture we adopt in this paper. Section 4 presents our proposal of the context-aware grouping mechanism. In Section 5, we discuss how this mechanism can be used as a low level privacy mechanism. In Section 6 we present some experimental results before concluding in Section 7.

2 Related Work

Context distribution is a key aspect for every context-aware application. As such, it is handled by middleware systems on top of which these applications are built. In the

[1] This example is inspired by the MUSIC Project [12] scenarios.

literature, centralized solutions in which central server concentrates and manages context information collected from the environment remain numerous [1]. Examples of such centralized architecture for context distribution include Henricksen *et al.* [6] and Paganelli *et al.* [13], which propose centralized entities (context repositories for the former and domain server for the latter), managing the context information and handling the client requests for it. These centralized approaches are prone to scalability and single point of failure requests.

A different approach is required in a distributed architecture in which the generation and the management of context information are distributed over the network. This is the case of Ye *et al*. [17], who adopted a P2P approach for context sharing, in which context information remains locally stored on the peers and only an access reference is registered on remote peers. The discovery of new peers is performed by broadcasting messages and context queries to remote peers are sent through unicast messages. The broadcasting of registration messages and the flat structure may raise scalability and security issues on ubiquitous environments, since every peer potentially registers references to available context information on every known peer.

The P2P approach is particular interesting for mobile and ubiquitous computing environments. It supports high dynamic environments, with no need of centralized elements, providing a more resilient solution considering scalability and nodes failure [4][8]. The distributed aspects of the P2P paradigm motivate the proposal of architectures such as [5], which implements a topic-based publish-subscribe system with SIP/SIMPLE [16]. This proposal uses P2P SIP [2] for the structure and maintenance of a structured P2P overlay.

Considering traditional P2P architectures, several works in the literature [11] [9] [3] propose grouping mechanism. Cutting *et al.* [3] propose implicit group definition based on tags used by users to identify interesting content. Peers interested on the same content tags are considered to belong to the same implicit group. Messages are then addressed to tags forming the groups. Similarly Khambatti *et al.* [11] propose a grouping technique based on the notion of community. Communities are groups formed based on users (or peers) common interests, declared as peer attributes. Thus, communities are automatically defined over common *claimed attributes*, which correspond to published attributes from the set of all observable/available attributes:

- a *community* C is a non-empty set N of peers sharing a common signature
- a *signature* S is the intersection set of claimed attributes claim(k) for all $k \in N$.

Our group membership is inspired by these dynamic group definition proposals. In our case, the context groups can be seen as communities defined using common context information. However, instead of automatically discovered/defined communities, in our case the application should be able to determine claimed sets in order to define different context groups and to associate with each group a different allowed information set for diffusion purposes, according to the application needs. The explicit association of a particular context type with the group definition distinguishes our proposal from traditional P2P grouping mechanism, such as [11] [9], which maintain this association implicitly. For instance, in [11], applications are supposed to send to the group only information related to the group signature. Moreover, the context grouping mechanism we propose in this paper is particularly designed considering the distribution of context information over ubiquitous environments. In addition of

handling the high dynamic character of context information, context distribution over such environments should match requirements such as heterogeneity, scalability and robustness [8], which we are aiming at.

3 A Peer-to-Peer Context Distribution Architecture

In previous research, the MUSIC project proposed a hybrid peer-to-peer architecture for context distribution [8], which categorized peers into 3 kinds, mainly according to device capabilities and associated roles for context distribution purposes: (*i*) *sensor peers,* which correspond to devices providing only raw context data (typically, context sensors); (ii) *consumer peers*, resource-constrained devices that need polished (and synthesized) context information; (iii) *disseminator peers*, which represent peers providing both provider and consumer functionalities, they provide context distribution services and perform the context reasoning.

In this paper, we are particular interested in disseminator peers, which take full responsibility for context distribution in this architecture. Each peer offers a "*Context Service*", which provides context clients with a way to query context or to subscribe to context updates, and a "*Distribution Service*", which disseminates context information in a certain scope (see Fig. 1) [8]. In this paper, we go further by improving these services proposing a more advanced context grouping mechanism. This new grouping mechanism differs from the previously proposed one [8] mainly by the use of context information as criteria for group formation and by the explicit attribution of an allowed context-related content to each group. We start by considering these services as forming two independent layers: a "reasoning" layer, which manages the context grouping (together with other context reasoning functionalities, which is out of the scope of this paper), and a "distribution" layer, which handles the context dissemination according to the grouping definitions. In this way, the grouping mechanism can be defined independently of the actually used distribution technologies (cf. Section 4.3).

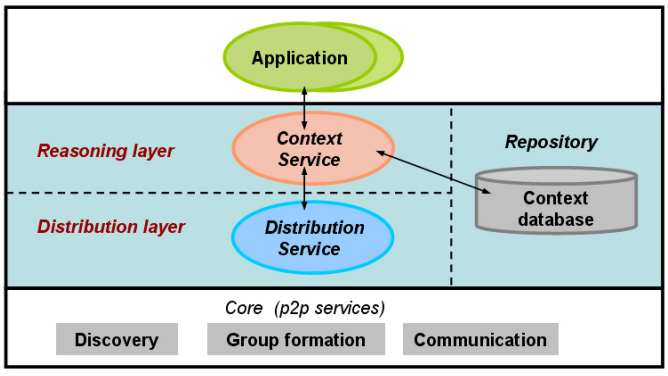


Fig. 1. Architecture of context disseminator peers

In the next sections, we present a context-aware grouping mechanism based on this architecture and describe how the grouping is performed by these layers.

4 Context-Aware Grouping Mechanism

Based on the architecture presented in Section 3, we propose to organize context dissemination in groups based on context information. The main principle consists in forming groups with peers that share common observable context items (for example, the same location, the same network connection, etc.). In this sense, grouping can be seen as the peer neighborhood. The notion of neighborhood is here extended to include not only the notion of network neighborhood (peers in the same network), but also geographical neighborhood (peers in the same location) or other application-defined criteria (peers executing over similar devices, peers acting on behalf of users playing a given role in an organization, etc.).

Thus, in the "reasoning" layer, applications can determine the criteria for the group formation. We consider that these criteria for forming context groups depend on the context characteristics, which can be described by a context model, in our case the MUSIC Context Model (see Section 4.1). The MUSIC context model delimits what context elements can be potentially used as grouping criteria: all context elements defined in the context model can be used and combined by the application to form groups (for example, groups based on the location, on the network connection, etc.). Once the criteria are defined, the groups are formed (see Section 4.2) and used as "context addresses" by the "distribution" layer (see Section 4.3).

The Fig. 2 illustrates the grouping definition we propose. This figure shows a set of peers forming two groups: (*i*) a first group is formed based on the peer location (group formation criterion) and disseminating information about device available memory, screen size and battery; (*ii*) and a second group is defined based on the user role and disseminating location information. The criteria defining the groups are translated to real values of peer observable context (location = room x, user role = expert, …), which are used by the distribution layer to disseminate the allowed context information (device resource information for the first group, user location for the second one) and queries about it. In this sense, this context grouping mechanism can be compared to a decentralized publish-subscribe mechanism since applications can decide to join only groups disseminating context information in which the application is interested on.

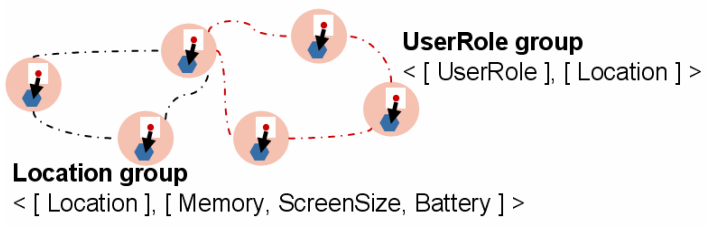


Fig. 2. Example of the context group definition including criteria and allowed content

4.1 Context Modeling: The MUSIC Context Model

In the MUSIC project [12], the term "context" denotes the circumstances and conditions under which software services and systems are being used. The approach chosen in MUSIC for context modeling is detailed in [14]. It identifies three basic layers of abstraction that correspond to the three main phases of context management: the conceptual layer, the exchange layer and the functional layer. The conceptual layer enables the representation of context information in terms of *context elements*, which provide context information about *context entities* (the concrete subjects the context data refers to: a user, a device, etc.) belonging to specific *context scopes*. The conceptual layer incorporates also the use of ontologies (Fig. 3) that are described in OWL; the context scopes are intended as semantic concepts belonging to them. All context scopes are associated with one or more *Representations*, which are also specified in the ontology, that describe the context data. Moreover the ontology is used to describe relationships between entities, e.g. a user has a brother or a device belongs to a user. Anyway, since ontology reasoning on mobile devices with limited resources can be onerous, we allow to reference context entities and context scopes also through predefined types. The type implicitly corresponds to a certain semantic concept and to a default representation of the context information.

In previous work the MUSIC project defined also a Context Query Language detailed in [15] for querying context data, based on XML and strongly related to the context model previously described. This language allows applications to submit to the context system queries about an entity and referred to a context scope, specifying a set of constraints that represent the filters of the query. The most simple query has no constraints and refers to a context scope represented as a predefined type. Queries can also be more complex, using one or more constraints connected with logical operators (potentially with unlimited nesting) and containing semantic references. In this way queries about relationships between entities are allowed.

These features enable criteria for context group definition. Using CQL, applications can submit to the context system simple or complex queries in order to retrieve data needed for group creation. Moreover, using queries with semantic references,

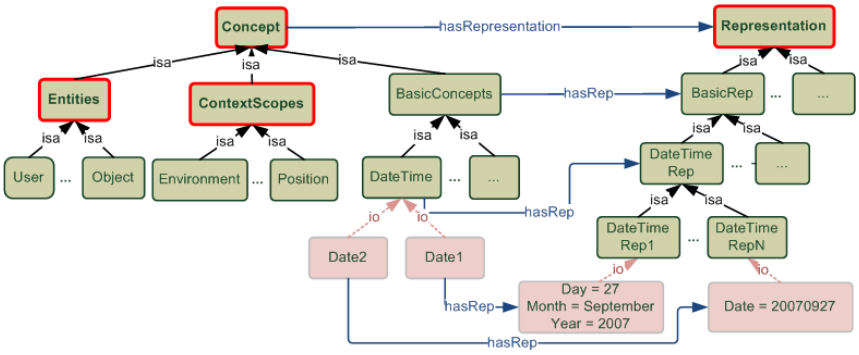


Fig. 3. The MUSIC context ontology

relationships between entities can be used as new criteria for the group definition. For example an application can require through a complex query with semantic references, the list of users located in a specific city and belonging to the same family. Such queries can be used for context group definition.

4.2 Context Service: A Grouping Reasoning Layer

The context service acts as a grouping reasoning layer, which is responsible for forming and managing peer groups. It handles the application requests for context information and for the grouping formation. We consider two types of context requests: pull requests in which peers query context information from other peers in the neighborhood, and push requests in which peers proactively send context information to other peers in the neighborhood.

For group management, we consider that the application determines the criteria for forming a group and the context information which is allowed to be distributed in this group. Only information related to the group is disseminated to the group members. The application indicates what context elements determine a group (location, network, user role, etc.), and the context service translates the criteria based on the current context values (location=room x, network=xxx.xxx.xxx.xxx, user role=expert, etc.). Thus, we may consider context groups are defined as follows:

$$G_D = < C_D, I_D >, \textit{ where}$$

- C_D is the *criteria set*, *i.e.* set of context elements that determine the context group
- I_D is the *dissemination set* (also called *working set*). It is the context information that can be disseminated on this group, *i.e.* the context elements that are allowed to be disseminated (by push requests) or requested (by pull requests) in this group.

Thus, an application defines a group "template" in which it stipulates, using queries in the CQL, the criteria and dissemination sets that define the group. For instance, Fig. 4 illustrates two CQL queries defining, respectively, the criteria set (Fig. 4a) and the dissemination set (Fig. 4b), which act as a template for a context group. This template states for a context group based on the user's location information (location equals to user's current room) and which allows the dissemination of context information related to the user's device memory status (Fig. 4b). It is worth noting that, for readability purposes, we express this group "template" using literal representation as illustrated by Fig. 4c.

For the application, the group definition remains stable over time, since its template remains the same. However, since this definition relies on context information, it is naturally dynamic. The values corresponding to the group criteria and the dissemination set are updated according to the context changes by the reasoning layer. The context group "template" is "instantiated": the query representing the group criteria is processed and the current values of the corresponding context elements are used to form the context group properly speaking. For example, considering a group G_1 defined as <[location], [memory]> in Fig. 4, its definition criterion is translated by the context service to [location=room x], corresponding to the current value of this context element. Once the value of the context elements used as criteria changes (*e.g.*, the

user is moving around), the context service updates the group definition to the new value ([location=room y]), and informs the distribution service about this change. The distribution service acts accordingly, leaving and joining (or creating) the corresponding group at the network level. In this way, changes related to the context groups are hidden from the applications.

It is worth noting that the grouping mechanism depends on the query determining the grouping criteria. If this query is not well-formed or it is ambiguous, the context service is unable to interpret it and to form or join the context groups. For instance, considering a location-based context group, an application can consider that two users are in the same location if they are in the same room (such as in the query on Fig. 4a), or in the same city, or in the same coordinate set. Even if the CQL defined in the MUSIC project [15] allows a query just indicating the context concept "location", this is not enough to indicate without ambiguity what the application considers as being in the same location. Applications using the proposed context grouping mechanism should avoid queries with any ambiguity when defining context grouping criteria.

Based on the group definitions, the context service organizes the context dissemination and requests, by handling accordingly the *context query messages* and *dissemination messages* coming from other peers over the network, as well as *context request messages* coming from the application.

4.2.1 Forming Context-Aware Groups

The formation of groups in the reasoning layer is the process of specifying G_D for use in context information requests. Ideally, this functionality may be used explicitly or implicitly. Applications that are aware of context grouping use the methods exported by the reasoning layer to manage for the group information explicitly. For privacy issues that are specified on the application level (as opposed to the middleware or

(a)

```
<contextQL>
  <ctxQuery>
    <action type="SELECT"/>
    <entity ontConcept="prefix:music:username">#user</entity>
    <scope ontConcept="prefix:music:Location"
           ontRep="prefix:music:DefaultLocationRep">#location</scope>
    <conds>
        <cond type="ONVALUE">
                <constraint param="#location.#room" op="EQ"/>
        </cond>
    </conds>
  </ctxQuery>
</contextQL>
```

(b)

```
<contextQL>
  <ctxQuery>
    <action type="SELECT"/>
    <entity ontConcept="prefix:music:devicename">#any</entity>
    <scope ontConcept="prefix:music:Resource"
           ontRep="prefix:music:DefaultMemoryRep">#memory</scope>
  </ctxQuery>
</contextQL>
```

(c)

G_d = < [position], [memory] >

Fig. 4. Example of CQL queries defining a context grouping criteria and working sets

```
<groupProfile GroupId="gamingGroup" xmlns="http://GroupProfile/1.0"
                                    xmlns:cql="http://ContextQL/1.0">
   <grpCriteria>
        <cql:contextQL>
            <cql:ctxQuery>
                <cql:action type="SELECT"/>
                <cql:entity>#user</cql:entity>
                <cql:conds>
                    <cql:cond type="ONVALUE">
                        <cql:logical op="AND">
                            <cql:constraint param="#location.#room" op="EQ"/>
                            <cql:constraint param="#user.#age" op="LT" value="30"/>
                            <cql:constraint param="#device.#freememory" op="GT" value="40"/>
                        </cql:logical>
                    </cql:cond>
                </cql:conds>
            </cql:ctxQuery>
        </cql:contextQL>
   </grpCriteria>

   <grpWorkingSet>
        <cql:contextQL>
            <cql:ctxQuery>
                <cql:action type="SELECT"/>
                <cql:entity>#device</entity>
                <cql:scope>#memory</scope>
            </cql:ctxQuery>
            <cql:ctxQuery>
                <cql:action type="SELECT"/>
                <cql:entity>#device</entity>
                <cql:scope>#battery</scope>
            </cql:ctxQuery>
        </cql:contextQL>
   </grpWorkingSet>
</groupProfile>
```

Fig. 5. Example of a group profile

network level), the applications should use these methods to associate a group specification to the requests. When an application uses the context grouping implicitly, it is not aware of the group specifications and does not need to use these methods. The grouping mechanism may then be used 'silently' in the distribution layer, to optimize the availability of context information or to locate the parts of the network to which the requests should be sent. In this paper, we are focusing only on explicit context grouping, leaving implicit grouping as a future direction.

Explicit group formation is used when the application itself specifies the criteria for grouping. Depending on the requirements of applications, the context service provides the group management mechanisms to create, update and leave context groups. Each context group is associated with a group specification, called the *Group Profile* (*i.e.* the group "template"). The Group Profile reflects the above definition of the context group by containing C_D, the common context that determines the group scope and I_D, the offered context in the group which decides the function of the group. Also, the policies are included in the group profile to determine how to manage the group membership. Each group is established with its own membership policy including open (anyone can join) to highly protected (require credentials to gain membership) since some groups are built with user private data. Besides, some groups are formed depending on the presence of one or multiple members, on the availability of certain resources within the federation, at specific times or in a special location. Thus, these criteria related to formation and termination of the groups can be also stored in the group profile to decide when the group can be formed, updated, even terminated.

For example, an ad hoc gaming group defined by a group profile could be G = < [User.current location = same room, User.age = Member.age <30; condition= User.freememory > 40M], [Member.DeviceBattery, Member.DeviceMemory] >. Fig. 5 illustrates this group profile in XML.

When applications or other components in the system request a certain context, the context service will firstly reason the request and search for the corresponding context group to forward the request. If such a group does not exist, the context service will use a predefined or a (semi-)automatically generated group profile to initiate the group (*e.g.* the group creator could decide the member's age on the spot, or the current location of a user is translated into real values in the group profile). After the group is formed, the group profile will be published to other peers in order to be found by the potential members for joining the group.

Each application is able to define multiple groups, according its particular needs or policies for context distribution and privacy. Once groups are defined, the application performs its request for context information in a transparent way: it does not need to specify to which group of peers a request is addressed. The distribution service decides it automatically according to each group criteria and dissemination sets. This way, the context group may be used as an address targeting all peers in the network.

The application interacts with the context service through an API, which supplies methods such as *discoverGroup*, *createGroup, joinGroup, leaveGroup, optimizeGroup* or *updateGroup* for the management of groups through applications. The optimization of context groups is not discussed in this paper. The discovery of groups is described in section 4.2.2. Updating of groups is explained in section 4.2.3.

4.2.2 Discovering and Joining Context Groups

In the previous section, we discussed how an application, running on top of one peer, defines its context groups using the context service located in that peer. In this section, we focus on how the context service can discover and join similar context groups defined by other peers in the network.

The application defines and submits to the context service the desired group profile to declare which groups it is willing to join. By using this profile, each peer can either actively search for a group, or subscribe to the group discovery function and get notified when its desired group is created by other peers. Fig. 6 shows several interactions taking place during the group joining process of two peers. In this example PeerA is the group creator who uses a group profile to create a context group (1). In order to receive shared context information, PeerB has registered itself as a group discovery listener using its desired group profile (2). The desired group profile describes the context types which PeerB is interested in. In other words, only context groups which can provide the required context information will be discovered by PeerB. When the context group is made available by PeerA in the network, its group profile is broadcasted (3). If this group is able to provide the context information required by PeerB, it is discovered by PeerB by using its registered group listener (4). Before applying for the group membership, the context service of PeerB evaluates the group membership policy against its individual profile (*e.g.* if it allows sharing the personal information such as age or device resources) and the current context information (*e.g.* if PeerB is currently in the same location) (5). If certain information is not defined in the profile, the user will get an alert to ask for the permission since the user should still be

able to decide the profile sharing on the spot (*e.g.* an alert to ask if the user wants to share the age). If group policies match its personal profile, the context service of PeerB will enable to join the group (6).

Before a peer can interact with the group, a process is required to allow the peer to establish its identity within the peer group. Each context group has a membership policy that governs who can join the group. The potential members have to interact with the policy by the exchange of an authenticated identifier, credentials or any other mechanism that a context group membership implementation requires. In Fig. 6, PeerB applies for the membership by sending the required information enclosed in an individual profile to PeerA (7). After authentication by PeerA (8), PeerB is accepted by the context group. PeerA sends the group related information (*e.g.* group id, group member list) to PeerB, who can use the information to initiate the context group and interact with other group members (9). It is worth noting that the group creator role is not fixed: any peer can assume this role and take the initiative of creating a group.

4.2.3 Updating Context Groups

Updating context groups, which means creation, change or termination of context groups, is necessary to keep the context groups aligned to the behavior of the applications, changes in the observable context, privacy requirements or for optimization of groups. When and how to update context groups is dependent on the group policy described by the group profile. For example the peer only needs to switch the context group when the person changes to another floor or building; when the member stops to share his personal data, he has to leave the group and the context group needs to be terminated when there are less than two group members.

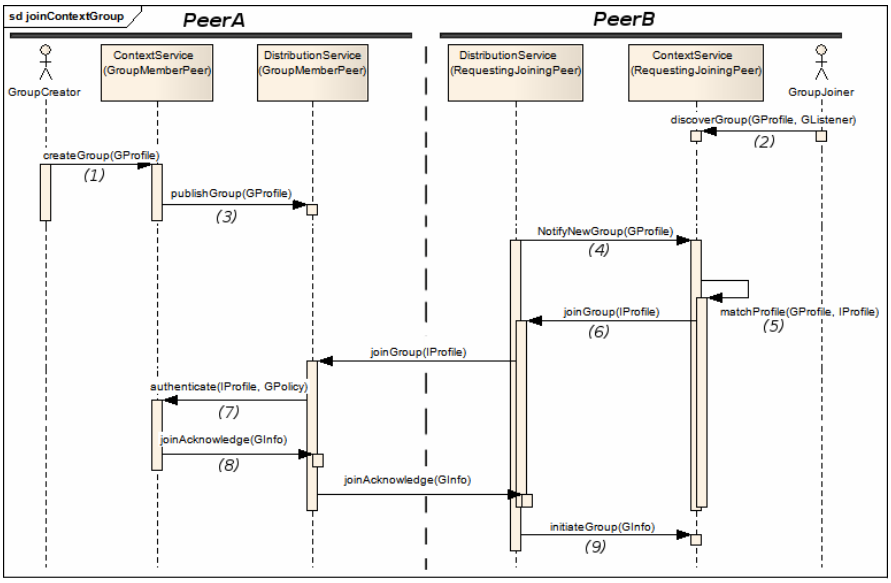


Fig. 6. Sequence diagram of joining a context group

During the execution of an application, the reasoning layer may decide that the previously created groups do not fulfill the needs of the application anymore. For example, the person using the application may be in a different location and the location based group does not refer to it. At this point, the context service may decide to search for or create another group, based on the new location of the user. Then, the peer leaves the current group and joins the new group. New requests coming from the application are associated to the newly created group. Conceptually, the old group is irrelevant at this point and should be removed by the distribution layer. However, for performance reasons, the old group is not removed instantly, and the context service puts the 'old' group in a cache, to remove it later. Since the network will typically be dynamic, changing groups in the distribution layer may be inefficient and cause a lot of redundant traffic, especially because applications and users may follow patterns (for example, they can move between the same rooms). In our scheme, irrelevant groups that become relevant again, can be reused instantly.

When a group is not used for a long period (expressed in terms of average number of requests per group), the group should be terminated. The *leaveGroup* method is called and the distribution layer will remove the group from the overlay network.

4.3 Distribution Service: Group-Based Context Dissemination

The distribution service corresponds to the context dissemination layer of our architecture. It interprets the groups defined by the context service and controls the distribution of the context information over the network according to these definitions. The grouping criteria, translated into the current context values, are used to form the "group address" referring to a specific group.

The Distribution service handles the context group formation at the network level. As shown in Fig. 1, it acts as a mediator between the context service and the mechanisms offered by the P2P core layer. This means to translate the context groups to traditional P2P groups that can be understood by the technologies used on the P2P core level of the architecture. Ubiquitous environments are quite heterogeneous, and consequently, the distribution service can be potentially confronted to different technologies. That is the reason motivating the separation between reasoning and distribution layers in the proposed architecture. On the one hand, the context grouping mechanism can be defined independently of the technology used at the network level. On the other hand, the distribution service decouples the operations performed on the groups from the P2P technology used to implement these operations.

In the proposed architecture, the context service interacts with the distribution service through a well defined API. This interface defines the functionalities supplied by the distribution layer, hiding technology details to the context service. Fig. 7 presents some elements of the proposed API. Through this API, it is possible to consider different implementations for the distribution service, without compromising the context service implementation. In Section 6, we discuss an evaluation implementation scenario considering JXTA [10] technology.

Besides, the distribution service handles context requests accordingly to the group definitions. For instance, considering the gaming application mentioned in Section 4.2.1, when such application queries (pull request) context information regarding available memory, this request is redirected to the group defined by the profile in

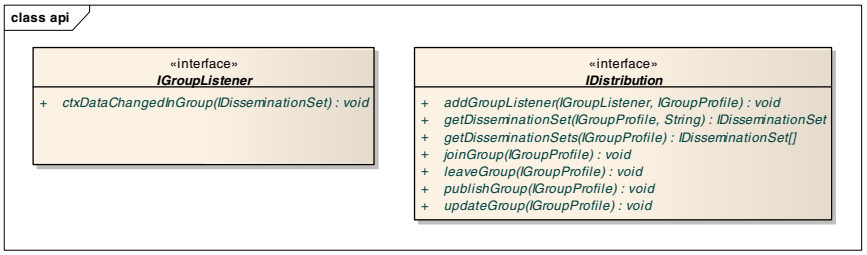


Fig. 7. Elements of the distribution service API

Fig. 5. The context service requests this information from the distribution service using the *getDisseminationSet* method in the API. The distribution service transfers the corresponding context query to the other group members of this group, according to the underlying technology. Pull requests are performed in a similar way. When the distribution service receives a message updating a given context information, it notifies the context service using the group listener corresponding to each group whose definition includes this context information in the dissemination set.

5 Handling Privacy with Context Groups

Context information is often considered sensitive and should be disseminated with caution. Organizing context distribution in order to prevent a massive distribution of such information is the first step for building privacy mechanism proper to the context information. In this sense, the grouping mechanism we propose in this paper represents a low-level mechanism that can be used for privacy issues. The main idea, in this case, is to limit the dissemination of certain context information to specific context groups which are allowed to receive this information.

Using our context grouping, an application can express its grouping needs through the group profiles proposed in Section 4.2.1. These needs can reflect privacy issues handled by the application. For instance, the group profile for the gaming application in Fig. 5 uses the location information as criterion for diffusing device available resources, meaning that only peers in the same room shall receive this information. However, the location information itself is not distributed. Even if it is used by the application (for content adaptation, for instance), it remains local and it is not distributed to other peers. Actually, by defining different group profiles, an application can specify "diffusion levels" representing different neighborhoods in which a given context information can be disseminated, avoiding sensitive data to be distributed outside allowed groups. For instance, our gaming application can define a second context group disseminating user's availability (context group dissemination set) only for other users belonging to the same gaming community, based on the user profile information (group criteria). Through such group definitions, this gaming application expresses two different neighborhoods, one controlling the dissemination of device resources information (which should not be disseminated outside the user's current room), and another for user's availability information (which is addressed only to members of the user's gaming community). Some peers may belong to both

neighborhoods, by joining the corresponding groups, but peers belonging to just one group are not allowed to receive context information related to other group.

Since group criteria are expressed using MUSIC CQL queries, the definition of such neighborhoods handling sensitive context information depends on the expression power of the CQL [15] and on the context elements represented using the MUSIC context model [14] (see section 4.1). As discussed in [15], the CQL proposed in the MUSIC project [12] allows the expression of complex queries, exploiting logical constraints and semantic references to entities represented in the context model. Such complex queries allow the definition of very fine-grained groups, supplying an interesting mechanism for handling sensitive context information. However, it is worth noting that, the proposed context grouping mechanism alone is not enough to define secure privacy policies. This mechanism represents a first step further to the definition of such policies, providing a low-level mechanism capable of controlling the dissemination of context information.

6 Evaluation

In order to evaluate the proposed Context Grouping mechanism, we have developed a prototype which implements a meaningful subset of the functionalities described in the paper. In particular, we focused on: *(i)* validating the mechanism of group formation based on the context situation and on the application needs (Context Service); (*ii*) evaluating the mechanism of context data distribution among different peers which belong to the same group (Distribution Service). The Context Service, described in Section 6.1, was implemented on top of the MUSIC Middleware and the MUSIC Context Model. The Distribution Service, described in Section 6.2, was implemented leveraging the JXTA technology which allows point-to-point communication in a decentralised environment and that does not necessarily require any pre-existent infrastructure (ad-hoc networks).

6.1 Context Service Implementation

In order to provide this evaluation we implemented a prototype of the proposed Context Service able to process group creation requests. The group profile is described in XML (see Fig. 5). The Context Service parses it and interacts with the MUSIC Context Middleware to obtain context data that has to be evaluated. The first step implemented in this reasoning process is the identification of scopes involved in group criteria and of scopes belonging to the dissemination set. The group profile is translated into a corresponding query to be submitted to the Context Middleware that verifies through its query processor if the constraints are satisfied and, in the affirmative, returns the context data composing the dissemination set.

A specific peer belongs (or not) to a group depending on its current context thus the Context Service has to continuously monitor the context changes in order to update its membership to a group, if necessary. For that reason the Context Service acts as a Context Listener for the middleware subscribing itself to any context changes related to the scope under the criteria. At every context update the middleware notifies this event and the Context Service evaluates if group criteria are still verified; in

the affirmative an update group request is sent to the Distribution Service, in the negative the peer leaves the group invoking the corresponding method of the Distribution Service as well.

Since the MUSIC Context Middleware implementation is still in progress and some features are still missing, we implemented a simple simulator. The simulator supports the same middleware interfaces thus the prototype that already uses the available functionalities, could be easily integrated with the fully implemented middleware as soon as it will be available.

6.2 Distributed Service over JXTA

JXTA [10] is an open network platform designed for peer-to-peer computing which provides a common set of open protocols for peer-to-peer communication. Basically, it standardizes the manner in which peers perform common operations such as discovering each other, self-organizing into peer groups, communicating, etc. Developers can leverage open source implementations of JXTA protocols for creating point-to-point applications. Such openness and standardization of JXTA motivated the choice of this technology for this first prototype.

During the prototyping of the Distributed Service, we leveraged the features offered natively by the JXTA platform and by its open source or reusable implementation. In particular, we selected the Java library "Ubinet" [7] that eases the development of JXTA-based applications by providing high level access to basic peer-to-peer functionalities. In the following, we describe how we implemented the methods defined by the Distribution Service interfaces depicted in Fig. 7.

When the platform is launched, the Distribution Service initializes itself and starts the underlying JXTA platform by extending and customizing components provided by the Ubinet library. Then, the Context Service detects and creates one or more group profiles on behalf of the running application and invokes the *joinGroup()* method on the Distribution Service. In the current implementation, this method checks if the required group profile is already matched by existing groups discovered in the JXTA network. If there is a match, it joins such group; otherwise a new group is created by the current peer, and advertised over the JXTA network.

After joining a group, the Context Service can register itself as a listener for context changes in such group by invoking the *addGroupListener()* method on the Distribution Service. In order to register, the Context Service itself implements the *IGroupListener* interface, which allows a listener to be notified whenever a change occurs in the dissemination set of a peer participating to the group. The notification is driven by the Distribution Service, which receives messages containing a dissemination set from the peers that triggered a context change and, as a consequence, invokes the *ctxDataChangedInGroup()* method on each registered listener, delivering the corresponding context data.

As an alternative to this "push" notification mechanism, the Context Service can also retrieve context data related to other peers in a "pull" fashion, by invoking the *getDisseminationSet*() and *getDisseminationSets*() methods: the former provides the dissemination set for a given peer; the latter retrieves the dissemination sets for all the peers that joined a group.

The Context Service can deliver the related dissemination set by invoking the *updateGroup()* method on the Distribution Service whenever a change occurs in the local context data. This method leverages the JXTA functionalities to send a message to all the peers that joined the group, containing the new context data. When the group criteria are no more matched by the current context, the Context Service invokes the *leaveGroup()* method on the Distribution Service, which de-registers the current peer from the group and stops receiving messages from its participants.

6.3 Evaluation Results

For evaluation purposes, we have implemented a testing application using the implementations described above. This application considers the group profile represented in Fig. 5. Context information is statically defined: a thread simulates context updates on regular intervals of time (5s) considering a predefined set of context elements for each peer.

For this first experience, we preferred a well-controlled environment in which we could analyze the behavior of the proposed mechanism, without considering exceptions or abnormal conditions that could raise on mobile devices using wireless networks. Thus, we executed our testing application on several peers (from 2 up to 4 peers, with at least two matching the group criteria) using computers interconnected by an IEEE 801.3u network. We used Intel x86 processors (Core2 2GHz, Pentium D 3.2GHz or Pentium 4 2.8GHz processors) with 500MB up to 2GB of memory, running J2SE version 5 or version 6.

Our experiments showed that the overall behavior of our context grouping mechanism remains constant over the time. For example, the memory usage remains near the average (around 4.4MB, including testing application and MUSIC context middleware), as illustrated in Fig. 8. In this figure, we present the memory usage of different peers during 9,5min of execution. All peers present the same behavior. The sawtooth variation can be explained by the Java garbage collector. Such constant

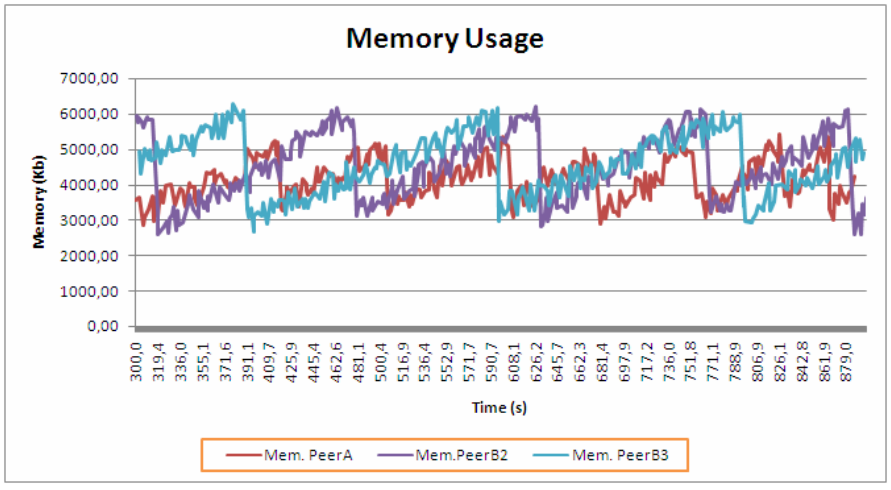


Fig. 8. Memory usage analysis

behavior is also visible on the response time of the observed peers. By simulating context updates on the group dissemination set on regular intervals of time, we could observe that the management of context groups generates no expressive overhead (testing application receives updates from remote nodes on slightly the same regular interval of time). Besides, Fig. 8 represents only peers belonging to the defined context group. During the experiment, we noticed that peers that do not participate to the defined group effectively do not receive any context update from peers belonging to the group. It is worth noting also that, as expected, context information unrelated to the group dissemination set is not disseminated among group members even if this information is part of the group criteria. In our example, location and user age information are not disseminated in the group since they do not belong to the dissemination set, even if they are used as context criteria.

This constant and predictable behavior under well-controlled environments is an important result since it validates the proposed mechanism. Once the expected behavior is verified and validated on "normal" conditions, we are able now to check it on "abnormal" and uncontrolled environments in our future experiments.

7 Conclusions

This paper introduces a new context grouping mechanism for context distribution. This mechanism allows applications to organize context distribution on context groups, which are defined based on common context information and on which only context information associated with the group can be disseminated. The benefits of the proposed mechanism are two-fold: on the one hand, it avoids context information to be indiscriminately spread over all available peers on the network; on the other hand, it acts as a low-level mechanism for privacy purposes, by preventing that peers external to context groups could receive the corresponding context information.

Considering the evaluation of the proposed mechanism, this paper presents the results of an initial implementation using the JXTA platform. Although limited, these results highlight the feasibility of our proposal and encourage us to continue further with the full integration of the context grouping mechanism in the MUSIC middleware [12]. As a future work we plan to implement the Distribution Service also on top of SIP [16] technology, for context distribution over WLAN. In this case, the distribution service is built using a centralized approach, considering a SIP proxy acting as a context group repository. This will allow us to compare the behaviour of our proposed Context Grouping mechanism under two different technologies.

Among all the future directions foreseen by this paper, we are particularly interested on the implicit context grouping definition. In this case, instead of assuming that applications are aware of grouping mechanism and actively request forming groups, we consider to implicitly define these groups by analyzing context information requests performed by applications using information clustering and pattern recognition techniques.

Acknowledgements. The authors would like to thank their partners in the MUSIC-IST project and acknowledge the partial financial support given to this research by the European Union (6th Framework Programme, contract number 35166).

References

1. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. International Journal of Ad. Hoc. and Ubiquitous Computing 2(4), 263–277 (2007)
2. Bryan, D., Matthews, P., Shim, E., Willis, D.: Concepts and Terminology for Peer to Peer SIP, IEFT Draft (2008), http://tools.ietf.org/html/draft-ietf-p2psip-concepts-01
3. Cutting, D., Quigley, A.,, Landfeldt, B.: Special Interest Messaging: A Comparison of IGM Approaches. The Computer Journal (2007) (Access published, December 2, 2007), http://comjnl.oxfordjournals.org/cgi/content/abstract/bxm076v1
4. Delmastro, F., Passarella, A., Conti, M.: P2P Multicast for pervasive ad-hoc networks. Pervasive and Mobile Computing 4(1), 62–91 (2008)
5. Harjula, E., Ala-Kurikka, J., Howie, D., Ylianttila, M.: Analysis of peer-to-peer SIP in a distributed mobile middleware system. In: IEEE Global Telecommunications Conference (Globecom 2006), pp. 1–6 (2006)
6. Henricksen, K., Indulska, J., McFadden, T., Balasubramaniam, S.: Middleware for distributed context-aware systems. In: International Symposium on Distributed Objects and Applications (DOA) (2005)
7. http://www.igd.fhg.de/igd-a1/dynamite-project/software/tools/ubinet-javadoc.zip
8. Hu, X., Ding, Y., Paspallis, N., Bratskas, P., Papadopoulos, G.A., Barone, P., Mamelli, A.: A Peer-to-Peer based infrastructure for Context Distribution in Mobile and Ubiquitous Environments. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 236–239. Springer, Heidelberg (2007)
9. In, H.P., Meintanis, K.A., Zhang, M., Im, E.G.: Kaliphimos: A Community-Based Peer-to-Peer Group Management Scheme. In: Yakhno, T. (ed.) ADVIS 2004. LNCS, vol. 3261, pp. 533–542. Springer, Heidelberg (2004)
10. JXTA Community, https://jxta.dev.java.net/
11. Khambatti, M., Ryu, K.D., Dasgupta, P.: Structuring peer-to-peer networks using interest-based communities. In: Aberer, K., Kalogeraki, V., Koubarakis, M. (eds.) VLDB 2003. LNCS, vol. 2944, pp. 48–63. Springer, Heidelberg (2004)
12. MUSIC Consortium, Self-Adapting Applications for Mobile Users in Ubiquitous Computing Environments (MUSIC), http://www.ist-music.eu/
13. Paganelli, F., Bianchi, G., Giuli, D.: A Context Model for Context-Aware System Design Towards the Ambient Intelligence Vision: Experiences in the eTourism Domain. In: Stephanidis, C., Pieper, M. (eds.) ERCIM Ws UI4ALL 2006. LNCS, vol. 4397, pp. 173–191. Springer, Heidelberg (2007)
14. Reichle, R., Wagner, M., Khan, M.U., Geihs, K., Lorenzo, L., Valla, M., Fra, C., Paspallis, N., Papadopoulos, G.A.: A Comprehensive Context Modeling Framework for Pervasive Computing Systems. In: Meier, R., Terzis, S. (eds.) DAIS 2008. LNCS, vol. 5053. Springer, Heidelberg (2008)
15. Reichle, R., Wagner, M., Khan, M.U., Geihs, K., Valla, M., Fra, C., Paspallis, N., Papadopoulos, G.A.: A Context Query Language for Pervasive Computing Environments. In: 5th IEEE Workshop on Context Modeling and Reasoning (CoMoRea), 6th IEEE Int. Conf. on Pervasive Computing and Communication (PerCom), pp. 434–440 (2008)
16. Rosenberg, J.: SIMPLE made Simple: An Overview of the IETF Specifications for InstantMessaging and Presence using the Session Initiation Protocol (SIP), IETF Draft, Expires August 27 (2008)
17. Ye, J., Li, J., Zhu, Z., Gu, X., Shi, H.: PCSM: A Context Sharing Model in Peer-to-Peer Ubiquitous Computing Environment. In: International Conference on Convergence Information Technology, November 21-23, pp. 1868–1873 (2007)

Annex IV

Paper

Cassales, G.W.; Charão, A.S.; Kirsch-Pinheiro, M.; Souveyet, C. & Steffenel, L.-A. “Improving the performance of Apache Hadoop on pervasive environments through context-aware scheduling”, *Journal of Ambient Intelligence and Humanized Computing*, 7(3), 2016, 333-345.

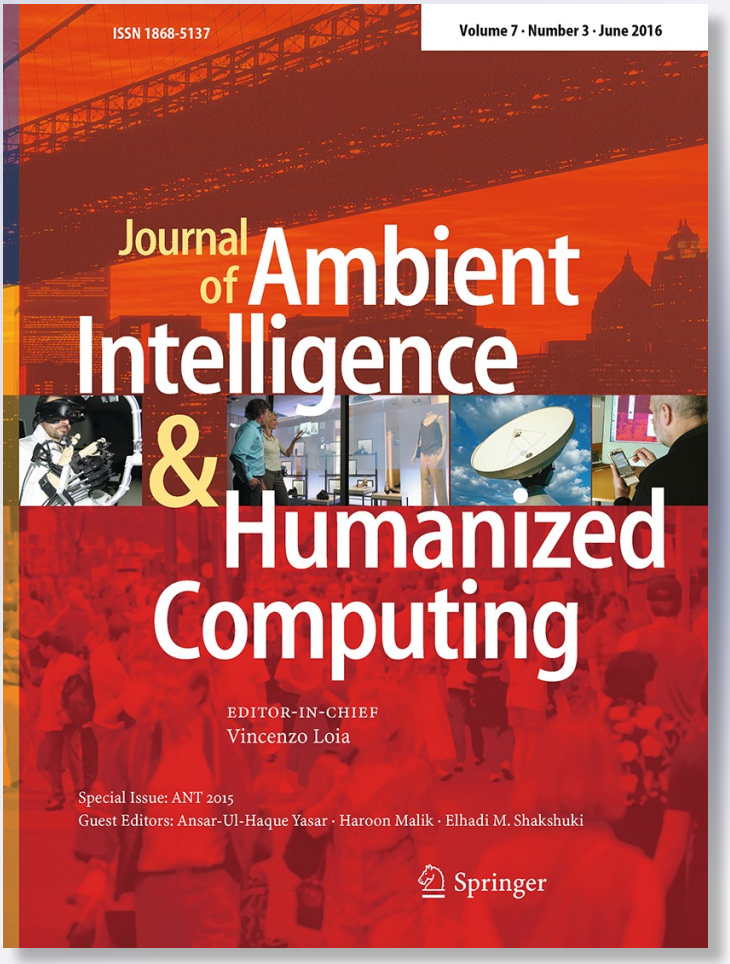
Improving the performance of Apache Hadoop on pervasive environments through context-aware scheduling

Guilherme W. Cassales, Andrea Schwertner Charão, Manuele Kirsch-Pinheiro, Carine Souveyet & Luiz-Angelo Steffenel

Journal of Ambient Intelligence and Humanized Computing

ISSN 1868-5137
Volume 7
Number 3

J Ambient Intell Human Comput (2016)
7:333-345
DOI 10.1007/s12652-016-0361-8



Springer

J Ambient Intell Human Comput (2016) 7:333–345
DOI 10.1007/s12652-016-0361-8

ORIGINAL RESEARCH

Improving the performance of Apache Hadoop on pervasive environments through context-aware scheduling

Guilherme W. Cassales[1] · Andrea Schwertner Charão[1] · Manuele Kirsch-Pinheiro[2] · Carine Souveyet[2] · Luiz-Angelo Steffenel[3]

Received: 5 October 2015 / Accepted: 10 January 2016 / Published online: 9 March 2016

Abstract This article proposes to improve Apache Hadoop scheduling through a context-aware approach. Apache Hadoop is the most popular implementation of the MapReduce paradigm for distributed computing, but its design does not adapt automatically to computing nodes' context and capabilities. By introducing context-awareness into Hadoop, we intent to dynamically adapt its scheduling to the execution environment. This is a necessary feature in the context of pervasive grids, which are heterogeneous, dynamic and shared environments. The solution has been incorporated into Hadoop and assessed through controlled experiments. The experiments demonstrate that context-awareness provides comparative performance gains, especially when some of the resources disappear during execution.

✉ Luiz-Angelo Steffenel
luiz-angelo.steffenel@univ-reims.fr

Guilherme W. Cassales
cassales@inf.ufsm.br

Andrea Schwertner Charão
andrea@inf.ufsm.br

Manuele Kirsch-Pinheiro
manuele.kirsch-pinheiro@univ-paris1.fr

Carine Souveyet
carine.souveyet@univ-paris1.fr

[1] Laboratório de Sistemas de Computação, Universidade Federal de Santa Maria, Santa Maria, Brazil

[2] Centre de Recherche en Informatique, Université Paris 1 Panthéon-Sorbonne, Paris, France

[3] Laboratoire CReSTIC—Équipe SysCom, Université de Reims Champagne-Ardenne, Reims, France

Abbreviations

API	Application programming interface
DHT	Distributed hash table
FIFO	First in, first out
HDFS	Hadoop distributed file system
P2P	Peer-to-Peer
PER-MARE	Pervasive map-reduce project
SLA	Service-level agreement
VM	Virtual machine
YARN	Yet another resource negotiator

1 Introduction

Apache Hadoop is a popular framework for distributed and parallel computing. It implements the MapReduce programming paradigm, which aims at processing big datasets (Dean and Ghemawat 2008) and to scale up from a single server to thousands of machines.

Without specific configuration by the administrator, Apache Hadoop supposes the use of dedicated homogeneous clusters for executing MapReduce applications. As the overall performance depends on the task scheduling, Hadoop performance may be seriously impacted when running on heterogeneous and dynamic environments it was not designed for.

This is a special concern when deploying Hadoop over pervasive grids. Pervasive grids are an interesting alternative to costly dedicated clusters, as the acquisition and maintenance of a dedicated cluster remain high and dissuasive for many organizations. According to Parashar and Pierson (2010), pervasive grids represent the extreme generalization of the grid concept, in which the resources are pervasive. Pervasive grids use resources embedded in

pervasive environments to perform computing tasks in a distributed way. Concretely, they can be seen as computing grids formed by existing resources (desktop machines, spare servers, etc.) that occasionally contribute to the computing grid power. These resources are inherently heterogeneous and potentially mobile, dynamically joining and leaving the grid. Knowing that, in essence, pervasive grids are heterogeneous, dynamic, shared and distributed environments, their efficient management becomes a very complex task (Nascimento et al. 2008). Task scheduling is thus severely affected by the environment complexity.

Many works proposed to improve the Hadoop framework on environments that diverge from the original working specifications (Kumar et al. 2012; Zaharia et al. 2008; Rasooli and Down 2012; Sandholm and Lai 2010). The PER-MARE project (STIC-AmSud 2014), in which this work was developed, aims at adapting Hadoop to pervasive environments (Steffenel et al. 2013).

Therefore, adapting the execution to dynamic environments is a necessity as Hadoop is based on static configuration files that do not adapt to resources variations. Attempting install on heterogeneous clusters imply for the administrators to manually set the characteristics for each resource, a repetitive and time consuming task. All these factors prevent deploying Hadoop on more volatile environments, and our objective is to improve Hadoop so that it could adapt itself to the execution context and therefore be deployed over pervasive grids.

In order to adapt Hadoop to a pervasive grid environment, supporting context-awareness is essential. Context-awareness is the capacity of an application or software to detect and respond to environment changes (Maamar et al. 2006). A context-aware system is able to adapt its operations without human intervention, therefore improving the usability and efficiency of the system (Baldauf et al. 2007). In pervasive grids, context-aware data may help task schedulers to make better decisions based on real feedback from the system.

This work focuses on our developments to introduce context-awareness capabilities on Hadoop task scheduling mechanisms. Through a context collection procedure and minimal changes on Hadoop's resource manager, we are able to update the information about the availability of resources in each node of the grid and then influence the scheduler tasks assignments. It also extends the preliminary observations from Cassales et al. (2015) through the analysis of additional performance benchmarks.

The rest of the paper is organized as follows: Sect. 2 presents Apache Hadoop architecture and scheduling mechanisms. Section 3 discusses related work, focusing on context-awareness and on other works that try to improve Hadoop schedulers. Section 4 presents our proposal of context-aware scheduling, while Sect. 5 presents the experiments conducted and the results achieved. Section 6 introduces general discussion about the results and future challenges. We finally conclude this paper in Sect. 7.

2 About Hadoop scheduling

Before discussing related works and presenting our proposal, it is worth introducing some basic concepts about the Hadoop framework and its schedulers.

2.1 Hadoop framework architecture

The current Apache Hadoop framework is organized as a master and slave architecture, with two main services: storage (HDFS) and processing (YARN). Both services have their own master and slave components, as presented on Fig. 1: the *NameNode* and *ResourceManager* services are the masters of the HDFS and YARN respectively, and the *DataNode* and *NodeManager* their slave counterparts. It is also possible to note the *ApplicationMaster*, the component responsible for internal application management (task scheduling), while *ResourceManager* is responsible for job scheduling. Each node also runs a set of *Containers*, where the execution of Map and Reduce tasks takes place.

2.2 Hadoop schedulers

The Hadoop framework has two execution layers that depend on schedulers, according to their granularity. The coarse-grain instance is the job, while the fine-grain instance is represented by the tasks that compose a job.

Job-level scheduling is performed by the *ResourceManager*, an entity that has a global view of the available resources and can, for instance, arbitrate available resources among the competing applications. Each application is associated to an *ApplicationMaster* that tries to acquire the right to use a given number of resource units (*containers*) from the *ResourceManager*; the latter must optimize the resources utilization according to different constraints such as capacity guarantees, fairness, and SLAs.

The *ResourceManager* can be optimized to different constraints and parameters through the use of pluggable schedulers. The simplest scheduler algorithm, *Internal Scheduler*, processes all jobs in the order of arrival (FIFO). This scheduler performs well in dedicated clusters where the competition for resources is not a problem. Another scheduler available is the *Fair Scheduler*, which uses a two level scheduling to fairly distribute the resources among batches of small jobs (Apache 2014). A third widely known scheduler is the *Capacity Scheduler*, which is designed to run Hadoop MapReduce in a shared, multi-tenant cluster. The Capacity Scheduler is designed to allow sharing a large cluster while giving each organization a minimum

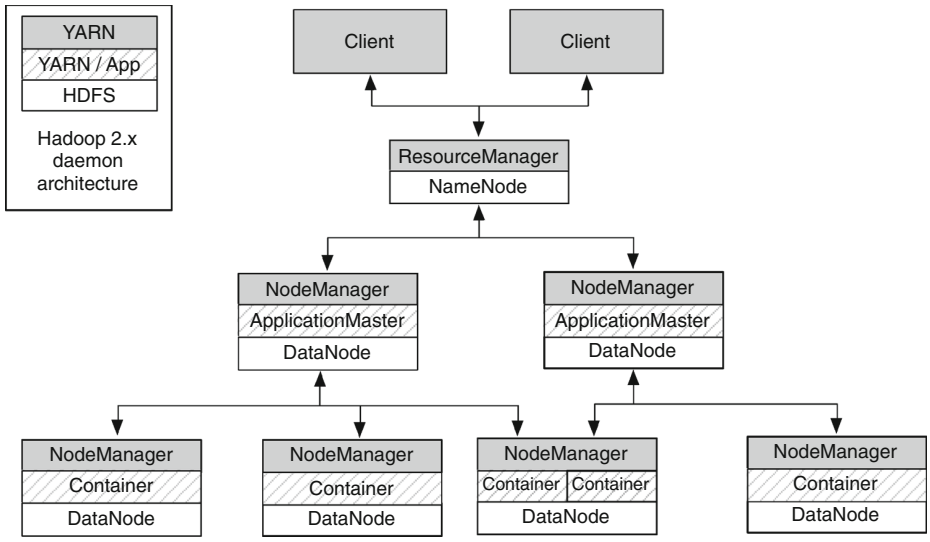


Fig. 1 General Apache Hadoop architecture

capacity guarantee. The central idea is that the available resources in the Hadoop MapReduce cluster are partitioned among multiple organizations that collectively fund the cluster based on computing needs (Apache 2014).

These schedulers allows a flexible management of the framework at the job level. Yet, the available schedulers neither detect nor react to the dynamicity and heterogeneity of the computing environment, a requirement for pervasive grids. Indeed, the design of Hadoop YARN passes on this responsibility to the *Application Master*, which can better adapt to the application needs.

At a fine-grain, scheduling is performed by the *ApplicationMaster* according to the number of tasks and assigned resources. There is almost no documentation on the default task scheduler, but we can assume that it implements a progressive filling approach where tasks are fed to all granted containers in one node before starting filling the next node. This behavior was experimentally observed in (Cassales et al. 2014).

It is also worth noting that the *ApplicationMaster* is not aware of the real execution context as it relies on the resources granted by the *ResourceManager*, i.e., it has only a limited view on the computing platform. Modifying the *ApplicationMaster* scheduling algorithms to consider real context-awareness requires changes on the *ResourceManager* itself. After reviewing some related work on the next session, we will introduce our approach to bring context-awareness to Hadoop.

3 Related work

Over the years, different works proposed improvements to the Hadoop scheduler mechanisms in order to respond to specific needs. These contributions may be classified as new scheduling methods or as improvements for the resource distribution.

Works like Kumar et al. (2012), Tian et al. (2009) and Rasooli and Down (2012) assume that most applications are periodic and demand similar resources regarding CPU, memory, network and hard disk load. These assumptions allow the applications and nodes to be analyzed regarding the CPU and I/O potential, enabling the optimization of execution through matching of nodes and applications with the same characteristics. Isard et al. (2009) focus on a new scheduling method proposing the usage of a capacity-demand graph that assists the calculation of optimal scheduling based on an overall cost function.

While previous works focus on performance improvement using static information about resources and applications, other works sought to incorporate task specific information into their proposals. For example, Zaharia et al. (2008) and Chen et al. (2010) attempted to improve tasks distribution as a way to reduce the response time in large clusters. Zaharia et al. (2008) use heuristics to infer the estimated task progress and to make a decision about the launching of speculative tasks. Speculative tasks are launched when there is a possibility that the original task is on a node either faulty or too slow node. Another work (Chen et al. 2010) proposes the use of historical execution data to improve decision making.

Both scheduling mechanics and resource distribution methods result in a load rebalancing, forcing faster nodes to process more data and slower nodes to process less data. Sandholm and Lai (2010) try to achieve that through a system based on resource supply and demand, allowing each user to directly influence scheduling through spending rates. The main objective is to allow a dynamic resource sharing based on preferences set by each user.

There are also works such as Xie et al. (2010), which attempt to provide a performance boost in jobs through better data placement, mainly using data location as information to decision making. The performance gain is achieved through data rebalancing on nodes, increasing the load on faster nodes. This proposal reduces the number of speculative tasks and data transfers over the network. A similar proposal is observed by Cavallo et al. (2015), which address scheduling and data distribution issues on geographically distributed clusters. These authors present a new hierarchical scheduling mechanism based mainly on throughput and application data profiling. As for Xie et al. (2010), Cavallo et al. (2015) also focus on optimizing data transfer through the network.

Marozzo et al. (2012) uses a P2P structure to arrange the cluster. In this approach, nodes can change their function (master/slave) over time and can have both functions at the same time, the functions being tied to the applications and not to the cluster. The objective of this work is the adaptation of MapReduce paradigm to a P2P environment, which given the natural volatility of P2P environments, would offer support to pervasive grids. However, this proposal focuses on providing a resilient infrastructure and does not explore the scheduling of jobs and tasks. Another work relying on a P2P overlay, Steffenel and Kirsch Pinheiro (2015) offers MapReduce-like computing for pervasive environments on top of a general distributed computing platform. However, this platform focuses on fault-tolerance and volatility aspects through a fully distributed task scheduling mechanism, which for the moment does not implement context-aware scheduling optimizations.

Indeed, most of previously cited works do not actually consider the evolving state of the available resources. Resources are described, not observed. However, works on context-aware computing (Baldauf et al. 2007; Maamar et al. 2006; Ramakrishnan et al. 2014; Najar et al. 2015) have demonstrated that this observation is possible and that the execution environment may influence application behavior. This raises a question: can we improve MapReduce scheduling by observing current execution environment? The next sections will try to answer this question.

4 Context-aware scheduling

The main goal of this work is to improve the scheduling of Hadoop by adding support to dynamic changes at the *Resource Manager* level. Unlike the works on Sect. 3 we opted to feed dynamic context information to an existing scheduler (*Capacity Scheduler*) and therefore modifying the Hadoop source code the least possible.

In the default Hadoop implementation, a *NodeManager* declares its computing resources to the *ResourceManager* when joining the Hadoop network, this information being usually obtained from static configuration files. In order to detect dynamic changes, the scheduler must collect context information that, in this case, refer to available resources on the nodes. Then, a *NodeManager* communicates periodically with the *ResourceManager* in order to keep information updated and let the scheduler adapt to the new context. In the following section we present a more detailed explanation of the changes implemented in Apache Hadoop.

4.1 Context collector

By default, Hadoop reads information about the nodes from XML configuration files. These files contain many configuration parameters, including the resource capacity of each node. Once loaded, the information will not be updated until the service is restarted. As pervasive environments may face performance changes during the execution of an application, we need a mechanism that collects contextual information at runtime and subsequently updates the *ResourceManager* knowledge base.

Therefore, we integrate into Hadoop a collector module, allowing to observe contextual information about the available resources. The collector was developed for the PERMARE project (STIC-AmSud 2014), and its class diagram is presented in Fig. 2 (Cassales et al. 2014). The collector module is based on the standard Java monitoring API (Oracle 2014), which allows to easily access the real characteristics of a node. It allows collecting different context information such as the number of processors (cores) and the system memory using a set of interface and abstract/concrete classes that generalize the collecting process. Due to its design, it is easy to integrate new collectors and improve the resources available for the scheduling process, providing data about the CPU load or disk usage, for example.

Nevertheless, replacing XML configuration files by the context collector is not enough, since a global knowledge of the available resources is necessary in order to adapt scheduling to runtime conditions. Indeed, to allow *ResourceManager* to adapt scheduling to the current available resources, the context collector associated to each node must be able to communicate its current state to the *ResourceManager* during the execution. In order to do so, we have improved communication capabilities of both *ResourceManager* and *NodeManager* as explained in the following section.

4.2 Communication

Gathering the context information requires to feed the Hadoop scheduler requires transmitting this information through the network from slave nodes (*NodeManager*) to the master node (*ResourceManager*) which is in charge of the scheduling. Instead of relying on a separate service, we

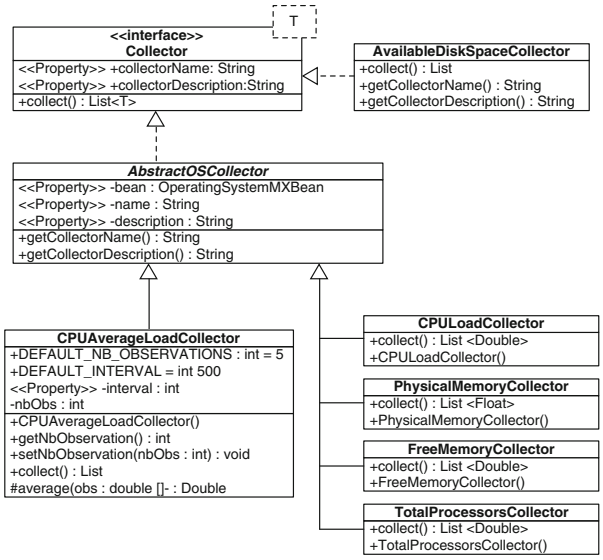


Fig. 2 Context collector structure

chose to use the ZooKeeper API (Hunt et al. 2010) that provides efficient, reliable, and fault-tolerant tools for the coordination of distributed systems. In our case, ZooKeeper services are used to distribute context information.

As illustrated in Fig. 3, all slaves (*NodeManager*) run an instance of the *NodeStatusUpdater* service, which collects data about the real resources availability (for example, every 30 s). If the amount of available resources changes, the DHT on ZooKeeper will be updated. Since the Operating System produces some variations on the resources, this information will only be changed if the variation is big enough to raise/lower the maximum capacity of containers on that node. This small change will spare a lot of executions where the information changed, but the variation was not enough to impact the scheduling. Similarly, the master (*ResourceManager*) also creates a service to watch ZooKeeper's zNodes. If the Zookeeper node detects a DHT change, the master will be notified and will update the scheduler information based on the new information. This solution extends a previous one presented in Cassales et al. (2014) by offering a real time observation of available nodes. Indeed, our previous solution only updates information regarding the resources on service initialization, replacing the XML configuration file, while this one updates resource information whenever the availability changes. As a result, scheduling is performed based on the current resource state.

5 Experiments and results

In order to evaluate the impact of context awareness on the job scheduler mechanism, we performed two sets of experiments. In the first one, presented on Sects. 5.3 and 5.4, we considered a small dedicated cluster so that the stress of incorrect resources estimation could easily be controlled and analyzed. On the second set of experiences, presented on Sects. 5.5 and 5.6, we run the TeraSort benchmark in a larger number of nodes but a real shared environment (i.e., with nodes shared with other users and applications). Both the first and the second experiments considered different execution scenarios as well as different application profiles, as presented in the following section.

5.1 Execution scenarios

In the context of this work we compare the Hadoop behavior under different execution scenarios using the available memory and the number of nodes (*v-cores*) as resource metrics. These parameters are reported to the *ResourceManager* and represent the main elements considered by the Capacity Scheduler algorithm. In addition, the available memory parameter can be considered as closely related to the node real environment (contrarily to the *v-cores*) and is easily configurable through the *yarn.nodemanager.resource.memory-mb* property.

The execution scenarios were tailored to reproduce different combinations of reported resources, available resources and context awareness. Indeed, the execution performance is closely related to the assignment of computing resources to the applications. Incorrect configuration parameters or external factors may heavily impact the nodes performance, especially when these parameters lead to the resources overload. Similarly, changes on the available resources during the execution may drive a well-configured environment to an overloaded state if the changes are not reported to the scheduler. The four scenarios described below try to reproduce these situations, thus providing sufficient information for the analysis and discussions at the end of this paper.

Scenario A: in this scenario we simulate a dedicated Hadoop cluster so that the reported memory always correspond to the available memory, which can be considered as the "best case" scenario. We consider the reported memory as the information that the scheduler use in the scheduling process, while available memory is the free memory of the node or cluster. Using a direct notation, the reported memory is 100 % and the available memory is also 100 % all through the execution.

Scenario B: in this case nodes can be used for other purposes than running Hadoop, so the reported memory may at some point differ from the available memory initially configured for Hadoop usage. This case corresponds to the default behavior of Hadoop, where memory resources are provided through a property *yarn.nodemanager.resource.memory-mb*. Because the scheduler never

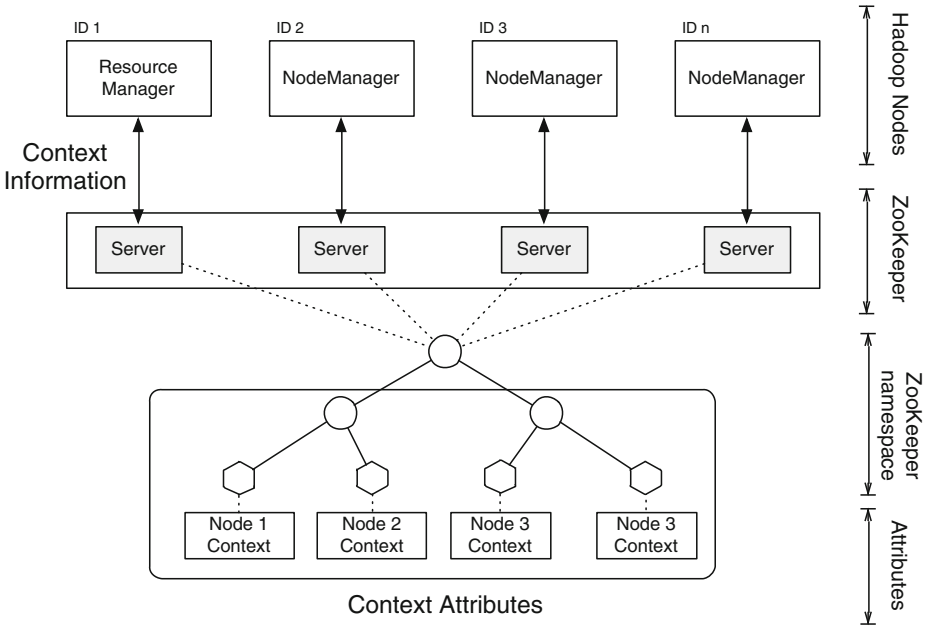


Fig. 3 Using ZooKeeper to distribute context information

adapts to the reduced resources, we consider it the "worst case" scenario. On this scenario, using a direct notation, the reported memory is 100 %, but the available memory is 50%.

Scenario C: in this case, the nodes are shared with other applications (like in Scenario B) but the context-awareness collector is active, updating context information every 30 seconds. When the MapReduce application is launched, the scheduler is aware of the execution context and can assign tasks according to the available resources. Using a direct notation, the reported memory and the available memory correspond to 50 % of the values reported on Scenario A.

Scenario D: this scenario presents an extension of Scenario C where the MapReduce application starts before the context collector update. The scheduler starts with wrong available resource information and must therefore adapt during the execution with the help of the context collector. Using a direct notation, the reported memory at the beginning of the execution is 100 % of the resources from Scenario A (wrong information), but at runtime this information is updated to 50 %.

Scenario A corresponds to the "best case" for Hadoop framework, in which all resources are available during the entire execution. Scenario B simulates Hadoop execution in a heterogeneous environment where resources fluctuate during the execution but the scheduler is never aware of the changes (a "worst case"). If we consider context information, Scenario C illustrates a situation in which the context collector is able to detect environment changes before application execution starts while scenario D suffers from late information update but can still adapt to the changes. These two scenarios allow us to evaluate the impact of the context-aware scheduler during the deployment of an application in a heterogeneous environment.

5.2 Application benchmarks

While big-data applications are expected to rely on memory, other factors like CPU usage and I/O may also impact the execution of the applications. For this reason, this work uses three different benchmarks, each characterized by different memory, CPU and I/O usages (Li et al. 2013). The three benchmarks are the following:

- TeraSort: The goal of TeraSort benchmark (Hamilton 2008) is to sort a given amount of data as fast as possible. It is a benchmark that combines testing the HDFS and MapReduce layers of a Hadoop cluster. Because of the sorting algorithms, this benchmark stress memory and CPU;
- WordCount: the WordCount benchmark is the basic MapReduce example. Its objective is to count the number of occurrences of each word in a given text. Because WordCount memory and I/O usage are limited (both memory and output structures are small in comparison to the input file), the main performance is notably determined by the CPU;
- TestDFSIO: The TestDFSIO benchmark is a read and write test for HDFS. It is helpful for tasks such as stress testing HDFS, to discover performance bottlenecks in the network, OS and Hadoop setup; it aims at giving a first impression of how fast a cluster is in terms of I/O. Memory and CPU are less solicited.

For the experiments we use the HiBench benchmarks suite, which is detailed in Huang et al. (2010). The TeraSort benchmark was run using a data set of 15 GB, TestDFSIO was run with 90 files of 250 MB and WordCount used a 10 GB file as input.

5.3 Environment setup and configuration for the controlled experiments

In order to test the new behavior of the framework, we conducted experiments with the Grid'5000 platform (Grid'5000 2013). We configured a dedicated network with 4 slaves, each having the following configuration: 2 Intel Xeon CPU E5420 @ 2.50 GHz (totalizing 8 cores per node) and 8 GB of RAM. All nodes run Ubuntu-x64-12.04, with JDK 1.7 installed, and the Apache Hadoop 2.5.1 distribution.

The resources evaluated in the experiments were the memory and number of cores, which have a direct impact on the amount of Map tasks allocated. In addition to the overall performance of each benchmark on the different scenarios, we performed an in-depth investigation on the tasks (containers) distribution throughout the execution. The information about the execution of each container was obtained from the Hadoop logs. These information allow us to present detailed Gantt charts of the tasks placement during the experiments.

To emulate the scenarios with reduced resources (Scenarios B, C and D), we opted to halve the number of nodes while reporting the same amount of global available memory as in Scenario A. While this approach is unrealistic (resources from failing nodes should be withdrawn from the available resources set), it provides a clear reference for the analysis of the scheduler decisions.

5.4 Results from the controlled experiments

The results from experiments are presented in both Tables 1, 2, 3 and Figs. 4, 5, 6, respectively for TeraSort, TestDFSIO and WordCount. All Tables summarize the experiments presenting the total time used by all map tasks, the average execution time, the standard deviation and the number of speculative tasks launched on each scenario and for each benchmark.

Figures 4, 5 and 6 present the Gantt charts for each benchmark and scenario. For a given benchmark, each scenario is composed by either 2 or 4 lines, one for each node in the cluster during the experiment. As stated in the description of the scenarios, Scenarios B, C and D run on half of the nodes in Scenario A to simulate a reduced amount of resources. On each line we consolidate the resources according to a color scale where the darker the tone, the more containers run simultaneously and thus, more overloaded the node is. For instance, white means no container executing, and black means 16 containers. Additionally, the lines are segmented along the chart to indicate that a container has either finished or started at that moment. The charts are scaled in seconds and all charts for a given benchmark have the same scale. Hence, TeraSort charts go from 0 to 790 s, TestDFSIO go from 0 to 450 s and WordCount from 0 to 1010 s.

Table 1 Summary of TeraSort results expressed in seconds

Scenario	A	B	C	D
Total map time (*s*)	149	788	348	477
Avg. map time (*s*)	39.47	222.97	38.38	68.42
Standard deviation	15.73	59.86	18.09	29.91
# Map tasks	76	76	76	76
# Speculative tasks	2	1	3	1

Table 2 Summary of TestDFSIO results expressed in seconds

Scenario	A	B	C	D
Total map time (*s*)	139	444	239	364
Avg. map time (*s*)	38.95	85.01	32.20	81.62
Standard deviation	17.20	69.08	8.30	73.60
# Map tasks	90	90	90	90
# Speculative tasks	0	9	0	1

Table 3 Summary of WordCount results expressed in seconds

Scenario	A	B	C	D
Total Map Time (*s*)	155	1009	309	805
Avg. Map Time (*s*)	43.76	208.39	41.73	175.80
Standard Deviation	15.61	128.90	10.99	151.59
# Map Tasks	90	90	90	90
# Speculative Tasks	1	15	1	10

While the Gantt charts represent the tasks (containers) execution, some containers are not affected by scheduling, such as the *ApplicationMaster* or the Reduce tasks. For this reason, we ignore these tasks and concentrate the analysis on the elements that can be affected by the context-aware scheduling.

From the analysis of the Tables, it is possible to identify some common trends. Indeed, all experiments present a similar behavior when we consider the Total Map Time: case A was the fastest, followed by C, D, and finally B. In addition, scenarios A and C showed not only the smallest Average Map Times and Standard Deviations among their applications, but also very similar results in these categories, regardless the benchmark. This is due to the fact that the nodes were never overloaded since the scheduler had an updated information before the start of the

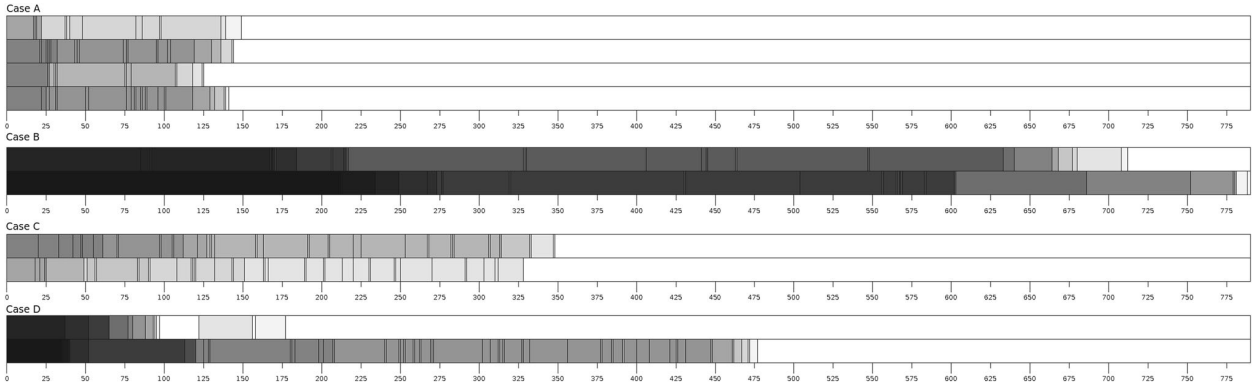


Fig. 4 Gantt charts of the TeraSort experiments

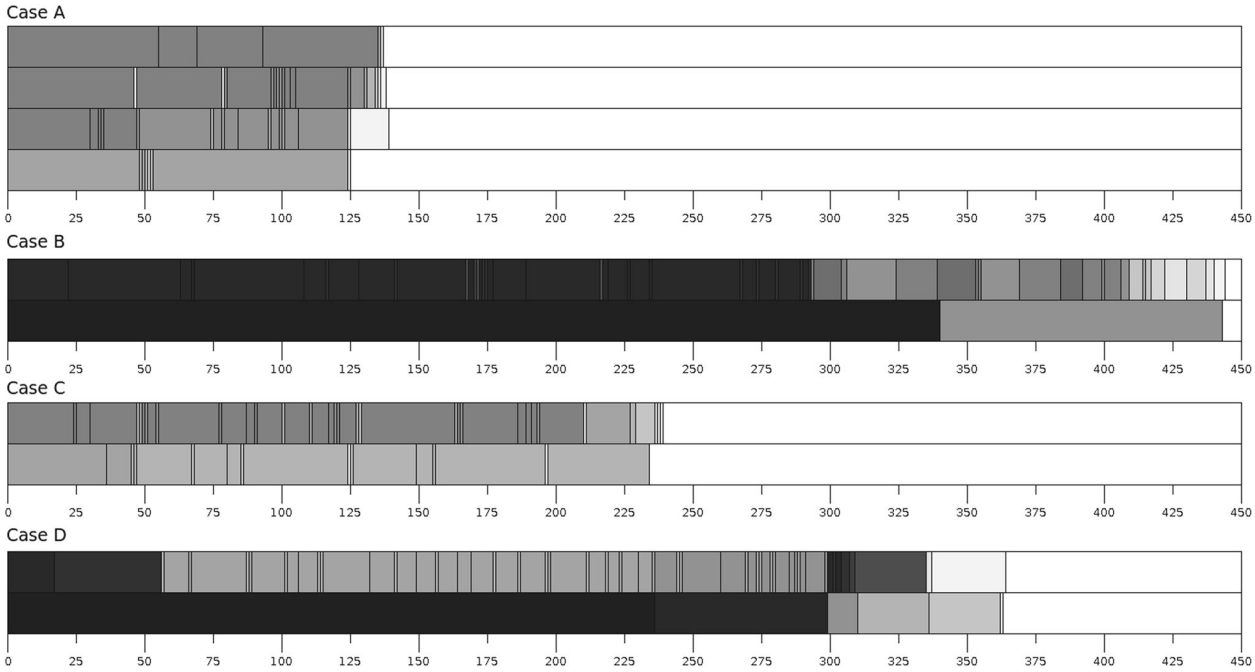


Fig. 5 Gantt charts of the TestDFSIO experiments

application. The charts also corroborate this analysis: in all experiments, cases A and C have much lighter tones than the others, meaning that less containers are running simultaneously. It is also worth noting that case C takes approximately twice the time case A used to complete the execution, a result to be expected since case C had half the resources of case A.

The analysis of the number of speculative tasks also gives interesting insights. In the case of TeraSort, the number of speculative tasks is reduced and almost similar in all scenarios. TestDFSIO and WordCount, on the other hand, deploy many more speculative tasks when the system is overloaded (scenarios B and D). This may be due to the factors that trigger a speculative task: a speculative task is launched only after all the tasks for a job have been launched, and then only for tasks that have been running for some time (at least a minute) and have failed to make as much progress, on average, as the other tasks from the job. In the case of TeraSort, tasks evenly require both memory, CPU and I/O, so their execution time is less subject to specific resource depletion. TestDFSIO and WordCount, on the contrary, rely on more specific resources and thus are more subject to resource overload. In all the cases, the use of context-awareness on scenario D helps minimizing the number of speculative tasks when comparing to the "context-unaware" scenario B.

Concerning the execution flow, as illustrated by the Gantt charts, it is possible to note that both cases B and D

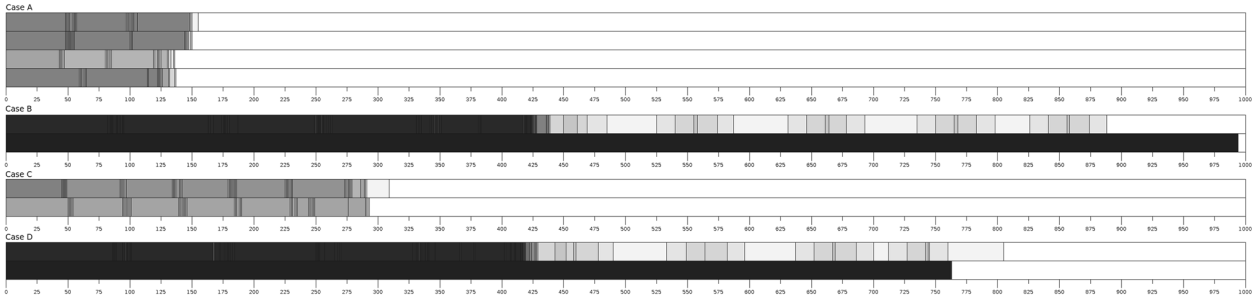


Fig. 6 Gantt charts of the WordCount experiments

have a dark tone at the beginning, meaning that 16 containers are running (twice the real capacity). Also, the first containers took 20 to 50 s to complete execution in scenarios A and C in all experiments (cf. the first segment line). On the opposite side, during most of scenario B (and to a lesser extent D) 70 or more seconds are required, evidencing an overload on the nodes. The exception here is the TestDFSIO scenarios B and D, which have segment lines at about 25 seconds. Through further analysis, it is possible to note that all TestDFSIO experiments had at least one segment line near the 25 s mark, meaning this was a task very easily processed and somehow not affected by the node overload.

Although both B and D scenarios had the same initial conditions (50 % available resources and 100 % reported resources), case D took less time to complete in all experiments (20 to 40 % faster than B). The reason for this is that the context collector updates the reported resources in D, allowing the scheduler to reorganize tasks after the first tasks complete. This behavior is easily noted on TeraSort and TestDFSIO charts. Indeed, all D scenarios had high concentration of executing containers at the start of the job but lessened the load over time, while B scenarios had the nodes overloaded until the end due to the absence of updated information. Although the scheduler does not preempt excess containers, it is possible to observe a performance improvement of about 40 % on TeraSort and 20 % on TestDFSIO experiments as the scheduler avoids overloading the nodes.

There are some specificities that might be pointed out to better understand the charts. On all benchmarks, the first node is seemingly less charged in scenarios B and D after the initial executions. This is mostly due to the fact that after the initial computations this node hosts the Reduce containers, which were not included in the chart. It is important to remember that reducer tasks may start before the end of map tasks and that a reducer may even complete before all maps have finished processing.

Although other specificities about the characteristics of the proposed jobs can be discussed, scenarios C and D show that regular context updates contribute to reduce the execution time on a dynamic Hadoop cluster. We demonstrated that even when starting with the same circumstances as in the worst case (Scenario B), updating the information helps the scheduler to minimize the execution time. Our solution contributes therefore to both provide correct information before the execution starts (Scenario C) and adapt the execution to resources changes (Scenario D).

5.5 Environment setup for the shared execution environment

As stated previously, this second set of experiments considered a more realistic environment, where we use 10 slave nodes in which part of the nodes are shared with another user. Indeed, most resource management systems (Slurm, Grid'5000 OAR) allow users to reserve only part of a node (for example, 8 from 16 cores). As the resources are not always evenly shared and some applications don't respect the limits from the allocated resources, this makes an interesting environment for the context-aware scheduler. Also, by increasing the number of slaves nodes, we aim to acquire more data about the solution scalability.

Hence, this second environment presents a Scenario A with fully dedicated nodes, while in Scenarios B, C and D the first half of the nodes are shared. more exactly, each node has a total of 8 GB of memory and 8 cores but in a shared node only 2 GB and 2 cores are available for Hadoop (the remaining 3/4 of the resources are assigned to another user).

5.6 Results from shared environment experiments

The results from the TeraSort benchmark on this shared environment is presented in Table 4 and Fig. 7. Once again, we compared the scheduler behavior under four different Scenarios: A, B, C and D, as described in Sect. 5.1. Table 4 summarizes the experiments, presenting the total time used by all map tasks, the average execution time, the number of successful map tasks and the number of speculative tasks launched.

Table 4 Summary of scaled TeraSort results expressed in seconds

Scenario	A	B	C	D
Total map time (*s*)	273	2138	743	1421
Avg. map time (*s*)	55.69	322.66	51.49	85.35
# Map tasks	298	298	298	298
# Speculative tasks	2	26	1	1

Figure 7 presents the Gantt charts for each scenario, where each line represents one slave node in the cluster during the experiment. As stated before, Scenarios B, C and D have five nodes with shared resources. Hence, Hadoop only access 1 / 4 of the overall resources on these nodes (for a total of 40 GB and 40 cores on these five nodes, 30 GB and 30 cores are allocated to another user in the cluster). Each line portraits the consolidated resources by that node on a color scale, where the darker the tone, the more containers are executing simultaneously. The maximum number of containers in these experiments is 8 (present on scenario A), consequently, the darkest tone on the Gantts corresponds to 8 containers. The same way as before, the lines are segmented along the chart to indicate that a container has either finished or started at that moment.

Both Fig. 7 and Table 4 show that the general behavior observed in the previous experiment is maintained. Scenario A has the fastest execution time, followed by Scenarios C, D then B. Scenarios A and C have similar averages on Map Time, followed by Scenarios D then B.

One may wonder why a small number of containers being executed on the first 5 nodes from Scenario C while the second half has a higher load. Actually, the nodes have all resources fully used but the Gantt only shows the Map tasks to simplify the view. If we included the Reduce tasks, the average load would be the same on all these nodes.

Nevertheless, we can observe some anomalies on Scenario D. While it was expected that the containers take a longer time to complete on the shared nodes, sometimes the second half of nodes (the dedicated ones) presented an abnormal small load in some nodes, sometimes even with zero running containers. This behavior can be explained by the way the Capacity Scheduler handles resource information. As explained in Sect. 4, our context-awareness collector updates the resource capabilities from each node to the Capacity Scheduler. More specifically, this information is stored in a global *totalResources* variable that is used by the Capacity Scheduler along with a second *usedResources* variable. If an update reduces the amount of global available resources, there is a transition period in which *usedResources* has a higher value than *totalResources*, leading to a halt on containers deployment as the scheduler believes that there are no free resources to command. Eventually the running containers finish their work and the Capacity Scheduler will be able to start new tasks, normalizing the situation.

There are several strategies to to minimize such anomaly. The simpler strategy considers that the update algorithm gradually reduces the available resources in order to prevent a halt on the containers scheduling. Indeed, the reduction pace according to the average execution time of containers, instead of drastically modifying the available resources. The best solution, however, requires a change on the Capacity Scheduler algorithm to allow distinct lists of *usedResources* and *totalResources* by node. Using these detailed lists, the scheduler knowns when to stop launching a container in an overloaded node while keeping the deployment on other nodes.

6 Discussions

Results presented in Sect. 5 demonstrate that, by observing context information during application execution, it is possible to improve job scheduling on Hadoop framework. Indeed, the experiments prove that by tracking the changes in the available resources (core CPUs and memory), we could keep Hadoop scheduler aware of the current status of these resources. Hadoop scheduler could then base its decisions on more accurate information, corresponding to the real execution context and not to a standard one extracted from configuration files. This allowed us to obtain important performance gains, about 40 % in the TeraSort experiment, for instance. This ability to observe context information is particularly important when executing on heterogeneous or non-dedicated clusters, in which resources may fluctuate during the application execution. Offering accurate information about the environment allows Hadoop scheduler to adapt itself to the current environment even if the execution conditions change after the job starts.

These results, although positive, have been based on a limited set of context information, the number of available cores and memory. These criteria correspond to those traditionally used by scheduling, and in this work we focus on changing only the parameter feeding of the scheduler, keeping unchanged the scheduling policy itself. This process allows the assessment of the impact of real-time context information on the performance. Even though, other context information can be acquired thanks to our context collector and could be used for scheduling purposes. For instance, information such as network bandwidth or latency, available storage and average CPU load could be used.

For instance, researches on Cloud Computing (Maurer et al. 2012; Assuncao et al. 2012) propose different

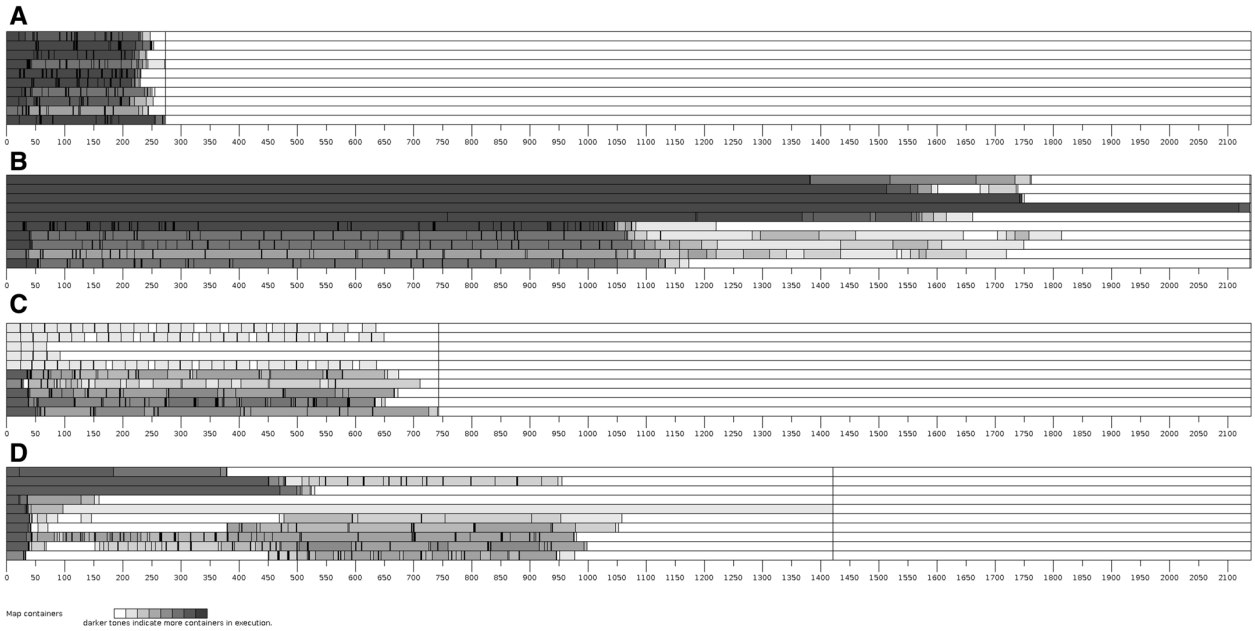


Fig. 7 Gantt charts for TeraSort on the shared environment experiment

scheduling policies to optimize resources use and to respect signed SLA (Service Level Agreement). Maurer et al. (2012) underlines the potential effects of external factors such as workload and environmental changes on VM allocation and performance. These authors (Maurer et al. 2012) consider workload volatility on resource allocation and reconfiguration in order to avoid under- or over-utilization of resources. In addition to workload information, which can be estimated using average CPU load and memory consumption, other context information can be considered. Assuncao et al. (2012) mention, for instance, the time of the day, other executing applications or the user's location. These authors suggest using context information for an adaptive job scheduling in order to rationalize resource consumption in the cloud. They assume that context-aware job scheduling is necessary in pervasive environments, in which limited mobile-based devices may delegate processing components to a cloud infrastructure. In this case, an adaptive job scheduling can be used to rationalize the consumption of cloud resources (and to avoid job violation events) based on the possible variations of the end user's local context.

These works (Maurer et al. 2012; Assuncao et al. 2012; Cavallo et al. 2015) illustrate the possible use of different context information on job scheduling. However, they also underline the necessity of an appropriate scheduling policy, specifically designed in order to explore context information. In the case of a MapReduce application in heterogeneous environments, several context parameters can be considered as relevant, and notably network conditions and data locality in addition to CPU and memory related information (e.g. available cores, CPU load, available memory, etc.). The challenge here is to figure out an adaptive scheduling policy capable of exploring this context information in an appropriate and advantageous way.

Finally, our results (cf. Sect. 5) demonstrate that it is possible to adapt Hadoop scheduler for efficient execution on heterogeneous environments. The possibility of executing MapReduce applications in such environments underlines the potential of using pervasive grids for MapReduce, and more generally, for Big Data.

Indeed, MapReduce applications, and particularly those based on the Hadoop framework, have been heavily used for Big Data to analyze large volumes of data, often unstructured, obtained from various heterogeneous sources (IoT devices, social media, etc.). These applications are mostly executed on homogeneous computing environments such as cloud computing platforms. Different reasons motivate the use of cloud computing for MapReduce applications, among them the elasticity and the cost-effectiveness. Indeed, thanks to cloud computing platforms such as Amazon Elastic MapReduce[1], it is possible to analyze an increasing volume of data, without the need of a local (and costly) infrastructure like a dedicated cluster that remains often underutilized in many organizations. Nevertheless, as underlined by Hofmann and Woods (2010), cloud computing also has several trade offs, including security, performance instability and and limited latency/

[1] http://aws.amazon.com/elasticmapreduce/

throughput network performances. Hofmann and Woods (2010), Schadt et al. (2010) also underline security and privacy issues of cloud platforms. As the indicate, several cloud providers find it hard to meet standards for auditability or to comply with legislation in the domain of security and privacy, raising the question of data sensibility.

In other words, although interesting, homogeneous cloud platform are not the only possible solution for data analysis. Several other solutions exist, such as using off the shelf GPU-based architectures (Engel et al. 2015) or pervasive grids (Steffenel and Kirsch Pinheiro 2015). In this paper, we demonstrate that, with an appropriate scheduling mechanism, pervasive grids can be used for Hadoop applications. Indeed, by introducing context-awareness capabilities, pervasive grids may represent a complementary alternative to costly dedicated clusters or to cloud infrastructures, particularly when these cannot be used due to data constraints.

7 Conclusion

In this work, we aimed at developing the ability to detect and adapt to resource availability changes on the environment of Apache Hadoop Capacity Scheduler. The improvements were implemented with a light-weight context collector and communication provided by ZooKeeper. These improvements go further than our previous work (Cassales et al. 2014) by including a continuous observation of node capabilities. The results show that this solution can positively impact the execution performance of MapReduce applications, especially in situations where the available resources drop after the beginning of the execution.

While Hadoop does not perform preemption/migration of tasks, the association of a context-aware scheduler and speculative tasks may contribute to circumvent the bottlenecks caused by the resources variability. Our future works will concentrate on modifying the scheduler algorithms, in order to consider a wider collection of context information than the current memory and cores parameters. We believe that additional parameters such as CPU speed, network speed and even battery capacity are essential factors in a pervasive grid. Finally, we intend to evaluate our context-aware scheduler under realistic situations with both MapReduce applications and pervasive grid environments composed of heterogeneous off-the-shelf volunteer computers.

Acknowledgments The authors would like to thank their partners in the PER-MARE project STIC-AmSud (2014) and acknowledge the financial support given to this research by the CAPES/MAEE/ANII STIC-AmSud collaboration program (project number 13STIC07).

References

Apache, Apache Hadoop, 2014. http://hadoop.apache.org/docs/r2.6.0/index.html. Last access: November 2014

Assuncao MD, Netto MAS, Koch F, Bianchi S (2012) Context-aware job scheduling for cloud computing environments. In: IEEE Fifth International Conference on Utility and Cloud Computing (UCC). 2012. pp 255–262. doi:10.1109/UCC.2012.33

Baldauf M, Dustdar S, Rosenberg F (2007) A survey on context-aware systems. Int J Ad Hoc Ubiquitous Comput 2(4):263–277

Cassales GW, Charao AS, Pinheiro MK, Souveyet C, Steffenel LA (2014) Bringing Context to Apache Hadoop. In: 8th International Conference on Mobile Ubiquitous Computing, Rome, Italy

Cassales GW, Charao AS, Kirsch Pinheiro M, Souveyet C, Steffenel LA (2015) Context-aware scheduling for apache hadoop over pervasive environments. Procedia Comp Sci 52:202–209. The 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015), the 5th International Conference on Sustainable Energy Information Technology (SEIT-2015). doi:10.1016/j.procs.2015.05.058. http://www.sciencedirect.com/science/article/pii/S1877050915008583

Cavallo M, Cusma L, Modica GD, Polito C, Tomarchio O (2015) A scheduling strategy to run Hadoop jobs on geodistributed data. In: 3rd Workshop on CLoud for IoT (CLIoT 2015), in conjunction with the European Conference on Service-Oriented and Cloud Computing (ESOCC 2015)

Chen Q, Zhang D, Guo M, Deng Q, Guo S (2010) SAMR: a self-adaptive MapReduce scheduling algorithm in heterogeneous environment, In: Proceedings of the 2010 10th IEEE International Conference on Computer and Information Technology. CIT '10 (IEEE Computer Society, Washington, DC, USA, 2010), pp 2736–2743 (978-0-7695-4108-2)

Dean J, Ghemawat S (2008) Mapreduce: simplified data processing on large clusters. Commun ACM 51(1):107–113

Engel T, Charo A, Kirsch-Pinheiro M, Steffenel LA (2015) Performance improvement of data mining in weka through multi-core and gpu acceleration: opportunities and pitfalls. J Ambient Intel Humaniz Comput 6(4):377–390. doi:10.1007/s12652-015-0292-9

Grid'5000, Grid 5000, 2013. https://www.grid5000.fr/, Last access: July 2014

Hamilton, J.: Hadoop Wins TeraSort, 2008. http://perspectives.mvdirona.com/2008/07/hadoop-wins-terasort/. Last access: September 2015

Hofmann P, Woods D (2010) Cloud computing: the limits of public clouds for business applications. IEEE Internet Comput 14(6):90–93. doi:10.1109/MIC.2010.136

Huang S, Huang J, Dai J, Xie T, Huang B: The HiBench benchmark suite: Characterization of the MapReduce-based data analysis. In: 2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW), 2010, pp 41–51. doi:10.1109/ICDEW.2010.5452747

Hunt P, Konar M, Junqueira FP, Reed B, ZooKeeper: wait-free Coordination for Internet-scale Systems. In: Proceedings of the USENIX Annual Technical Conference (USENIX Association, Boston, MA, USA, 2010), pp 11. http://dl.acm.org/citation.cfm?id=1855840.1855851

Isard M, Prabhakaran V, Currey J, Wieder U, Talwar K, Goldberg A (2009) Quincy: fair scheduling for distributed computing clusters, in Proceedings of the ACM SIGOPS 22nd symposium

on Operating systems principles. SOSP '09 (ACM, New York, NY, USA, 2009), pp 261–276 (978-1-60558-752-3)

Kumar KA, Konishetty VK, Voruganti K, Rao GVP (2012) CASH: context aware scheduler for Hadoop. In: Proceedings of the International Conference on Advances in Computing, Communications and Informatics. ICACCI '12, New York, NY, USA, 2012, pp 52–61 (978-1-4503-1196-0)

Li J, Wang Q, Jayasinghe D, Park J, Zhu T, Pu C (2013) Performance overhead among three hypervisors: an experimental study using Hadoop benchmarks. In: 2013 IEEE International Congress on Big Data (BigData Congress) 2013, pp 9–16. 2013, doi:10.1109/BigData.Congress..11

Maamar Z, Benslimane D, Narendra NC (2006) What can context do for web services? Commun ACM 49(12):98–103

Marozzo F, Talia D, Trunfio P (2012) P2p-mapreduce: parallel data processing in dynamic cloud environments. J Comput Syst Sci 78(5):1382–1402

Maurer M, Brandic I, Sakellariou R (2012) Self-adaptive and resource-efficient SLA enactment for cloud computing infrastructures. In: 2012 IEEE 5th International Conference on cloud computing (CLOUD), 2012, pp 368–375. doi:10.1109/CLOUD.2012.55

Najar S, Kirsch M, Pinheiro C (2015) Souveyet, service discovery and prediction on pervasive information system. J Ambient Intell Human Comp 6(4):407–423. doi:10.1007/s12652-015-0288-5

Nascimento AP, Boeres C, Rebello VEF (2008) Dynamic self-scheduling for parallel applications with task dependencies. In: Proceedings of the 6th International Workshop on MGC. MGC '08, New York, NY, USA, 2008, pp 1–116 (978-1-60558-365-5)

Oracle, Overview of Java SE Monitoring and Management, 2014. http://docs.oracle.com/javase/7/docs/technotes/guides/management/overview.html, Last access: July 2014

Parashar M, Pierson JM (2010) Pervasive grids: challenges and opportunities. In: Li K, Hsu C, Yang L, Dongarra J, Zima H (eds) Handbook of Research on Scalable Computing Technologies. (IGI Global, 2010), pp 14–30. doi:10.4018/978-1-60566-661-7.ch002 (978–160566661-7)

Ramakrishnan A, Preuveneers D, Berbers Y (2014) Enabling self-learning in dynamic and open IoT environments. In: Shakshuki E, Yasar A (eds) The 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014), the 4th International Conference on Sustainable Energy Information Technology (SEIT-2014), vol. 32, 2014, pp 207–214. doi:10.1016/j.procs.2014.05.416

Rasooli A, Down DG (2012) Coshh: a classification and optimization based scheduler for heterogeneous hadoop systems. In: Proceedings of the 2012 SC Companion: High Performance Computing, Networking Storage and Analysis. SCC '12 (IEEE Computer Society, Washington, DC, USA, 2012), pp. 1284–1291 (978-0-7695-4956-9)

Sandholm T, Lai K (2010) Dynamic Proportional Share Scheduling in Hadoop. In: Proceedings of the 15th International Conference on Job Scheduling Strategies for Parallel Processing. JSSPP'10, Berlin, Heidelberg, 2010, pp 110–131. (3–642-16504-4, 978-3-642-16504-7)

Schadt EE, Linderman MD, Sorenson J, Lee L, Nolan GP (2010) Computational solutions to large-scale data management and analysis. Nat Rev Genet 11(9):647–657. doi:10.1038/nrg2857

Steffenel LA, Kirsch Pinheiro M (2015) Leveraging data intensive applications on a pervasive computing platform: The case of mapreduce. Procedia Comp Sci 52:1034–1039 (2015). The 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015), the 5th International Conference on Sustainable Energy Information Technology (SEIT-2015). doi:10.1016/j.procs.2015.05.102. http://www.sciencedirect.com/science/article/pii/S1877050915009023

Steffenel LA, Flauzac O, Charão AS, Barcelos PP, Stein B, Nesmachnow S, Kirsch Pinheiro M, Diaz D (2013) PER-MARE: adaptive deployment of MapReduce over pervasive grids. In: Proceedings of the 2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing. 3PGCIC '13 (IEEE Computer Society, Washington, DC, USA, 2013), pp 17–24 (978-0-7695-5094-7)

STIC-AmSud, PER-MARE project, 2014. http://cosy.univ-reims.fr/PER-MARE, Last access: July 2014

Tian C, Zhou H, He Y, Zha L (2009) A dynamic MapReduce scheduler for heterogeneous workloads. In: Proceedings of the 2009 Eighth International Conference on Grid and Cooperative Computing. GCC '09 (IEEE Computer Society, Washington, DC, USA, 2009), pp 218–224 (978-0-7695-3766-5)

Xie J, Ruan X, Ding Z, Tian Y, Majors J, Manzanares A, Yin S, Qin X (2010) Improving MapReduce performance through data placement in heterogeneous Hadoop clusters, in Parallel and Distributed Processing, Workshops and Phd Forum (IPDPSW)

Zaharia M, Konwinski A, Joseph AD, Katz R, Stoica I (2008) Improving MapReduce performance in heterogeneous environments, in Proceedings of the 8th USENIX conference on Operating systems design and implementation. OSDI'08 (USENIX Association, Berkeley, CA, USA, 2008), pp 29–42

Annex V

Paper

Steffenel, L. & Kirsch-Pinheiro, M. “CloudFIT, a PaaS platform for IoT applications over Pervasive Networks”, In: Celesti A., Leitner P. (eds). *3rd Workshop on CLoud for IoT (CLIoT 2015)*. Advances in Service-Oriented and Cloud Computing (ESOCC 2015). Communications in Computer and Information Science, vol. 567, **2015**, 20-32.

CloudFIT, a PaaS Platform for IoT Applications over Pervasive Networks

Luiz Angelo Steffenel[1(✉)] and Manuele Kirch Pinheiro[2]

[1] CReSTIC Laboratory, SysCom Team,
Université de Reims Champagne-Ardenne, Reims, France
luiz-angelo.steffenel@univ-reims.fr
[2] Centre de Recherche en Informatique,
Université Paris 1 Panthéon-Sorbonne, Paris, France
manuele.kirsch-pinheiro@univ-paris1.fr

Abstract. IoT applications are the next important step towards the establishment of mobiquitous systems, but at the same time these environments raise important challenges when considering data distribution and processing. While most IoT applications today rely on clouds as back-end, critical applications that require fast response or enhanced privacy levels may require proximity services specially tailored to these needs. The deployment of private cloud services on top of pervasive grids represent an interesting alternative to traditional cloud infrastructures. In this work we present CloudFIT, a PaaS middleware that allows the creation of private clouds over pervasive environments. Using a Map-Reduce application as example, we show how CloudFIT provides both storage and data aggregation/analysis capabilities at the service of IoT networks.

1 Introduction

Today, cloud computing is a widespread paradigm that relies on the externalization of services to a distant platform with elastic computing capabilities. Unsurprisingly, Big Data analytics profits from the computing capabilities from the cloud, making it the predilection platform for information extraction and analysis.

The emergence of Internet of Things (IoT) has naturally attired the attention of developers and companies, which mostly rely on cloud services to interconnect devices and gather information. Indeed, platforms like Carriots[1] or ThinkSpeak[2] now propose PaaS APIs to collect information, visualize and control IoT devices.

Contrarily to the case of Wireless Sensor Networks (WSNs), however, IoT has a much more complex data transfer pattern that is not always tailored for a cloud. While data from WSNs naturally flows from the sensors to a "sink" repository that can gather information and handle it to the analytics software,

[1] https://www.carriots.com/.
[2] https://thingspeak.com/.

A. Celesti and P. Leitner (Eds.): ESOCC 2015 Workshops, CCIS 567, pp. 20–32, 2016.
DOI: 10.1007/978-3-319-33313-7_2

IoT devices have M2M (Machine-to-Machine) capabilities beyond simple raw data transmission, as they are also information consumers and even actuators to the real environment.

Simply relying on a distant cloud infrastructure for data storage, processing and control imposes a non-negligible latency, a complete dependency on wide-area communications and the transmission of potentially sensible data across the network. From this point of view, it is clear that not all IoT applications would benefit from an external data handling.

Deploying a privative PaaS cloud for IoT is an interesting alternative to the complete externalization, as it ensures fast reaction and privacy levels tailored to the specific needs of an enterprise or application. Indeed, the omnipresence of IoT devices often raises questions about the dissemination of sensitive data, a problem that public cloud systems can minimize through the use of heavy layers of cryptography and anonymization, but never solve.

In this paper we present CloudFIT, a distributed computing middleware designed for pervasive environments that offers IoT applications both storage, data aggregation and analysis capabilities. In addition, CloudFIT does not require a dedicated infrastructure as a CloudFIT "grid" can be deployed over existing resources on the enterprise (desktop PCs, servers, etc.) and perform both the data aggregation, filtering and analysis required by IoT devices.

After describing CloudFIT, we illustrate its operation through the deployment of a data intensive application over a cluster. We deploy a MapReduce application over CloudFIT, and compare its performance against the well-known Hadoop middleware[3], a Big Data platform specially designed for dedicated clusters.

The paper is structured as follows: Section 2 discusses the challenges for the IoT applications and the reasons why a traditional cloud services is not always recommended. Instead, we emphasize alternatives for cloud computing that ensure both efficiency and data privacy. Section 3 focuses on the case of data-intensive problems and discusses the main challenges for its deployment over pervasive grids, analyzing some related works. Section 4 presents the architecture of CloudFIT and its characteristics related to fault tolerance and volatility support. This session also discuss how to interface IoT devices and applications with CloudFIT. Section 5 introduces our implementation of a MapReduce application over CloudFIT, discussing both implementation issues and performance evaluations. Finally, Sect. 6 concludes this paper and sets the lines of our next development efforts.

2 Cloud Services and IoT

2.1 Private Clouds, Cloudlets and the IoT

When the cloud computing paradigm started, we observed the development of middlewares and tools for the establishment of private and mixed cloud

[3] http://hadoop.apache.org/.

infrastructures. Most of these tools, like Eucalyptus [18], Nimbus [12] or OpenNebula [17], are designed to provide IaaS on top of dedicated resources like clusters or private data-centers. While extremely powerful, the deployment of these environments is complex and requires dedicated resources, which minimizes their advantage against public cloud infrastructures like Amazon EC2.

Establishing on-demand cloud services on top of existing resources is also alternative to the complete externalization of services in a cloud. For example, [22] explore the limitations of mobile devices and the inaptitude of current solutions to externalize mobile services through the use of Cloudlets, i.e., virtual machines deployed on-demand in the vicinity of the demanding devices. Using cloudlets deployed as Wi-Fi hotspots in coffee shops, libraries, etc., the authors of [22] suggest a simple way to offer enough computing power to perform complex computations (services) all while limiting the service latency. Please note that these cloudlets do not work as a single entity/platform but instead act as detached handlers for specific demands.

Proximity cloud services can also be used to perform an initial processing on the data. For instance, [20] presents a platform where context information is collected, filtered and analyzed on several layers. This way, basic context actions may be decided/performed in a close area range, while a much deep analysis of the context information may be performed by external servers. This layered analysis can also be used to ensure privacy properties, for example by anonymizing the data that will be used to the global context analysis. As context my represent multiple and heterogeneous kind of information, this approach can also be implemented to general Big Data analytics on sensor data or access logs, for example.

Another usage for private clouds relates to the reinforcement of the security of a network [11]. In a mobile network (as well as in an IoT pervasive network), devices cannot rely in a single security device in the entrance of the network because multimodal connections may be established with outside devices via Wi-Fi, 3G, Bluetooth, etc. If nowadays similar procedures can be implemented through the use of 802.1x authentication or VPNs, their configuration complexity requires a high technical knowledge. A better alternative relies on a mutual monitoring system sharing information is created around a confidence zone (community). Joining a confidence zone is only possible if the device pass some control checks and, similarly, devices that become "dangerous" due to a virus or a Trojan can be blocked and removed from the community.

We consider that deploying cloud services for IoT over pervasive networks is a natural approach, as the heterogeneity and the dynamicity of the devices impose a frequent adaptation on both network interconnections and computing requirements.

2.2 Cloud Services over Pervasive Grids

Pervasive grids can be defined as large-scale infrastructures with specific characteristics in terms of volatility, reliability, connectivity, security, etc. According

to [19], pervasive grids represent the extreme generalization of the grid concept, seamlessly integrating pervasive sensing/actuating instruments and devices together with classical high performance systems. In the general case, pervasive grids rely on volatile resources that may appear and disappear from the grid, according their availability. Indeed, mobile devices should be able to come into the environment in a natural way as their owner moves [6]. Also, devices from different natures, from the desktop and laptop PCs until the last generation tablets, should be integrated in seamlessly way. These environments are therefore characterized by three main requirements:

- The volatility of its components, whose participation is a matter of opportunity and availability;
- The heterogeneity of these components, whose capabilities may vary on different aspects (platform, OS, memory and storage capacity, network connection, etc.);
- The dynamic management of available resources, since the internal status of these devices may vary during their participation into the grid environment.

Such dynamic nature of pervasive grids represents an important challenge for executing data intensive applications. Context-awareness and nodes volatility become key aspects for successfully executing such applications over pervasive grids, but also for the handling and transmission of voluminous datasets.

Our approach to implement cloud-like services over pervasive networks relies on the use of an overlay network provided by a P2P system. In this approach, the P2P overlay provides all communication and fault tolerance properties required for the operation on a pervasive network, as well as some additional services like DHT storage that can help implementing additional services.

Indeed, if P2P systems are widely known for their use on storage and sharing applications, they can also be used as platforms for coordination and distribution of computing tasks. Solutions like CONFIIT [10], DIET [3] have demonstrated the interest of P2P to support computing problems in distributed and heterogeneous environments.

3 Data-Intensive Applications on Pervasive Grids

In spite of a wide tradition on distributed computing projects, most pervasive grid middlewares have focused on computing-intensive parallel applications with few I/O and loose dependencies between the tasks. Enabling these environments to support data-intense applications is still a challenge, both in performance and reliability. We believe that MapReduce is an interesting paradigm for data-intensive applications on pervasive grids as it presents a simple task distribution mechanism, easily implemented on a pervasive environment, but also a challenging data distribution pattern. Enabling MapReduce on pervasive grids raises many research issues, which we can decompose in two subtopics: data distribution and data processing.

There are two approaches to distribute large volume of data to large number of distributed nodes. The first approach relies on P2P protocols where peers collaboratively participate to the distribution of the data by exchanging file chunks [7,15,25]. The second approach is to use a content delivery service where files are distributed to a network of volunteers [13,16].

Concerning data processing on pervasive grids, some authors have tried to improve the processing capabilities of Hadoop to take into account the volatility of the nodes. Indeed, Zaharia et al. [26] Chen et al. [5] or Ahmad et al. [1] deals with heterogeneity of the supporting infrastructure, proposing different scheduling algorithms that can improve Hadoop response time. Lin et al. [14] explore the limitations of Hadoop over volatile, non-dedicated resources. They propose the use of a hybrid architecture where a small set of reliable nodes are used to provide resources to volatile nodes.

Due to the simplicity of its processing model (map and reduce phases), data processing can be easily adapted to a given distributed middleware, which can coordinate tasks through different techniques (centralized task server, work-stealing/bag of tasks, speculative execution, etc.). Nevertheless, good performances can only be achieved through the minimization of data transfers over the network, which is one of the key aspects of Hadoop HDFS filesystem. Only few initiatives associate data-intense computing with large-scale distributed storage on volatile resources. In [4], the authors present an architecture following the super-peer approach where the super-peers serve as cache data server, handle jobs submissions and coordinate execution of parallel computations.

4 CloudFIT

In this work we present our efforts to enable MapReduce applications over the P2P distributed computing middleware CloudFIT [23]. The CloudFIT framework (Fig. 1) is structured around collaborative nodes connected over an overlay network. CloudFIT was designed to be independent of the underlying overlay, and the current version supports both Pastry [21] and TomP2P overlay networks [2]. Pastry is one of the most known P2P overlays and is widely employed in distributed computing environments. TomP2P is a more recent P2P library, enjoying an active development community.

An application for CloudFIT must provide a java class that implements two basic API methods: how many tasks to solve (`setNumberOfBlocks()`) and how to compute an individual task (`executeBlock(number, required[])`). When executing, each node owns the different parameters of the current computations (a list of tasks and associated results) and is able to locally decide which tasks still need to be computed and can carry the work autonomously if no other node can be contacted. Access to the storage is also provided through the API, if required. The status of completed tasks (optionally including the partial results from these tasks) are distributed among the nodes, contributing therefore to the coordination of the computing tasks and form a global view of the calculus.

The basic scheduling mechanism simply randomly rearranges the list of tasks at each node, which helps the computation of tasks in parallel without requiring

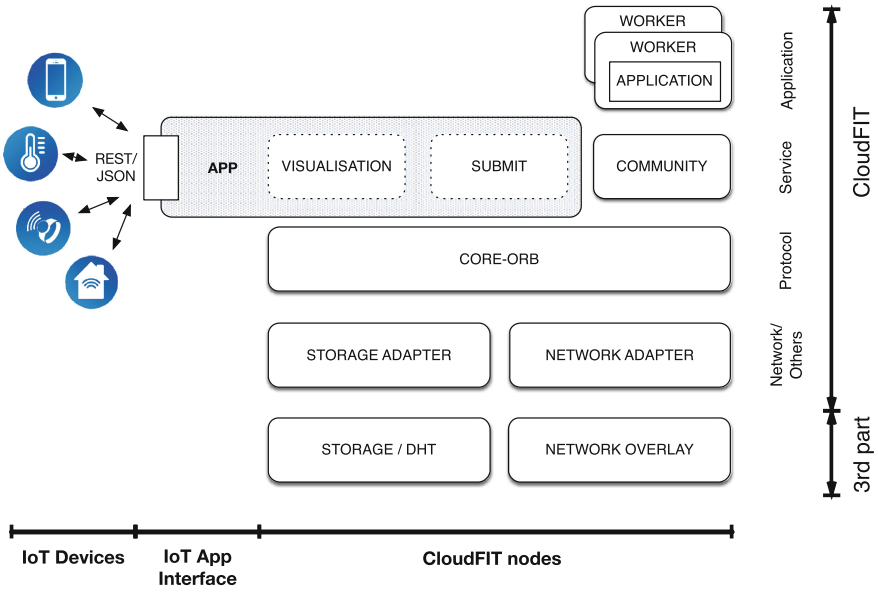


Fig. 1. CloudFIT architecture stack

additional communication between nodes. This simple scheduler mechanism was designed to allow idle processes to speculatively execute incomplete tasks, reducing the "tail effect" when a task is computed by a slow node. The scheduling mechanism supports task dependencies (allowing the composition of DAGs) and can be also be improved through the use of a context module [24] that provides additional information about the nodes capacities.

Finally, fault tolerance is ensured both by the overlay (network connections, etc.) and by the computing platform. Indeed, as long as a task is not completed, other nodes on the grid may pick it up for execution. In this way, when a node fails or leaves the grid, other nodes may recover tasks originally started by the crashed node. Inversely, when a node joins the CloudFIT community, it receives an update about the tasks current status and the working data, allowing it to start working on available (incomplete) tasks.

4.1 CloudFIT Services for IoT Devices and Applications

As previously stated, CloudFIT provides a pervasive PaaS for IoT applications. While we believe that CloudFIT can be deployed directly over IoT devices running Android (with the TomP2P overlay) or Linux on Raspberry Pi, the heterogeneity and limited resources of these devices make this approach very unreliable. Indeed, a node integrating the CloudFIT network must perform all the routing, storage and computing tasks as the others, and this can be both overloading and inefficient (please see Sect. 5.5).

A better approach, instead, is to use CloudFIT as a computing backend for IoT devices and applications. This mixed architecture, as illustrated in the left side of Fig. 1, allows an IoT application connected to CloudFIT network to act

as an interface to gather data and launch computing tasks according to the application needs.

While the development of an interface for IoT devices can be provided through *REST/json* calls or even a direct a connection to the devices via Bluetooth or Wi-Fi, it is outside the scope of this paper. Instead, the next sections illustrate the deployment of a MapReduce application over CloudFIT. This is one of several computing intensive tasks that can be performed on CloudFIT to support IoT applications.

5 MapReduce over CloudFIT

5.1 MapReduce

MapReduce [8] is a parallel programming paradigm successfully used by large Internet service providers to perform computations on massive amounts of data. The key strength of the MapReduce model is its inherently high degree of parallelism that should enable processing of petabytes of data in a couple of hours on large clusters.

Computations on MapReduce deal with pairs of key-values (k, V), and a MapReduce algorithm (a job) follows a two-step procedure:

1. map: from a set of key/value pairs from the input, the map function generates a set of intermediate pairs $(k_1; V_1) \rightarrow \{(k_2; V_2)\}$;
2. reduce: from the set of intermediate pairs, the reduce function merges all intermediate values associated with the same intermediate key, so that $(k_2; \{V_2\}) \rightarrow \{(k_3; V_3)\}$.

When implemented on a distributed system, the intermediate pairs for a given key k_2 may be scattered among several nodes. The implementation must therefore gather all pairs for each key k_2 so that the reduce function can merge them into the final result. Additional features that may be granted by the MapReduce implementation include the splitting of the input data among the nodes, the scheduling of the jobs' component tasks, and the recovery of tasks hold by failed nodes.

Hadoop, one of the most popular implementations of MapReduce, provides these services through a dual layered architecture where tasks scheduling and monitoring are accomplished through a master-slave platform, while the data management is accomplished by a second master-slave platform on top of the hierarchical HDFS file-system. Such master-slave architecture makes Hadoop not suitable for Pervasive Grids.

5.2 Map, Reduce and Task Dependencies

In order to implement a MapReduce application under the FIIT model, tasks inside a Map or Reduce job must be independent, all while preserving a causal relation between Map and Reduce. Therefore, several tasks are launched during

the Map phase, producing a set of (k_i, V_i) pairs. Each task is assigned to a single file/data block and therefore may execute independently from the other tasks in the same phase. Once completed, the results from each task can be broadcasted to all computing nodes and, by consequence, each node contains a copy of the entire set of (k_i, V_i) pairs at the end of the Map phase. At the end of the first step, a Reduce job is launched using as input parameter the results from the map phase.

In our prototype, the number of Map and Reduce tasks was defined to roughly mimic the behavior of Hadoop, which tries to guess the required number of Map and Reduce processes. For instance, we set the number of Map tasks to correspond to the number of input files, and the number of Reduce tasks depends on the size of the dataset and the transitive nature of the data. Please note that CloudFIT may optionally perform a result aggregation after each job completion, just like Hadoop *combiners*.

Because Hadoop relies on specific classes to handle data, we tried to use the same ones in CloudFIT implementation as a way to keep compatibility with the Hadoop API. However, some of these classes were too dependent on inner elements of Hadoop, forcing us to develop our own equivalents, at least for the moment (further works shall reinforce the compatibility with Hadoop API). For instance, we had to substitute the OutputCollector class with our own MultiMap class, while the rest of the application remains compatible with both Hadoop and CloudFIT.

5.3 Data Management, Storage and Reliability

As stated before, CloudFIT was designed to broadcast the status about completed tasks to all computing nodes, and this status may include the tasks' results. By including the results, CloudFIT ensures $n - resiliency$ as all nodes will have a copy of the data.

This resiliency behavior was mainly designed for computing intensive tasks that produce a small amount of data as result. On data-intensive applications, however, $n - resiliency$ may be prohibitive as not only all nodes need to hold a copy of all task's data, but also because broadcasting several megabytes/gigabytes over the network is a major performance issue.

In our efforts to implement MapReduce over CloudFIT we chose a different approach to ensure the scalability of the network all while preserving good reliability levels. Hence, we rely on the DHT to perform the storage of tasks results as {*task_key, task_result*} tuples, while the task status messages broadcast the keys from each task. As both PAST and TomP2P DHT implement data replication among the nodes with a predefined replication factor k, we can ensure minimal fault tolerance levels all while improving the storage performance.

5.4 Performance Evaluation Against Hadoop

In order to evaluate the performance of MapReduce over CloudFIT we implemented the traditional WordCount application and compared it against WordCount 1.0 application from Hadoop tutorial.

To make this first evaluation fair, we conducted this first experiment over 8 machines from the ROMEO Computing Center[4]. ROMEO cluster nodes are composed by bi-Intel Xeon E5-2650 2.6 GHz (Ivy Bridge) 8 cores and 32 GB of memory, interconnected by an Infiniband QDR network at 40 Gbps. Hadoop YARN nodes run with default parameters (number of *vcores* = 8, available memory = 8 GB), parameters that we reproduced on CloudFIT for fairness (i.e., by limiting the number of parallel tasks by node and setting the maximum java VM memory).

Two different versions of CloudFIT were tested, one using the FreePastry overlay with the PAST DHT at the storage layer, and the second one with the TomP2P overlay and its Kademlia-based DHT.

The experiments considered the overall execution time (map + reduce phases) of both CloudFIT and Hadoop implementations when varying the total amount of data (512 MB to 2 GB). The data was obtained from a corpus of textbooks from the Gutenberg Project and split in blocks of 64 MB to reproduce the size of an HDFS data block. The results obtained when running on an 8 nodes cluster are presented on Fig. 2, which shows the average of 10 executions for each data size.

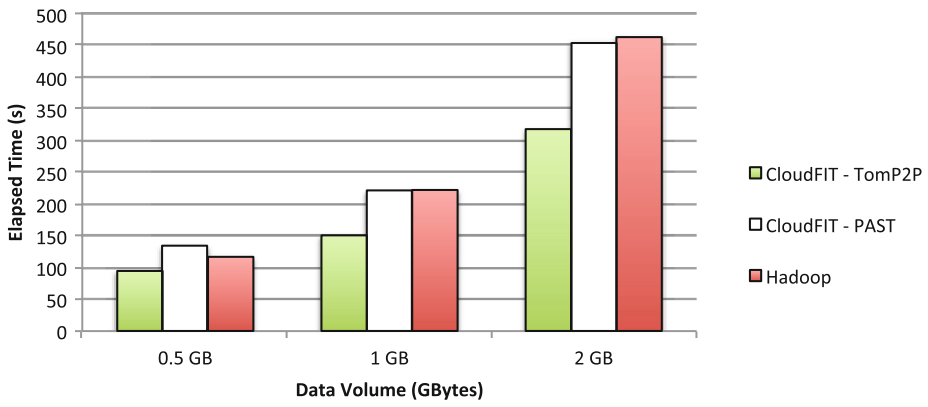


Fig. 2. WordCount MapReduce performance on 8 nodes

At first glance, we observe that the CloudFIT/TomP2P implementation easily outperforms both CloudFIT/PAST and Hadoop, which have aproximately the same performance. A deeper analysis of the CloudFIT/PAST implementation show that the PAST DHT experimented a performance bottleneck related to the use of mutable objects. Indeed, mutable objects are useful to gather $(k; V)$ pairs from different tasks but they force a non-negligible overhead at the DHT

[4] https://romeo.univ-reims.fr.

controller, which must scan the data for changes and trigger replication updates. One solution to improve the PAST performance is to rely on immutable objects that do not suffer from this problem, but this requires the usage of alternative data structures to reproduce the $(k; V)$ associations from MapReduce.

This is an encouraging result as it demonstrates the interest of CloudFIT as a platform for Big Data applications. Depending on the storage layer, we can provide good performance levels without sacrificing the platform flexibility. In addition, the modular organization of CloudFIT allows connecting other storage supports like BitDew [9], external databases, URLs, etc., according to the application requirements.

5.5 Performance Evaluation on a Pervasive Grid

As the previous section demonstrate that CloudFIT can execute MapReduce applications as fast as Hadoop in a HPC cluster, the next step in our experiments considered the creation of a pervasive cluster on top of common desktop equipments. For instance, we executed CloudFIT/TomP2P over a network composed by three laptop computers connected through a Wi-Fi 802.11 g network. The specifications for each model are presented in Table 1. Please note that these machines were not tuned for performance and indeed CloudFIT had to share the resources with other applications like anti-virus, word processors, etc.

Table 1. Specification of the nodes on the pervasive cluster

Laptop	Processor	GHz	Cores	Threads	Memory	OS
MacBook Air	Intel® Core™ i7-4650U	1.7	2	4	8 GB	MacOS 10.10.3
HP Pavillon dv6	Intel® Core™ i5-2450M	2.5	2	4	8 GB	Windows 7
Lenovo U110	Intel® Core™2 Duo L7500	1.6	2	2	4 GB	Ubuntu Linux 15.4

Figure 3 presents the execution of the WordCount application with three different data sets, and we compare the execution time obtained on the pervasive grid with the performance obtained over 3 nodes from the ROMEO cluster. Post-execution analysis indicated that in spite of the processors type and speeds, one factor that mainly influenced the performance was the network speed. Indeed, as the MapReduce application performs several read/write operation over the DHT, the network is a major bottleneck: to write 64 MB of data on the DHT using the ROMEO cluster (equipped with an Infiniband interconnection) we need in average 2 s, while the Wi-Fi connection used on the pervasive cluster required in average 15 s. Another element that contributes to the reduced performance of the pervasive environment is the competition between faster and slower nodes: while both node types have similar chances to draw tasks to execute at the beginning, faster nodes will complete their tasks first and finally re-execute the tasks from slower nodes, wasting computing resources.

While comparing both environments is not really fair, the conclusion is that one does need a dedicated environment to extract enough computing power for

several applications. In fact, the flexibility of the pervasive cluster allows nodes to join or leave the cluster without interfering with the execution, making it a strategic tool for most organizations that cannot rely neither in a dedicated cluster neither in a distant datacenter/cloud infrastructure. Further, CloudFIT has the advantage that it can be easily run on Windows, contrarily to Hadoop, which reinforcing its ability to create pervasive clusters from the available resources.

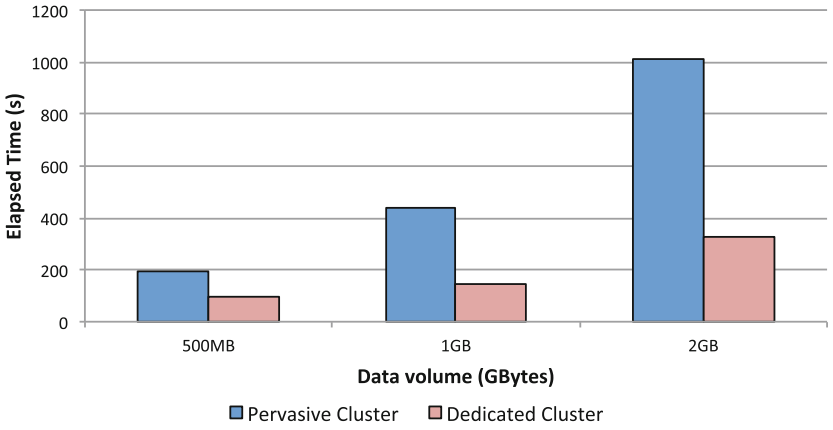


Fig. 3. WordCount MapReduce on 3 nodes: pervasive vs dedicated cluster

6 Conclusions and Future Work

IoT networks are the next important step towards the establishment of mobiquitous systems. Contrarily to Sensor Networks, IoT has a much richer M2M pattern that is not always adapted to the cloud computing paradigm. Indeed, moving data to distant platforms for filtering, analysis and decision-making is both expensive and time consuming, which not always fits the IoT applications requirements.

In this paper we present CloudFIT, a PaaS middleware that allows the creation of private clouds at the proximity of the demanding IoT devices. Using a P2P overlay, CloudFIT offers both storage and computing capabilities on top of pervasive networks.

We illustrate the usage of CloudFIT through the deployment of a MapReduce application and the comparative performance analysis with Hadoop. Indeed, we demonstrate that CloudFIT offers performance levels similar to those of Hadoop but with a better support for dynamic and heterogeneous environments.

Of course, the possibilities that CloudFIT offers to IoT are not limited to MapReduce applications. The CloudFIT API and its distributed computing model allow many other usages, as devices can use the platform as a storage support, data analysis support, intensive computing support, etc. By coordinating activities over CloudFIT, IoT devices and applications can elaborate a supply chain from data gathering to reasoning and actuation.

Acknowledgment. The authors would like to thank their partners in the PER-MARE project (http://cosy.univ-reims.fr/PER-MARE) and acknowledge the financial support given to this research by the CAPES/MAEE/ANII STIC-AmSud collaboration program (project number 13STIC07).

References

1. Ahmad, F., Chakradhar, S.T., Raghunathan, A., Vijaykumar, T.N.: Tarazu: optimizing mapreduce on heterogeneous clusters. SIGARCH Comput. Archit. News **40**(1), 61–74 (2012)
2. Bocek, T., et al.: TomP2P, a P2P-based high performance key–value pair storage library. http://tomp2p.net/
3. Caron, E., Desprez, F., Lombard, F., Nicod, J.-M., Philippe, L., Quinson, M., Suter, F.: A scalable approach to network enabled servers. In: Monien, B., Feldmann, R.L. (eds.) Euro-Par 2002. LNCS, vol. 2400, pp. 907–910. Springer, Heidelberg (2002)
4. Cesario, E., De Caria, N., Mastroianni, C., Talia, D.: Distributed data mining using a public resource computing framework. In: Desprez, F., Getov, V., Priol, T., Yahyapour, R. (eds.) Grids, P2P and Service computing, pp. 33–44, Springer (2010)
5. Chen, Q., Zhang, D., Guo, M., Deng, Q., Guo, S.: Samr: a self-adaptive mapreduce scheduling algorithm in heterogeneous environment. In: Proceedings of the 2010 10th IEEE International Conference on Computer and Information Technology, CIT 2010, pp. 2736–2743. IEEE Computer Society, Washington, D.C. (2010)
6. Coronato, A., Pietro, G.D.: MiPeG: a middleware infrastructure for pervasive grids. Future Gener. Comput. Syst. **24**(1), 17–29 (2008)
7. Costa, F., Silva, L., Fedak, G., Kelley, I.: Optimizing data distribution in desktop grid platforms. Parallel Process. Lett. **18**(3), 391–410 (2008)
8. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
9. Fedak, G., He, H., Cappello, F.: BitDew: a programmable environment for large-scale data management and distribution. In: SC 2008: Proceedings of the 2008 ACM/IEEE conference on Supercomputing, pp. 1–12. IEEE Press, Piscataway (2008)
10. Flauzac, O., Krajecki, M., Steffenel, L.: CONFIIT: a middleware for peer-to-peer computing. J. Supercomput. **53**(1), 86–102 (2010)
11. Flauzac, O., Nolot, F., Rabat, C., Steffenel, L.: Grid of security: a decentralized enforcement of the network security. In: Gupta, M., Walp, J., Sharman, R. (eds.) Threats, Countermeasures and Advances in Applied Information Security, pp. 426–443. IGI Global, April 2012
12. Keahey, K., Tsugawa, M., Matsunaga, A., Fortes, J.: Sky computing. IEEE Internet Comput. **13**(5), 43–51 (2009). http://dx.doi.org/10.1109/MIC.2009.94
13. Kelley, I., Taylor, I.: A peer-to-peer architecture for data-intensive cycle sharing. In: Proceedings of the First International Workshop on Network-Aware Data Management (NDM 2011), pp. 65–72. ACM, New York (2011)
14. Lin, H., Ma, X., Archuleta, J., Feng, W., Gardner, M., Zhang, Z.: Moon: mapreduce on opportunistic environments. In: Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC 2010), pp. 95–106 (2010)

15. Marozzo, F., Talia, D., Trunfio, P.: A peer-to-peer framework for supporting mapreduce applications in dynamic cloud environments. In: Antonopoulos, N., Gillam, L. (eds.) Cloud Computing. Computer Communications and Networks, pp. 113–125. Springer, London (2010)
16. Mastroianni, C., Cozza, P., Talia, D., Kelley, I., Taylor, I.: A scalable super-peer approach for public scientific computation. Future Gener. Comput. Syst. **25**(3), 213–223 (2009)
17. Moreno-Vozmediano, R., Montero, R.S., Llorente, I.M.: IaaS cloud architecture: from virtualized datacenters to federated cloud infrastructures. Computer **45**(12), 65–72 (2012)
18. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The eucalyptus open-source cloud-computing system. In: 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGrid 2009, Shanghai, China, 18–21 May 2009, pp. 124–131 (2009). http://doi.ieeecomputersociety.org/10.1109/CCGRID.2009.93
19. Parashar, M., Pierson, J.M.: Pervasive grids: challenges and opportunities. In: Li, K., Hsu, C., Yang, L., Dongarra, J., Zima, H. (eds.) Handbook of Research on Scalable Computing Technologies, pp. 14–30. IGI Global (2010)
20. Rottenberg, S., Leriche, S., Lecocq, C., Taconet, C.: Vers une définition d'un système réparti multi-échelle. In: UBIMOB 2012 - 8èmes Journées Francophones Mobilité et Ubiquité, pp. 178–183 (2012)
21. Rowstron, A., Druschel, P.: Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems. In: IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), pp. 329–350, November 2001
22. Satyanarayanan, M.: Mobile computing: the next decade. SIGMOBILE Mobile Comput. Commun. Rev. **15**, 2–10 (2011)
23. Steffenel, L., Flauzac, O., Charao, A.S., Barcelos, P.P., Stein, B., Nesmachnow, S., Pinheiro, M.K., Diaz, D.: PER-MARE: adaptive deployment of mapreduce over pervasive grids. In: Proceeding 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, October 2013
24. Steffenel, L., Flauzac, O., Charao, A., Barcelos, P., Stein, B.: Cassales, G., Nesmachnow, S., Rey, J., Cogorno, M., Kirsch-Pinheiro, M., Souveyet, C.: Mapreduce challenges on pervasive grids. J. Comput. Sci. **10**(11), 2194–2210 (2014)
25. Vazhkudai, S., Freeh, V., Ma, X., Strickland, J., Tammineedi, N., Scott, S.: FreeLoader: scavenging desktop storage resources for scientific data. In: Proceedings of Supercomputing (SC 2005), Seattle (2005)
26. Zaharia, M., Konwinski, A., Joseph, A.D., Katz, R., Stoica, I.: Improving mapreduce performance in heterogeneous environments. In: Proceeding of the 8th USENIX Conference on Operating Systems Design and Implementation, OSDI 2008, pp. 29–42. USENIX Association (2008)

Annex VI

Paper

Ben Rabah, N.; Kirsch Pinheiro, M.; Le Grand, B.; Jaffal, A. & Souveyet, C., "Machine Learning for a Context Mining Facility", *16th Workshop on Context and Activity Modeling and Recognition, 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2020, pp.678-684.

Machine Learning for a Context Mining Facility

Nourhène BEN RABAH, Manuele KIRSCH PINHEIRO, Bénédicte LE GRAND, Ali JAFFAL, Carine SOUVEYET
Centre de Recherche en Informatique, Université Paris 1 Panthéon Sorbonne
Paris, France
{ Nourhene.Ben-Rabah | Manuele.Kirsch-Pinheiro | Benedicte.Le-Grand | Ali.Jaffal | Carine.Souveyet }@univ-paris1.fr

***Abstract*—This paper considers generalizing context reasoning capabilities through a context mining facility offered to all Information System applications. This facility requires mining context data at the system scale, which raises several challenges for Machine Learning approaches used for such mining. Through a detailed literature review, we analyze these approaches with regard to the requirements of such a context mining facility at the Information System level, pointing to the potential and to the challenges raised by this perspective.**

***Keywords*-context mining, context data, machine learning**

I. INTRODUCTION

Observing and gathering context information from the physical environment is no longer a challenge. The development of the Internet of Things (IoT) technology and smart devices makes it now possible to easily collect multiple data about people from the applications they use and from their surrounding environments e.g. through sensors. In Information Systems (IS), these data can be exploited for decision making as well as for adaptation and automation purposes. An IS is not a simple set of applications. It is also a set of data, resources (both technical and human), and processes of a given organization. Those work together in order to help this organization fulfill its strategic goals and needs. An IS can be seen as a living organism that must evolve with the organization, whose goals and needs are constantly evolving too. Context data can contribute to the evolution of this ecosystem of applications and resources. For instance, we can easily imagine using context data to adapt business processes, by allocating tasks –or skipping some of them- according to actors' current context; such data can also help build new Key Performance Indicators for smart cities or digital twin scenarios.

Facing the potential offered by context data, Information Systems might now consider context support as a facility, i.e. as a service offered by the system itself to its applications. Viewing context provision as a facility represents an important shift from the traditional context support paradigm: instead of considering context data for particular applications with precise goals, it becomes necessary to gather as much data as possible and provide appropriate reasoning mechanisms for current but also future applications (i.e. consumers of the context facility).

This new vision represents an exciting challenge for context support. Besides obvious issues related to storage, privacy, security and legal issues, the massive availability of context data also raises some questions regarding Machine Learning (ML) approaches. Indeed, the added value of such context data relies on the capability of extracting knowledge from it, by mining such data for applications, organizations and individuals. Some features of context data, when considered at an Information System scale, may represent a challenge for ML techniques. First of all, context data is uncertain and heterogeneous by nature. Missing and potentially erroneous data from several different formats should be expected. Quality of data cannot be assured, since it depends on unpredictable environment events (e.g. device or network failure, low battery, human intervention, etc.). Besides, context data is evolutive: new sensors and context sources, with different data types and formats, may integrate the system, while others may disappear. Under such conditions, traditional ML tasks, such as selecting features, identifying relevant testing and training sets, labelling data or just training, may become complex.

Thinking of context as a facility implies collecting data not only for a specific application, but for several potential consumers, including future ones. Therefore, different ML approaches may be considered to address the various uses of such a context facility. For example, extracting indicators (KPI) and tendencies for stakeholders in business organizations will not necessarily be achieved the same way as providing context reasoning to applications for runtime adaptation. Indeed, different ML approaches are required and it is important to understand their strengths and drawbacks in view of these evolutive, uncertain and heterogeneous context data at a large scale.

In this paper, we study the opportunity of integrating ML approaches as part of a context mining facility, transforming context support into a service offered by Information Systems to their applications and users.

The remaining of this paper is organized as follows: Section II presents challenges of the proposed context mining facility, while Section III discusses the use of ML approaches on context data. In Section IV, we discuss the applicability of such approaches to this IS-level facility vision, before concluding in Section V.

II. TOWARDS CONTEXT MINING AS FACILITY

Numerous context-aware applications in the literature use ML techniques for reasoning or "mining" context data and for extracting meaningful information from these data. Examples include applications in health care [1], smart cities [2] and IoT [3] [4], just to cite a few. Most of them focus on specific applications, and demonstrate the interest of applying ML approaches to context data. For example, reasoning mechanisms may allow applications to detect or anticipate particular situations and to adapt their behavior accordingly [5] [6] [7].

Moreover, with the growing development of IoT and sensing technologies, gathering context data becomes

easier, and the potential of mining such data only starts to be foreseen, particularly for Information Systems. In such systems, mining context data may have multiple uses: adaptation of system behavior and applications, recommendation of actions, data prediction, anticipation of user's needs, and decision making (e.g. [8] [9]). With the availability of context data, we may expect a generalization of such "smart" behaviors, built upon mining solutions. According to [10], we can already observe a trend toward increasingly sophisticated systems, coined as "intelligent", "context-aware", "adaptive", "situated", etc., for which the notion of 'context' is central: they are aware of the context that they are used in, and intelligently adapt to this context at runtime.

However, generalizing such smart behavior implies generalizing context mining to the whole Information System. Making context data available only for the applications composing such system is not enough, since the cornerstone of this expected smart behavior remains the capability of reasoning (and mining) such data. Instead of collecting and mining context data application by application, we consider integrating these tasks to the Information System as a facility, i.e., as a service offered by the system to whatever application that needs it.

Considering context mining as an Information System facility implies an important shift in the way context support is currently handled, since the scale changes drastically. We no longer consider single applications, with precise goals and data, but offer a service for a full ecosystem of users (stakeholders, employees, customers, etc.) and applications supporting different business processes. When considering a given application, its developers/designers may define precisely what context data will be considered, the processing steps these data need and the most adapted ML technique to apply considering these data and the application's purposes. If we consider a service at the Information System scale, we can neither focus on a single purpose, nor predefine the set of context data to be used, since the same service is supposed to be available to many different applications. It is supposed to satisfy the needs of existing applications, considering currently available sources of context data, but also consider future sources of data and the needs of future applications and users.

Previous works on context-aware computing have considered the evolution of context sources through middleware and context models allowing new sources, data types and formats to be easily added [11] [12] [13]. However, the same does not necessarily apply to reasoning mechanisms. Indeed, as we will discuss in Section III, ML approaches usually consider specific data, which are formatted and prepared specifically for the algorithms that will be applied. Those algorithms may also vary according to the purpose of the analysis (classification, regression, clustering, etc.).

Thus, considering context mining as a facility involves overcoming such specificities in order to propose a general service, which must also cope with context data characteristics. Context-aware computing literature [11] [14] highlights multiple features of context data, summarized in Table I, which make such data particularly challenging for some ML approaches. Context data is naturally uncertain and incomplete; it may contain errors and be very dynamic; it is heterogeneous, including different formats (numeric and symbolic, structured and unstructured, etc.), types and sources; it may be observed using different frequencies or be pushed up by its sources (i.e. sensors). The sources of these data may also vary, as new sources can be integrated into the system, while others may disappear (temporarily or definitely). All this suggests new data and potentially new data formats for ML algorithms, which may be unable to handle them.

TABLE I. CONTEXT DATA MAIN FEATURES

Characteristic	Definition
Uncertainty	Context data may contain errors and imprecisions. Data quality cannot be assured.
Incompleteness	Context data may be incomplete; we cannot guarantee that 100% of possible observations have been recorded.
Heterogeneity	Context data may be gathered from multiple sources, using multiple formats even for the same data.
Dynamicity	Context data may evolve quickly; new data may arrive frequently, and data may be observed with different frequencies.

Therefore, considering context mining as a facility implies requirements that may impact the reasoning techniques at stake and notably those based on ML approaches (see Table II).

TABLE II. REQUIREMENTS FOR A CONTEXT MINING FACILITY

Requirement	Definition
R0	Guaranteeing security and privacy of context data.
R1	Supporting context data features (Table I).
R2	Supporting multiple heterogeneous context data sources.
R3	Supporting the evolution of context data sources and formats; new unexpected or non-predefined context sources and formats should be easily integrated in the system.
R4	Supporting dynamicity of context data sources; these sources may become offline unexpectedly, for small or large periods of time, they may also totally disappear, and other new sources from known and unknown formats and data types may integrate the system.
R5	Supporting online processing; the Information System should constantly remain running, and since critical applications may depend on context data, high availability is mandatory for supporting system applications.
R6	Operating with no or little human intervention; considering frequent human intervention for keeping the system running, for cleaning or preparing new context data is unfeasible considering the volume of data, the dynamicity of the data set and the need for availability.

The requirements listed in Table II consider a heterogeneous and constantly evolving data set formed by observed context data. Under such conditions, it is difficult to assume a previous knowledge about these data. Moreover, stopping the system for data preparation or preprocessing pre-treatment or data preparing may be

costly in terms of consequences for the system applications that use the service.

It is thus necessary to examine the impact of these requirements on ML approaches in order to achieve the smart behavior promoted by context-awareness at an Information System scale. In order to tackle this issue, we analyzed how ML approaches are currently used for mining context data and what their requirements are for correctly analyzing these data. Our goal is to evaluate whether our view of context mining facility is possible and which challenges should be tackled for applying ML approaches to context data at this scale.

III. USING MACHINE LEARNING FOR CONTEXT MINING

As explained previously, the vision we propose of context as an Information System facility requires the collaboration of several research areas, among which context-aware computing and ML. We have therefore considered various research communities in our literature review. We have first studied the articles published at the CoMoRea workshop, as it is dedicated to context modeling and reasoning. The papers we have selected from this conference reflect very interesting contributions. We have however noticed that many of them are application-specific (and therefore context-specific), whereas we seek more general solutions. Moreover, the justification of choice of the algorithm used for the reasoning part is not always very detailed in the papers we studied.

We have thus widened the spectrum of our literature review in order to include the ML and IoT communities. Instead of focusing on specific conferences, we have performed keywords-based queries like "context prediction" and "context awareness", with a precision on the research area, like "ML", "data mining", or "IoT". We have discarded papers published before 2013 as we wanted to focus on recent works. These keyword-based queries returned more than 200 research papers. We read their abstracts and selected those which went further than merely "provide context data" but also included some reasoning or mining mechanisms. Among the remaining articles, we focused on the approaches that seemed appropriate in terms of scalability and discarded some rule-based or ontology-based solutions, focusing on works using ML approaches only.

Indeed, since our main goal here is to better understand challenges of ML for a context facility use, we have focused only on these approaches. We are aware that ontology and ML can be combined. Several papers propose to facilitate ontology generation thanks to ML approaches [15] [16], while a few others try to use ontologies as knowledge representations to support ML approaches [17]. However, they still face some challenges (that we underline later in this section), in addition to some extra issues such as ontology learning, or temporal reasoning, as pointed by [16]. Thus, in order to focus on ML issues, we decided to focus our analysis on ML approaches, ignoring those that combine them with other reasoning techniques.

Following this methodology, we finally selected 30 papers that we analyzed in depth. These papers have been published in various communities, covering a large spectrum of expertise. About 20% come from the context modeling and reasoning community; 23% from sensors and mobile networks area; 20% have been published in ML journals of conferences. Finally, 20% come from generalist computer science sources and about 10% from areas outside the computer science fields, such as biomedicine or social sciences. As mentioned earlier, all papers have been published since 2013; half of them have even been published since 2016.

Whatever the adopted paradigm (supervised, unsupervised, semi-supervised or reinforcement learning [18] [19]), ML is about selecting an algorithm and training it on some data. The effectiveness of a given method depends on various factors, such as the quality of the training data, the chosen algorithm and its hyperparameters. Low quality data may compromise the success of the most powerful ML algorithms [20] [21]. As mentioned earlier, raw data obtained from heterogeneous sensors may be noisy, scattered and even incomplete. This may lead to various difficulties such as increased processing time, higher model complexity and overfitting.

In order to address these problems, many studies in context recognition include a pre-processing step. For example, in [7], the authors proposed a general context prediction structure based on a pre-processing phase and a context prediction phase. In [18], the authors presented a framework based on four steps: data processing, feature set generation, model selection and model combination for predicting the consumption of air conditioning in residential buildings. We detail frequently-used pre-processing tasks below.

Data cleaning techniques use one or more filters to identify noisy data and to correct or delete them [22] [23] [24]. They can also process missing values and detect outliers [18] [25].

Data transformation methods modify data representation to make them suitable as model inputs. They involve digitalization and normalization: digitalization encodes qualitative data into numerical data. Indeed, some algorithms can work directly with qualitative data, such as K-Nearest Neighbors (KNN), Naive Bayes (NB), Decision Tree (DT), or Random Forest (RF), However, many others require numerical input and output variables to operate properly. Normalization modifies the values of numerical data into a common scale. In deep neural networks, if numerical data do not have similar ranges of values then they will have a negative influence on gradient descent optimization methods, with a lower learning rate [25] [26].

Feature extraction and selection techniques identify relevant information and help remove as much irrelevant and redundant features as possible from raw data. If there are not enough informative features, the model will not be able to accomplish its task. If there are too many features or irrelevant ones, then the model will be more resource consuming and harder to train; practitioners agree that most of the time spent in building a ML pipeline is dedicated to feature engineering [27].

Data augmentation methods create additional training samples since some ML algorithms such as deep neural

networks need a huge amount of data to learn effectively [28]. However, collecting such training data is often expensive and laborious.

Unbalanced data processing methods are used to address the problem of class imbalance (i.e. when there is a disproportionate ratio of instances in each class) [29]. This is often the case with real world data, and the models learned from them generally have a good accuracy on the majority class but perform poorly on other classes.

These various pre-processing techniques have been used in several research studies for context modelling and recognition. In Table III, we compare the solutions proposed in the papers of our literature review according to the data processing mechanisms employed by each work, as well as the ML approach that has been used. We pay special attention to the following pre-processing tasks: cleaning (*c1*), transformation (*c2*), feature extraction and selection (*c3*), data augmentation (*c4*), and unbalanced data processing (*c5*). Regarding the ML approach, we indicate the algorithm(s) (*c7*) that have been used. We also report whether or not the authors mentioned hyperparameters tuning (*c6*). A hyperparameter is a parameter of the ML algorithm and not of the model. Setting hyperparameters can improve the performance of algorithms. Only 67 % of the 30 selected papers appear in Table III as we discarded the survey papers, which did not describe original experimentations.

TABLE III. *COMPARATIVE OVERVIEW OF CONTEXT RECOGNITION APPROACHES BASED ON MACHINE LEARNING*

Ref.	Data criteria					Reasoning criteria	
	c1	*c2*	*c3*	*c4*	*c5*	*c6*	*c7*
[23]	✓	×	✓	×	×	✓	SVM
[30]	×	×	✓	×	×	×	KNN, Gaussian Naive Bayes, DT, RF, RMD
[31]	×	✓	✓	×	×	×	NB, DT, RF, SVM, KNN, Adaptive Boosting, LR, ANN
[32]	✓	✓	✓	×	×	✓	MLP
[33]	✓	✓	✓	×	×	×	NB, ANN, Bayesian Network
[34]	×	×	✓	×	×	✓	KNN
[35]	×	×	✓	×	×	×	KNN, NB, RF
[36]	✓	×	✓	×	✓	×	NB, DT, LR, Adaboost, SVM
[18]	✓	×	✓	×	×	✓	Support Vector Regression, Ensemble tree, ANN
[37]	×	×	✓	×	×	×	RF, SVM, NB, Random Tree, Bayesian Network
[38]	×	×	✓	×	×	×	DT, MLP, LR
[39]	✓	×	✓	×	×	×	MLP, SVM, LogitBoost
[40]	×	×	✓	×	×	×	RF
[22]	✓	×	✓	×	×	✓	RF
[24]	✓	×	×	×	×	✓	CNN+ symbolic model
[28]	×	✓	×	✓	×	✓	CNN
[26]	✓	✓	×	×	×	✓	CNN+ LSTM
[25]	×	×	×	✓	✓	✓	CNN
[29]	×	×	×	×	✓	×	RNN
[41]	×	×	×	×	×	✓	CNN

(×) No (✓) Yes

We note that the authors of [23] propose a multi-class Support Vector Machine (SVM) based on features extracted from both an accelerometer and a gyroscope after a pre-processing step for noise reduction with a median filter and Butterworth filter. In [30], the authors propose a multi-class classifier to determine the position of smartphones in different contexts of use. They apply KNN, Gaussian Naive Bayes, DT, RF and Random Mixture Model (RMD), and report that KNN provide better performance than the other algorithms. For data pre-processing, the approach is based on a windowing step and a feature extraction step. In [32], the authors suggest a framework based on genetic algorithm to extract relevant features from IoT raw data and a Multilayer Perceptron (MLP) to classify these data in smart industrial applications.

To obtain better performance, several studies use ensemble methods such as [18] that combines the predictions of three models resulting from Support Vector Regression, Ensemble Tree and Artificial Neural Network (ANN) algorithms for predicting the consumption of air conditioning in residential buildings. To ensure that their data meet the input requirements of the models, they perform a linear interpolation for missing and incorrect data. Then, to select the right feature set, they use a statistical measure. In [37], the authors propose an approach for detecting the current transportation mode of a user from his/her smartphone sensors data. They propose to divide the collected data into consecutive non-overlapping time sequences and to extract four features for each sequence and each sensor. Then, they combine multiple learners to improve their performance. In [38], the authors present an ensemble method that combines the predictions of three models resulting from DT, MLP, and Logistic Regression (LR) for human activity recognition. To determine the class of a new activity, they consider the predictions (i.e. classes) of the three models, and choose the class with the highest number of votes. Their results show that ensemble learning can achieve significant improvements for activity recognition when compared to what each learning algorithm can achieve individually. The same problem is also investigated in [39]; in this case, however, the authors combine the results of other classifiers such as MLP, SVM, and LogitBoost. In addition, they use a clustering method to select 18 relevant features from 24 features and they obtain a good accuracy of 91.15%. In [22], the authors introduce a multi-class classification approach based on ultra-wide band sensor measurements and RF to detect when old people fall down. The pre-processing phase includes filtering, feature extraction, stream windowing, change detection and buffering. The classifier obtains the lowest error rate by setting the number of trees at 200.

To extract relevant features without effort and improve their performance, several studies use different deep neural networks architectures. The authors in [25] [28] [41] [42] [43] propose a Convolutional Neural Network (CNN) allowing feature extraction and classification for human activity recognition. The same problem is also investigated in [26], where the authors propose a generic deep framework for activity recognition based on convolutional and Long Short-Term Memory (LSTM) recurrent units. In

[34], the authors present a learning deep features for KNN to improve the classification performance. In these works, it is not necessary to extract the hand-crafted features or to use statistical methods or frequency transformation coefficients [34], as deep features can be extracted using deep learning approaches. The first layers of networks extract features that the following layers will combine to form increasingly complex and abstract concepts.

To ensure the performance of deep learning networks, the authors of [26] pre-process sensor data to fill in the missing values by linear interpolation and to perform channel normalisation to the interval [0,1]. For [25], a normalization step is required for the raw signal extracted from the accelerometers to have a common scale. They propose to apply a mean-zero normalisation. In [28], the authors use data augmentation methods such as Gaussian noise to artificially create new training data from existing learning data. In [29], the authors present a Recurrent Neural Network (RNN) based on a windowing approach for human activity recognition. They apply a synthetic minority over sampling technique to deal with the class imbalance problem.

In the next section, we analyze lessons learned from these ML approaches face to the challenges proposed by a context mining facility.

IV. Machine Learning Approaches face to Context Mining Challenges

The comparative table we proposed above (Table III), based on our literature review, allows us to make several observations. We observe that the authors from context and from sensors/mobile networks communities are up to date with regard to the state of the art in ML solutions. Indeed, there is no striking difference in the algorithms used in those communities as compared to the ML one. However, we notice that most of these "non machine learning experts" used the algorithms without mentioning any hyperparameters tuning phase. This may suggest that they could benefit from experts help in this area to better use existing ML solutions, notably considering hyperparameters optimization. Besides, we can note that deep learning solutions (among which CNN, frequently seen in Table III) are becoming more and more popular. Indeed, one of their strengths is that they allow skipping the feature extraction and selection step. Moreover, these approaches can handle large volumes of data, which is one of the requirements for context data.

Another observation we can make is that very few works consider the variety of context data and take into account all their features. The approaches we have reviewed apply ML techniques to subsets of very specific data, such as a predefined set of sensors.

As we may observe in Table III, ML approaches illustrated by our literature review heavily rely on data pre-processing phases. The quality of these approaches depends on these phases, which may be mandatory in some cases. Focusing on precise context data allows the execution of these pre-processing phases, since data types and formats are known in advance. However, when considering the context mining facility requirements highlighted in Table II, we may note that the execution of such phases cannot be guaranteed. Since new context data sources and formats may join the system at any moment, these pre-processing phases can be put in question. On the one side, if these phases are not reconsidered, new relevant data may remain ignored and context data unexplored. On the other side, stopping the facility for re-executing those may also have negative consequences on critical applications that may depend on it. This interruption does not concern only the training phase, but all the preprocessing tasks mentioned in Section III, and notably data augmentation and unbalanced data processing.

TABLE IV. Estimation of the impact of context mining facility requirements on ML data criteria

ML Data criteria	Context mining facility requirements					
	R1	**R2**	**R3**	**R4**	**R5**	**R6**
Cleaning	😐	😐		😐	😐	😐
Transformation	😐	😐	😐		😐	
Feature extraction		😐	😐	😐	😐	😐
Data augmentation	😐		😐	😐	😐	
Proc. unbalanced data	😐		😐	😐		

(😐) Negative impact estimated

Table IV confronts context mining facility requirements summarized in Table II, and ML practices reported in Table III (since we do not analyze security and privacy aspects here, requirement R0 is not considered in Table IV). We may observe that requirements in Table II make some steps preconized by most of the approaches previously discussed more difficult to achieve. For instance, processing unbalanced data can be challenging when considering context data characteristics (R1), and notably uncertainty and incompleteness. Similarly, feature extraction and selection can be complex without human intervention or previous knowledge about the data. To sum up, Table IV highlights the fact that, although ML algorithms have proven to be useful for mining specific context data, the overall process necessary for applying those can be challenging when considering a large scale. This is true not only for supervised approaches, but also for semi-supervised ones, since they also require human intervention somewhere in the pipeline. Considering the IS scale, even a minimal human intervention may be complex. Similarly, unsupervised approaches are impacted too since they also rely on pre-processing. Indeed, any ML algorithm will fail to discover a hidden pattern or trend from raw data if that data is inadequate, irrelevant or incomplete.

V. Conclusions

Recent advances in middleware solutions allow to integrate new context data sources at runtime [44] [45]. The availability of such data raises multiple issues: how can ML algorithm exploit these data? How can we make such reasoning capabilities available to whatever application in an Information System? How can we make ML approaches scale up to a system (and not a precise application) scale?

In this paper, we presented a literature review of works that applied ML techniques to context data. Through this study, we could observe that several approaches were not general enough and often focused on specific types of

context data, ignoring the heterogeneity of such data. Most of these works consider context data as a "traditional" data, and do not fully take into account their features. Besides, the scale involved in a context mining facility also implies additional requirements, since it opens the possibility of an evolving set of applications, customers and context data, rather than precise and well-identified applications.

These observations highlight the challenges of applying traditional ML process in the case of a context mining facility. However, the use of ML for reasoning is this context is not impossible. Although ML techniques are currently evolving, they still require a sequence of pre-processing tasks that are hardly scalable. Their performance is also very sensitive to some design decisions such as algorithm selection, hyperparameters tuning, etc. Therefore, in order to reach a true context mining facility, we should evolve these practices towards powerful tools that gradually remove the human from the loop. We can already observe this tendency through initiatives such as automated ML (AutoML) frameworks [46], which aim at automating the entire ML pipeline. Among these frameworks, we can mention commercial solutions, hosted by leading cloud providers such as Amazon Machine Learning, Microsoft Azure Machine Learning and Google Cloud AutoML, as well as academic and open source frameworks such as Auto-WEKA [47] or autosklearn [48]. However, the latter only allow hyperparameters tuning and algorithms selection. We strongly believe that the future of a large scale context mining lies in the automatization of preprocessing phases and in the development of configurable frameworks that remain parametrizable according to application and organization needs. The potential of generalizing such reasoning capabilities is huge on different application areas, such as Smart cities and Industry 4.0. We are convinced that a stronger collaboration between ML experts and context specialists could help make ML solutions more flexible and adapted to context data, and further help reaching the full potential of context data for large Information Systems.

REFERENCES

[1] B. Yuan and J. Herbert, "Context-aware Hybrid Reasoning Framework for Pervasive Healthcare," *Personal Ubiquitous Comput.,* vol. 18, no. 4, pp. 865-881, 2014.

[2] A. K. Ramakrishnan, D. Preuveneers and Y. Berbers, "A Bayesian Framework for Life-Long Learning in Context-Aware Mobile Applications," in *Context in Computing: A Cross-Disciplinary Approach for Modeling the Real World*, P. Brézillon and A. J. Gonzalez, Eds., Springer NY, 2014, pp. 127-141.

[3] A. M. Otebolaku and G. M. Lee, "Towards context classification and reasoning in IoT," in *14th International Conference on Telecommunications (ConTEL)*, 2017.

[4] H. Rahman, R. Rahmani and T. Kanter, "Multi-Modal Context-Aware reasoNer (CAN) at the Edge of IoT," *Procedia Computer Science,* vol. 109, pp. 335 - 342, 2017.

[5] R. Mayrhofer, "Context Prediction based on Context Histories: Expected Benefits, Issues and Current State-of-the-Art," in *1st International Workshop on Exploiting Context Histories in Smart Environments (ECHISE 2005), 3rd International Conference on Pervasive Computing (PERVASIVE 2005)*, 2005.

[6] S. Sigg, S. Haseloff and K. David, "An alignment approach for context prediction tasks in ubicomp environments," *IEEE Pervasive Computing,* vol. 9, no. 4, pp. 90-97, 2010.

[7] D. Ameyed, M. Miraoui and C. Tadj, "A survey of prediction approach in pervasive computing," *International Journal of Scientific & Engineering Research,* vol. 6, pp. 306-316, 2015.

[8] S. Najar, M. Kirsch-Pinheiro and C. Souveyet, "Service discovery and prediction on Pervasive Information System," *J. of Ambient Intelligence and Humanized Comp.,* vol. 6, no. 4, pp. 407-423, 2015.

[9] C. D. Maio, G. Fenza, V. Loia, F. Orciuoli and E. Herrera-Viedma, "A framework for context-aware heterogeneous group decision making in business processes," *Knowledge-Based Systems,* vol. 102, pp. 39 - 50, 2016.

[10] C. Bauer and A. K. Dey, "Considering context in the design of intelligent systems: Current practices and suggestions for improvement," *Journal of Systems and Software,* vol. 112, pp. 26-47, 2016.

[11] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Computing,* vol. 6, no. 2, pp. 161-180, Apr 2010.

[12] N. Paspallis and G. A. Papadopoulos, "A Pluggable Middleware Architecture for Developing Context-aware Mobile Applications," *Personal Ubiquitous Comp.,* vol. 18, no. 5, pp. 1099-1116, 2014.

[13] M. Wagner, R. Reichle and K. Geihs, "Context as a service - Requirements, design and middleware support," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, 2011.

[14] K. Henricksen, J. Indulska and A. Rakotonirainy, "Modeling context information in pervasive computing systems," in *LNCS 2414 - First International Conference in Pervasive Computing (Pervasive 2002)*, 2002.

[15] D. Riboni and C. Bettini, "COSAR: hybrid reasoning for context-aware activity recognition," *Personal and Ubiquitous Computing,* vol. 15, pp. 271-289, 2011.

[16] G. Civitarese, R. Presotto and C. Bettini, "Context-driven Active and Incremental Activity Recognition," *arXiv preprint arXiv:1906.03033,* 2019.

[17] M. A. Razzaq, M. B. Amin and S. Lee, "An ontology-based hybrid approach for accurate context reasoning," in *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2017.

[18] C. Lork, B. Rajasekhar, C. Yuen and N. M. Pindoriya, "How many watts: A data driven approach to aggregated residential air-conditioning load forecasting," in *CoMoRea 2017, IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017.

[19] O. B. Sezer, E. Dogdu and A. M. Ozbayoglu, "Context-aware computing, learning, and big data in internet of things: a survey," *IEEE Internet of Things Journal,* vol. 5, pp. 1-27, 2017.

[20] S. García, J. Luengo and F. Herrera, "Tutorial on practical tips of the most influential data preprocessing algorithms in data mining," *Knowledge-Based Systems,* vol. 98, pp. 1-29, 2016.

[21] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak and F. Herrera, "A survey on data preprocessing for data stream mining: Current status and future directions," *Neurocomputing,* vol. 239, pp. 39-57, 2017.

[22] G. Mokhtari, Q. Zhang and A. Fazlollahi, "Non-wearable UWB sensor to detect falls in smart home environment," in *CoMoRea 2017, IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017.

[23] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez and J. L. Reyes Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proceedings of the 21th International European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2013.

[24] F. M. Rueda, S. Lüdtke, M. Schröder, K. Yordanova, T. Kirste and G. A. Fink, "Combining Symbolic Reasoning and Deep Learning for Human Activity Recognition," in *CoMoRea 2019, IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2019.

[25] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu and J. Zhang, "Convolutional neural networks for human activity

recognition using mobile sensors," in *6th Int. Conf. on Mobile Computing, Applications and Services*, 2014.

[26] F. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors,* vol. 16, p. 115, 2016.

[27] A. Zheng and A. Casari, Feature engineering for machine learning: principles and techniques for data scientists, " O'Reilly Media, Inc.", 2018.

[28] R. Grzeszick, J. M. Lenk, F. M. Rueda, G. A. Fink, S. Feldhorst and M. Hompel, "Deep neural network based human activity recognition for the order picking process," in *4th Int. Workshop on Sensor-based Activity Recognition and Interaction*, 2017.

[29] F. Al Machot, S. Ranasinghe, J. Plattner and N. Jnoub, "Human activity recognition based on real life scenarios," in *CoMoRea 2018, IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2018.

[30] I. Diaconita, A. Reinhardt, F. Englert, D. Christin and R. Steinmetz, "Do you hear what i hear? using acoustic probing to detect smartphone locations," in *CoMoRea 2014, IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, 2014.

[31] I. H. Sarker, A. S. M. Kayes and P. Watters, "Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage," *Journal of Big Data,* vol. 6, p. 57, 2019.

[32] N. Mishra, C.-C. Lin and H.-T. Chang, "A cognitive adopted framework for IoT big-data management and knowledge discovery prospective," *International Journal of Distributed Sensor Networks,* vol. 11, p. 718390, 2015.

[33] S. Saeedi, A. Moussa and N. El-Sheimy, "Context-aware personal navigation using embedded sensor fusion in smartphones," *Sensors,* vol. 14, pp. 5742-5767, 2014.

[34] S. Sani, N. Wiratunga and S. Massie, "Learning deep features for kNN-based human activity recognition.," in *ICCBR*, 2017.

[35] M. Miettinen, S. Heuser, W. Kronz, A.-R. Sadeghi and N. Asokan, "ConXsense: automated context classification for context-aware access control," in *Proceedings of the 9th ACM symposium on Information, computer and communications security*, 2014.

[36] L. D. Turner, S. M. Allen and R. M. Whitaker, "Push or delay? decomposing smartphone notification response behaviour," in *Human Behavior Understanding*, Springer, 2015, pp. 69-83.

[37] L. Bedogni, M. Di Felice and L. Bononi, "Context-aware Android applications through transportation mode detection techniques," *Wireless communications and mobile computing,* vol. 16, pp. 2523-2541, 2016.

[38] C. Catal, S. Tufekci, E. Pirmit and G. Kocabag, "On the use of ensemble of classifiers for accelerometer-based activity recognition," *Applied Soft Comp.,* vol. 37, pp. 1018-1022, 2015.

[39] A. Bayat, M. Pomplun and D. A. Tran, "A study on human activity recognition using accelerometer data from smartphones," *Procedia Computer Science,* vol. 34, pp. 450-457, 2014.

[40] Z. Feng, L. Mo and M. Li, "A Random Forest-based ensemble method for activity recognition," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015.

[41] D. Ravi, C. Wong, B. Lo and G.-Z. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," in *IEEE 13th Int. Conf. on Wearable and Implantable Body Sensor Networks (BSN)*, 2016.

[42] J. Yang, M. N. Nguyen, P. P. San, X. L. Li and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *24th Int. Joint Conf. on Artificial Intelligence*, 2015.

[43] C. A. Ronao and S.-B. Cho, "Deep convolutional neural networks for human activity recognition with smartphone sensors," in *Int. Conf. on Neural Information Processing*, 2015.

[44] P. Pradeep and S. Krishnamoorthy, "The MOM of context-aware systems: A survey," *Computer Communications,* vol. 137, pp. 44 - 69, March 2019.

[45] P. Temdee and R. Prasad, Context-Aware Communication and Computing: Applications for Smart Environment, Springer, 2018.

[46] F. Hutter, L. Kotthoff and J. Vanschoren, *Automated Machine Learning-Methods, Systems, Challenges,* Springer, 2019.

[47] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter and K. Leyton-Brown, "Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA," *The Journal of Machine Learning Research,* vol. 18, pp. 826-830, 2017.

[48] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum and F. Hutter, "Efficient and Robust Automated Machine Learning," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama and R. Garnett, Eds., Curran Associates, Inc., 2015, pp. 2962-2970.

Annex VII

Paper

Kirsch-Pinheiro, M.; Vanrompay, Y. & Berbers, Y., « Context-aware service selection using graph matching ». In: Paoli, F. D.; Toma, I.; Maurino, A.; Tilly, M. & Dobson, G. (Eds.), *2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC'08), at ECOWS 2008*, CEUR Workshop proceedings, vol. 411, 2008.

Context-Aware Service Selection Using Graph Matching

M. Kirsch-Pinheiro[1], Yves Vanrompay[2], Y. Berbers[2]

[1] Centre de Recherche en Informatique, Université Paris 1 Panthéon-Sorbonne
90 rue de Tolbiac, 75013 Paris, France
[2] Department of Computer Science, Katholieke Universiteit Leuven
Celestijnenlaan 200A, B-3001 Leuven, Belgium

Manuele.Kirsch-Pinheiro@univ-paris1.fr, Yves.Vanrompay@cs.kuleuven.be,
Yolande.Berbers@cs.kuleuven.be

Abstract. The current evolution of Ubiquitous Computing and of Service-Oriented Computing is leading to the development of context-aware services. Context-aware services are services whose description is enriched with context information related to the service execution environment, adaptation capabilities, etc. This information is often used for discovery and adaptation purposes. However, context information is naturally dynamic and incomplete, which represents an important issue when comparing service description and user requirements. Actually, uncertainty of context information may lead to inexact matching between provided and required service capabilities, and consequently to the non-selection of services. In order to handle incomplete context information, we propose in this paper a graph-based algorithm for matching contextual service descriptions using similarity measures, allowing inexact matches. Service description and requirements are compared using two kinds of similarity measures: local measures, which compare individually required and provided properties (represented as graph nodes), and global measures, which take into account the context description as a whole, by comparing two graphs corresponding to two context descriptions.

1 Introduction

The term Ubiquitous Computing, introduced by Weiser [22], refers to the seamless integration of devices into users' everyday life [1]. This term represents an emerging trend towards environments composed by numerous computing devices that are frequently mobile or embedded and that are connected to a network infrastructure composed of a wired core and wireless edges [13]. In pervasive scenarios foreseen by Ubiquitous Computing, context awareness plays a central role. Context can be defined as *any information that can be used to characterize the situation of an entity* (a person, place, or object considered as relevant to the interaction between a user and an application) [5]. Context-aware systems are able to adapt their operations to the current context, aiming at increasing usability and effectiveness by taking environmental context into account [1].

The dynamicity of pervasive environments encourages the adoption of a Service Oriented Architecture (SOA). Service-Oriented Computing (SOC) is the computing paradigm that utilizes services as fundamental elements for developing applications [15]. The key feature of SOA is that services are independent entities, with well-defined interfaces, that can be invoked in a standard way, without requiring the client to have knowledge about how the service actually performs its tasks [8]. Such loose coupling fits the requirements of high dynamic pervasive environments, in which entities are often mobile, entering and leaving the environment at any moment.

The adoption of SOA in pervasive environments is leading to the development of "context-aware" services. Context-awareness becomes a key feature necessary to provide adaptable services, for instance when selecting the best-suited service according to the relevant context information or when adapting the service during its execution according to context changes [6]. As pointed out by Maamar *et al*. [11], multiple aspects related to the users (level of expertise, location, etc.) and to the computer resources (on fixed and mobile devices), among others aspects, can be considered in the development of context-aware services. Thus, *context-aware services* can be defined as services which description is associated with contextual (notably non-functional) properties, *i.e*., services whose description is enriched with context information indicating the situations to which the service is adapted to.

According to Suraci *et al*. [18], in order to provide context-aware services, one has to consider *context inputs*, *besides functional inputs, and outputs*, which may also depend on contextual information. Several authors, such as Suraci *et al*. [18], Tonielli *et al*. [21] and Ben Mokhatar *et al*. [2], propose to increase service description with context information. This information is normally used for adaptation purposes: for adapting service composition; for indicating an execution environment (device capabilities, user's location, etc.) to which the service is designed for; for indicating adaptation capabilities (mainly content adaptation) of the service, etc. This context information needs to be compared to the real user's or execution context before starting to use the service.

However, in ubiquitous environments, context information is naturally dynamic and incomplete. Dynamic context changes and incomplete context information may prevent perfect matches between required and provided properties, which may lead to the non-selection of one (or all) service(s). Service selection mechanisms have to cope with these issues: if some needed context information is missing, service selection still has to proceed and choose a corresponding service that best matches the current situation, even if context information is incomplete. In other words, when executing in pervasive environments, service matching mechanisms have to deal with the question: how to reduce problems related to mismatching between contextual conditions related to the execution of a service and current context information?

In order to overcome this issue, we propose in this paper a graph-based algorithm for matching context-aware services. The proposed service selection mechanism assumes that suitable services exist. This means our approach is employed only after the question whether suitable services are available has been answered positively. The proposed algorithm matches contextual non-functional descriptions of context-aware services using similarity measures, allowing inexact matches. Service description and the current context are interpreted as graphs, in which properties correspond to graph nodes and the edges represent the relations between these properties. Through this

graph representation, service description and requirements are compared using two kinds of similarity measures: local measures, which compare individually required and provided properties (represented as graph nodes), and global measures, which take into account the context description as a whole, by comparing two graphs corresponding to two context descriptions. Moreover, we consider here only non-functional and context-related aspects of context-aware services. Even if functional aspects are the most relevant, once all services whose capabilities match functional requirements have been discovered, one has to select what service, among all the possible services, is the most suitable one, considering non-functional properties related to each service. Our graph-based service selection algorithm aims at selecting among available compatible services the most appropriate one considering the current context and taking into account the incompleteness of context information.

This paper is organized as follow: Section 2 presents an overview on related work. Section 3 introduces our approach of service selection. Section 4 presents the proposed matching algorithm and similarity measures. We conclude in Section 5.

2 Related work

A growing interest in context-aware services can be observed in the literature. For instance, several European projects are focusing on Service-Oriented Computing [16], and context-awareness appears as a crosscutting issue for these works. According to Tonielli *et al*. [21], in pervasive scenarios, users require context-aware services that are tailored to their needs, current position, execution environments, etc. According to Suraci *et al*. [18] user and service entities have requirements on context information they need in order to work properly. A user may have requirements on context of the service he is looking for (availability, location…) and on the context provided by the environment (wireless connection…). A service can require the user to provide specific context information (location, terminal capabilities…) and the environment to provide context information too (network QoS…).

The support for context-aware services depends on an improved semantic modeling of services by using ontologies that support formal description and reasoning [8]. Such a semantic modeling may contribute not only to handle problems related to service interoperability, but also in order to take into account different aspects of the environment in which the service is executed. Indeed, authors, such as Zarras *et al*. [24], advocate that semantic matching is essential for pervasive systems.

In the literature, several works, such as Ben Mokhatar *et al* [2], propose the semantic modeling and matching of services based on ontologies often expressed in OWL-based languages for enriching service description. These authors [2] propose the use of ontologies (in OWL-S) for the semantic description of functional and non-functional features of services in order to automatically and unambiguously discover such services. Klusch *et al*. [9] propose a service matching algorithm which combines reasoning based on subsumption and similarity measures for comparing inputs and outputs of service description and user request. Reiff-Marganic *et al*. [18] propose a method for automatic selection of services based on non-functional properties and

context. However, inexact matching caused by incomplete or uncertain context information is not taken into account.

Other authors such as Suraci *et al*. [18] and Yau & Liu [23] propose to improve service modeling with context information. Suraci *et al*. [18] propose a semantic modeling of services in which service profile description in OWL-S is enriched with a "context" element pointing to this required context information. Yau & Liu [23] propose to enrich service description with specific external pre- (and post-) conditions expressed in the OWL-S service description denoting contextual conditions for using a given service.

Tonielli *et al*. [21] propose a framework for personalized semantic-based service discovery. This framework aims at integrating semantic data representation and match-making support with context management and context-based service filtering. In such framework, services, users and devices are modeled through a set of profiles. describing capabilities and requirements of the corresponding service. The integration is then performed in a middleware using a matching algorithm based mainly on subsumption reasoning.

The majority of research cited above concentrates the semantic matching on solving ambiguity problems related to service inputs and outputs. Such works focus mainly on functional aspects, using semantic descriptions to enrich input and output description of services. Most works related to context-aware services, as those cited above, do not consider the natural uncertainty of context information. Context information is naturally dynamic and uncertain: it may contain errors, be out-of-date or even incomplete. Uncertainty in context information is traditionally handled by appropriate models, such as Chalmers *et al*. [4], who represent context values by intervals or sets of symbolic values. In these models, incompleteness of context information is seldom considered. However, semantic matching of context-aware services should take this into account. When considering context-aware services, matching algorithms have to consider the fact that some context information can be simply missing. Such incomplete information may lead to an inexact match between service description and requirements related to the user's current context.

In this paper, we focus particularly on this issue: how to deal with incompleteness of context information when selecting context-aware services. We propose a graph-based approach, in which service descriptions and requests are interpreted as graphs whose nodes and overall structure are compared by using similarity measures. The use of similarity measures in Computer Science is not new, as testifies the work of Liao *et al*. [10]. However, unlike Liao *et al*. [10], our work does not focus on proposing such measures. Our focus is to handle incompleteness of context information on service selection by using similarity measures. Such measures, in our case and unlike those proposed by Klush *et al*. [9], focus on non-functional and context-related aspects of context-aware services, and not on functional input and output of such services. In this sense, our approach is similar to the one proposed by Bottaro *et al*. [3], who propose ranking services according to context models evaluating the interests of a service in a composition. However, contrary to these authors, we are not particularly focusing on service composition, but on service selection in general.

3 Graph-based service selection

3.1 Proposal overview

The graph-based service selection approach proposed in this paper is part of a larger initiative, the MUSIC Project. The MUSIC Project [14] is a focused initiative aiming at the development of context-aware self-adapting applications. The main target is to support both the development and run-time management of software systems that are capable of being adapted to highly dynamic user and execution context, and to maintain a high level of usefulness across context changes. MUSIC adopts a component-based architecture, on which modeling languages allow the specification of context dependencies and adaptation capabilities. Such adaptation capabilities are based on the specification, at design time, of multiple variations (implementations) for each component. The selection of the most appropriate variation is performed by the MUSIC middleware, during run-time execution, based on the context dependencies associated with each variant and based on the current execution context.

In addition to MUSIC components, the MUSIC project aims at exploiting SOA by allowing MUSIC applications to use external services (*i.e.* services that are executing on non-MUSIC nodes). When considering those external services, we are interested in exploiting variability and non-functional properties of context-aware services in a similar way we consider for native MUSIC components. In other words, we consider that several service implementations can supply the same functional capabilities (with a similar syntax), but with different non-functional context-related properties.

The graph-based service selection approach proposed here contributes to the service selection mechanism used by the MUSIC Middleware for selecting the most suitable service among discovered and compatible services. Using this approach, the MUSIC Middleware compares context-aware service descriptions and current execution context in order to select most suitable service, considering current situation. The proposed service selection mechanism assumes that suitable services exist. It is part of a two-step process in which the first step selects all services whose functional properties match the functional requirements that are needed. This means our approach (the second step), dealing with non-functional requirements, is employed only after suitable services are discovered. So the proposal premises is the following: if there are several discovered services able to satisfy a request formulated by a user, one has to select the service that suits best the current execution context. Such service selection should take into account the fact that context information is naturally dynamic and incomplete.

We focus our approach on non-functional context-related aspects of service description. Indeed, we do not investigate functional aspects (inputs and outputs) of a service, but only non-functional contextual conditions related to the execution environment of a service. We consider that functional aspects of a service have the priority, since mismatching on service input or output may have negative (even disastrous) effects on the running application. Incompleteness on service input or output entries (missing input or output) can lead to severe exceptions (or errors), which may affect correctness and execution flow on both service and calling application. Thus, we decide to focus on non-functional aspects of context-aware

services, assuming a selection process for meeting functional requirements already took place.

We consider that each context-aware service describes a set of "contextual" conditions (non-functional properties) describing context elements needed for using it appropriately (in the best conditions). For instance, considering a content sharing service (*e.g.* a photo sharing), several variations of this service can be proposed using different implementations (*e.g.* implementations focusing a given user profile, a particular location, etc.). These contextual conditions refer potentially to any observed context element and they can be expressed using the MUSIC context model [17].

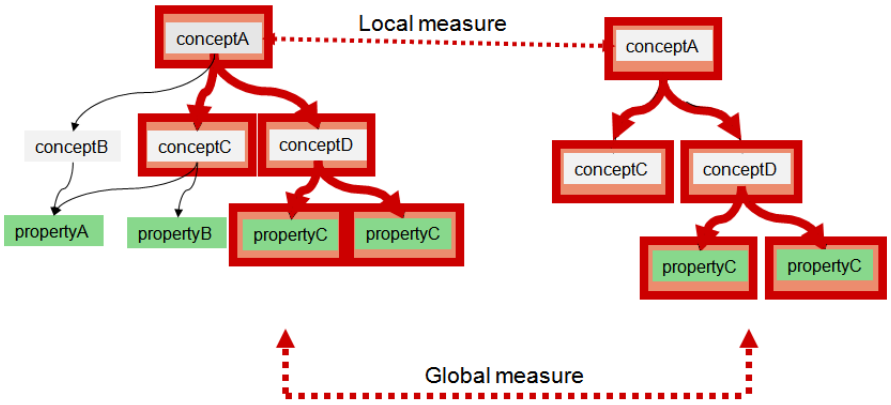


Fig. 1. Local and global measures comparing two graphs.

In order to perform service selection based on a "contextual" matching, service descriptions are enriched with non-functional context-aware properties related to the execution environment most suited for the service. Such requirements are included in the service profile description, using OWL-S. Such contextual description is analyzed as a graph, in which objects represent concepts and properties and edges represent the relations connecting such concepts. The same analysis is performed on the description of the current execution context, which is represented based on an OWL-ontology, and which acts like a "request" (requested execution environment) for the service. This allows us to compare both based on similarity measures between graphs. The proposed service selection algorithm then ranks the available services, indicating to the MUSIC middleware (our user) the services that best match the current context.

In order to compare the graphs built using service description and current context description, we propose local and global similarity measures. Local measures compare two nodes individually, considering only the concept it represents and its properties. Global measures take into account the graph as a whole, evaluating, for instance, the proportion of similar elements in both graphs. By using such measures, our approach allows dealing with incomplete context information and inexact matching between conditions expressed in the service description and current context description, since missing information on the latter will not block the analysis and the ranking of the former. This means that the selection looks for the service that matches the best the contextual conditions, but is not necessarily a perfect match. Fig. 1 illustrates these measures. It shows a local measure comparing two individual

concepts labeled *conceptA*, and a global measure comparing the graphs formed by these concepts (highlighted in Fig.1). Moreover, this approach assumes that several measures can be considered in order to evaluate local values. These local measures are associated to particular context scopes defined in the MUSIC ontology, taking into account the semantic aspects represented in the ontology.

3.2 Describing context-aware services

Service descriptions are expressed in OWL-S. According to Suraci *et al*. [18], "*for describing the semantics of services, the latest research in service-oriented computing recommends the use of the Web Ontology Language for Services (OWL-S)*." These authors consider that, even if OWL-S is tailored for Web Services, it is rich and general enough to describe any service. We consider to enrich this description with context information describing the execution context for which the service is best suited. For instance, let us consider a mobile content sharing platform that enables users to browse, search for, and share multimedia content scattered on such devices in different situations, such as conferences, shopping malls, football stadiums, etc. This scenario foreseen by the MUSIC Project is called Instant Social [7] and it proposes to explore cooperating multi-user applications hosted on mobile devices carried by users. In this scenario, several content sharing services can be available on the platform. Each service can indicate contextual conditions in which it runs appropriately. For example, a given photo sharing service can be particularly designed considering client devices with high screen resolution and memory capacities, a second implementation of the same service can be designed considering a particular location (a conference hall or a stadium), or a particular user profile (*e.g*. adult users).

Such contextual information can be considered as part of the service description, since it indicates situations to which the service is better suited. A service description in OWL-S includes three main parts [12]: (i) service profile; (ii) service model; and (iii) service grounding. The service profile corresponds roughly to the service description. The service model specifies the process executed by the service. The service grounding indicates how the service can be accessed (like an API).

Thus, similarly to Suraci *et al*. [18], we propose to enrich the service profile with a "*context*" element pointing to context description related to the service. This description should be included in an external file (indicated in the "*context*" element) and not directly in the OWL-S description. Context information is dynamic and cannot be statically stored on the service profile. On the one hand, context properties related to the execution of a service can evolve and vary according to the service execution environment itself. For instance, the load of the device executing a service may affect the service and consequently the context properties related to it. On the other hand, the service profile is supposed to be a static description of the service in the sense that it is not supposed to change in short intervals of time (as context information does). An external file describing contextual non-functional requirements and properties related to a service allows the service supplier to easily update such context information related to the service without modifying the service description itself. Fig. 2 presents an example of service profile including the "*context*" element. This example illustrates the extended profile of a photo sharing service, like those

foreseen in MUSIC project scenario. This service returns, for a given request on input, a list of interesting photos and a map locating them. As stated before, such a service may have different implementations, considering particular contexts. The one related to this particular implementation is given in Fig. 3.

```
- <profile:Profile rdf:ID="CONTEXT_SHARING_MAP_PROFILE">
    <service:isPresentedBy rdf:resource="#CONTEXT_SHARING_MAP_SERVICE"/>
    <profile:serviceName xml:lang="en"> ContextMapPhotoSharingService </profile:serviceName>
  - <profile:textDescription xml:lang="en">
      This service provides a facility to find shared photos available in a location.
    </profile:textDescription>
    <eprofile:context rdf:resource="http://127.0.0.1/services/contextdescriptionV2.xml"/>
    <profile:hasInput rdf:resource="#_REQUEST"/>
    <profile:hasOutput rdf:resource="#_LIST"/>
    <profile:hasOutput rdf:resource="#_MAP"/>
    <profile:has_process rdf:resource="CONTEXT_SHARING_MAP_PROCESS"/>
  </profile:Profile>
```

Fig. 2. Example of service profile including the property "context".

Fig. 3 presents an example of a context description related to the service in Fig. 2. This description follows the MUSIC Context Model described in Reichle *et al*. [17]. The MUSIC context modeling approach identifies three basic layers of abstraction that correspond to the three main phases of context management: the conceptual layer, the exchange layer and the functional layer.

The conceptual layer enables the representation of context information in terms of *context elements*. The *context elements* provide context information about *context entities* (the concrete subjects the context data refers to: a user, a device, etc.) belonging to specific *context scopes*. Such context scopes are intended as semantic concepts belonging to a specific ontology described in OWL. Moreover, the ontology is used to describe relationships between entities, *e.g*. a user has a brother. The exchange layer focuses on the interoperability between devices. Context data in this layer is represented in XML and is used for communication between nodes. The functional layer refers to the implementation of the context model internally to the different nodes.

The description illustrated in Fig. 3 belongs to the exchange level, since it is used for information exchange among different nodes. Thus, context information in Fig. 3 is described in XML by context elements, which refer to a given entity and scope, and a set of context values, which also refer to a given scope. It is worth noting that Fig. 3 supplies two separate context descriptions: (*i*) a first description (under the element "*condition*") supplying the conditions under which this service adapts the best (*i.e*. the contextual situation in which it is most appropriate to call this service); and (*ii*) a second description referring to the current state of the service execution context (under which conditions this service is running on the service supplier). Thus, through the condition element in Fig. 3, the service supplier indicates that the content supplied by this service implementation (whose profile is represented in Fig. 2) is proper to tourist users (who are familiar with the city they are visiting) and that this service disposes of a detailed database for the city of Paris, which makes it better adapted to being used when in this location. The next section describes how the proposed graph-based matching algorithm considers and handles these descriptions.

```
− <ctx:context xsi:schemaLocation="http://www.ist-music.org/ContextSchema ContextSchema.xsd ">
  − <ctx:condition>
    − <ctx:contextElement>
        <ctx:hasEntity resource="http://www.ist-music.org/Ontology/ContextModel.owl#concept.entityType.user"/>
        <ctx:hasScope resource="http://www.ist-music.org/Ontology/ContextModel.owl#concept.contextScope.location"/>
        <ctx:hasRepresentation resource="http://www.ist-music.org/Ontology
        /ContextModel.owl#concept.representation.locationDefaultRepresentation"/>
      − <ctx:contextValueSet>
        − <ctx:contextValue>
            <ctx:hasScope resource="http://www.ist-music.org/Ontology
            /ContextModel.owl#concept.contextScope.location.city"/>
            <ctx:hasRepresentation resource="http://www.ist-music.org/Ontology
            /ContextModel.owl#concept.representation.locationDefaultRepresentation"/>
            <ctx:value>Paris</ctx:value>
          </ctx:contextValue>
        </ctx:contextValueSet>
      </ctx:contextElement>
    − <ctx:contextElement>
        <ctx:hasEntity resource="http://www.ist-music.org/Ontology/ContextModel.owl#concept.entityType.user"/>
        <ctx:hasScope resource="http://www.ist-music.org/Ontology
        /ContextModel.owl#concept.contextScope.userprofile"/>
        <ctx:hasRepresentation resource="http://www.ist-music.org/Ontology
        /ContextModel.owl#concept.representation.profileDefaultRepresentation"/>
      − <ctx:contextValueSet>
        − <ctx:contextValue>
            <ctx:hasScope resource="http://www.ist-music.org/Ontology
            /ContextModel.owl#concept.contextScope.profile.category"/>
            <ctx:hasRepresentation resource="http://www.ist-music.org/Ontology
            /ContextModel.owl#concept.representation.profileDefaultRepresentation"/>
            <ctx:value>Tourist</ctx:value>
          </ctx:contextValue>
        </ctx:contextValueSet>
      </ctx:contextElement>
    </ctx:condition>
  + <ctx:state></ctx:state>
  </ctx:context>
```

Fig. 3. Example of context description associated to a service.

4 Graph-based matching

4.1 From description to graphs

The first step for performing the graph-based matching is to analyze the context description associated with the available service. Based on the context description presented above, we propose a graph-based approach for ranking and selecting services. In this approach, non-functional context-related properties of the services represented in the context description file described previously are interpreted as a graph. In this graph, nodes represent the context elements indicated in this description, and the edges represent the relations that can exist between these elements. The same interpretation is used when analyzing the current execution context. The MUSIC middleware is responsible for service selection and for collecting and managing context information related to the user. It keeps this

information in context elements expressing their current values. These context elements are seen as graph nodes, whereas relations between such elements are seen as graph edges. Thus, a graph G is defined as follow:

- $G = < N, E >$ where:
  - $N = \{ C_{Ei} \}_{i>0}$: set of context elements C_{Ei} ;
  - $E = \{ < C_{Ei}, C_{Ej} > \}$: set of relations between context elements C_{Ei} and C_{Ej}.

Thus, comparing two graphs representing two different context descriptions corresponds, with regard to the MUSIC Context Model, to comparing two sets of context elements and their relations.

4.2 Matching algorithm

Once all available services have been analyzed and their corresponding graphs are created, the matching based algorithm may proceed. The goal of this matching algorithm is to rank the available services based on their contextual non-functional properties. It compares the graph generated by each proposed service to the graph created based on the current execution context information. This matching starts by comparing nodes from both graphs (from the context description of the service and from the current context) individually, using local similarity measures. Based on the results of these measures, the matching algorithm compares the graphs globally, using global similarity measures that also consider the edges connecting the nodes. The results of such global measures are used to rank the services corresponding to the compared graphs. Next sections present both local and global similarity measures.

4.2.1 Comparing graph nodes: local similarity measures

When comparing two nodes from two graphs defined in Section 4.1, we are comparing two *context elements* representing context information about a given entity and referring a given scope. By considering these elements individually, we focus on how similar their context values are. In order to perform this comparison, we consider local similarity measures $Sim_l (C_{Ei}, C_{Ej})$ that compares two context elements C_{Ei} and C_{Ej} locally (i.*e*. without considering their position in the corresponding graphs). This measure can be defined as follows:

- $Sim_l (C_{Ei}, C_{Ej}) = x$, where $x \in \mathbb{R}, x \in [0, 1]$

Ideally, the similarity measure $Sim_l (C_{Ei}, C_{Ej})$ depends on the context scope. If the context elements being compared do not belong to compatible context scopes, their similarity is by definition zero. For example, we cannot compare context elements referring to the user's age or preferences with context elements referring to the user's location because both elements belong to context scopes that are incompatible. Similarity measure $Sim_l (C_{Ei}, C_{Ej})$ has to consider the representation associated with the context elements. In the MUSIC context modeling approach each context element is associated with a corresponding representation. For instance, considering location information, this can be represented using geographical coordinates like latitude and longitude (*e.g. 48°49'38" N, 2°21'02" E*), as well as using a representative name (*e.g.*

Paris, France). Each measure $Sim_l\ (C_{Ei}, C_{Ej})$ is proposed considering a given set of possible representations, which it may handle. Only context elements that correspond to the context scope and representation supported by the giving measure can be compared using it. The MUSIC middleware keeps then a library with all knows similarity measures $Sim_l\ (C_{Ei}, C_{Ej})$. Before comparing two nodes, it looks for the appropriate measure in its library.

Once the appropriate similarity measure $Sim_l\ (C_{Ei}, C_{Ej})$ is chosen, the matching starts by taking each node in the graph corresponding to the context description of the service and comparing it to the nodes with a compatible scope and representation from the graph corresponding to the current execution context. For each node, it keeps tracks of the best-ranked node, in order to use this value in the global similarity measures (Section 4.2.2). Thus, being $G_{Sk} = < N_{Sk}, E_{Sk} >$ the graph corresponding to the service S_k and $G_C = < N_C, E_C >$ the graph corresponding to the current context, we compare each node C_{Ei} from G_{Sk} to all nodes C'_{Ei} in G_C for which $C_{Ei}.scope$ and $C'_{Ei}.scope$ and $C_{Ei}.representation$ and $C'_{Ei}.representation$ are compatible, keeping in memory the best-ranked C'_{Ei}. For example, considering the graph generated by the context description in Fig. 3, the node referring to the user's profile is compared to all nodes having the same scope (*user profile*) in the graph corresponding to current user's context.

4.2.2 Comparing graphs: global measures

The main goal of global similarity measures is to compare overall composition of two graphs, taking into account both nodes and edges composing each graph. We define such measures as follow:

- $Sim_g\ (G_{Sk}, G_C) = x, x \in \mathbb{R}$, where
  - G_{Sk} corresponds to the graph determined by the context description of the service;
  - G_C corresponds to the graph determined based on the current execution context.

Several global measures $Sim_g\ (G_{Sk}, G_C)$ are possible for comparing two graphs. These measures can be based on different well-know algorithms such as subgraphing matching or graph isomorphism. The most important aspect for us is that the global similarity measure $Sim_g\ (G_{Sk}, G_C)$ must support incompleteness of context information represented in these graphs. This means that the $Sim_g\ (G_{Sk}, G_C)$ should not stop processing if some context information is missing. For instance, if the context description of a service refers to a given context element for which there is no corresponding element with a compatible context scope in the current context description, the similarity measure $Sim_g\ (G_{Sk}, G_C)$ should continue the processing, arriving in a valuable result that takes into account this fact.

In the MUSIC middleware, we consider a single yet powerful similarity measure $Sim_g\ (G_{Sk}, G_C)$ defined based on the proportion of nodes and edges belonging to the context description of the service that have a similar correspondence in the current context description. For this, the similarity measure considers the results obtained by the local similarity measures. For each pair $<C_{Ei}, C'_{Ei}>$, with $C'_{Ei} \in G_{Sk}$ and $C'_{Ei} \in G_C$ and C'_{Ei} being the node of G_C with the greatest value for $d(C_{Ei}, C'_{Ei})$, the proposed measure $Sim_g\ (G_{Sk}, G_C)$ analyses the similarity among the edges connecting

these nodes to their neighbors. The similarity between two edges is calculated based on the similarity of their corresponding labels (or weights), if the edges are labeled, and the similarity between the objects forming the edges. Similarly to the local measures, we consider in the global measure only the greatest value obtained when comparing each edge connecting a node C_{Ei}. Then, we sum up both nodes and edges best similarities measures and make the proportion taking into account the total number of nodes and edges in graph defined by the context description of the service. Fig. 4 shows the definition of the measure $Sim_g\ (G_{Sk}, G_C)$.

It is worth noting that, since the maximum value for $Sim_l\ (a,b)$ is *1* (cf. Section 4.2.1), if the graph G_{Sk} is a subgraph of G_C, for each node and edge, we will have a corresponding node or edge for which the local similarity measure is 1. Thus, by considering the proportion of the greatest values obtained for all individual nodes and edges in the total size of the graph, this measure considers implicitly that some nodes or edges may have no similar element ($max(Sim_l\ (a,b))=0$). This eventuality leads to a reduction in the value of the global similarity measure $Sim_g\ (G_{Sk}, G_C)$, but it does not prevent a valuable result. Even if the compared graphs have no element in common ($max(Sim_l\ (a,b))=0$ for all $a \in G_{Sk}$ and $b \in G_C$), the measure $Sim_g\ (G_{Sk}, G_C)$ still returns a value that can be used to rank the service. For instance, when considering the photo sharing service represented Fig. 2, the measure $Sim_g\ (G_{Sk}, G_C)$ gives a valuable result ($x \geq 0$) even if the current user's context does not possess any context element referring to the location scope (user's device has no GPS or any location sensor available). This resulting value is then used to rank this particular implementation of photo sharing service. Incompleteness of context information is dealt with in this way.

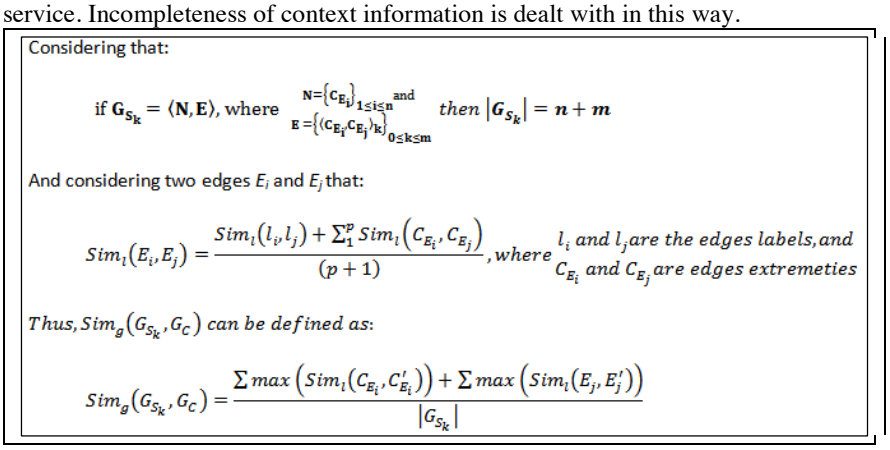


Fig. 4. Definition of the global similarity measure Sim_g.

5 Conclusions

In this paper, we present a graph-based approach for service selecting in ubiquitous computing. The main goal of this approach is to select the most adapted service for the current situation. We compare contextual non-functional properties of context-aware services to the current execution context in which they are called. Our approach

considers particularly the natural incompleteness of context information when selecting a context-aware service among all available services. For this, our approach is based on a graph-based analysis of both current context situation and context description associated with the service. This analysis is the basis for a set of similarity measures that compare graphs representing these descriptions. Such measures allow us to compare graphs that represent context information by considering scope and incompleteness of such information.

Currently, we are testing the proposed approach with the MUSIC middleware in order to evaluate its performance in ubiquitous environments. We also intend to compare our results with other libraries of similarity measure such as SimPack [20].

Acknowledgements. The authors would like to thank their partners in the MUSIC-IST project and acknowledge the partial financial support given to this research by the European Union (6th Framework Programme, contract number 35166).

References

[1] Baldauf, M.; Dustdar, S. & Rosenberg, F., A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing, vol. 2, n.4, 2007, 263–277

[2] Ben Mokhtar, S.; Kaul, A.; Georgantas, N. & Issarny, V., Efficient Semantic Service Discovery in Pervasive Computing Environments, Proceedings of the ACM/IFIP/USENIX 7th International Middleware Conference (Middleware'06), 2006

[3] Bottaro, A.; Gerodolle, A. & Lalanda, P., Pervasive service composition in the home network, 21st International IEEE Conference on Advanced Information Networking and Applications (AINDA'2007), 2007

[4] Chalmers, D.; Dulay, N. & Sloman, M., Towards Reasoning About Context in the Presence of Uncertainty, 1st international workshop on advanced context modelling, reasoning and management, Nottingham, UK, September 2004

[5] Dey, A., Understanding and using context. Personal and Ubiquitous Computing, vol. 5 n. 1, 2001, 4-7.

[6] Eikerling, H.-J.; Mazzoleni, P.; Plaza, P.; Yankelevich, D. & Wallet, T., Services and mobility: the PLASTIC answer to the Beyond 3G challenge. White Paper, PLASTIC Project, December 2007. http://www-c.inria.fr/plastic/dissemination/plastic/dissemination

[7] Fraga, L.; Hallsteinsen S. & Scholz, U., InstantSocial – Implementing a Distributed Mobile Multi-user Application with Adaptation Middleware, Communications of the EASST, vol. 11. http://eceasst.cs.tu-berlin.de/index.php/eceasst/issue/view/18.

[8] Issarny, V.; Caporuscio, M. & Georgantas, N., A Perspective on the Future of Middleware-based Software Engineering. In: Briand, L. and Wolf, A. (Eds.), Future of Software Engineering 2007 (FOSE), ICSE (International Conference on Software Engineering), IEEE-CS Press. 2007

[9] Klusch, M.; Fries, B. & Sycara, K., Automated semantic web service discovery with OWLS-MX, Proceedings of the 5th International joint conference on Autonomous agents and multiagent systems (AAMAS '06), ACM, 2006, 915-922

[10] Liao, T.W.; Zhang, Z. & Mount, C.R., Applied Artificial Intelligence, Volume 12, Number 4, 1 June 1998 , Taylor & Francis, 267-288

[11] Maamar, Z.; Benslimane, D. & Narendra, N. C., What can context do for web services?, Communication of the ACM, vol. 49, n° 12, Dec. 2006, 98-103

[12] Martin, D. (Ed.), OWL-S: Semantic Markup for Web Services, W3C Member Submission 22 November 2004, http://www.w3.org/Submission/OWL-S

[13] Moran, T. & Dourish, P., Introduction to this special issue on context-aware computing. Human-Computer Interaction, vol. 16, n. 2-3, 2001, 87–95

[14] MUSIC Consortium, Self-Adapting Applications for Mobile Users in Ubiquitous Computing Environments (MUSIC), Website: http://www.ist-music.eu/

[15] Papazoglou, M. P. & Georgakopoulos, D. Service-Oriented Computing, Communication of ACM, vol. 46, n. 10, Oct. 2003, 24-28
[16] Patouni, E.; Alonistioti, N. & Polychronopoulos, C. (eds.), Service Adaptation over Heterogeneous Infrastructures, White Paper, 2008. http://www.opuce.tid.es/Publications.htm
[17] Reichle, R.; Wagner, M.; Khan, M.U.; Geihs, K.; Lorenzo, L.; Valla, M.; Fra, C.; Paspallis, N. & Papadopoulos G.A. A Comprehensive Context Modeling Framework for Pervasive Computing Systems, 8th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS), 4-6 June, 2008, Oslo, Norway, Springer.
[18] Reiff-Marganiec, S., Qing Yu, H., Tilly, M., Service Selection based on Non-Functional Properties, NFPSLA-SOC07 Workshop at The 5th International Conference on Service Oriented Computing (ICSOC2007), Sept. 17 2007, Vienna, Austria.
[19] Suraci, V.; Mignanti, S. & Aiuto, A., Context-aware Semantic Service Discovery, 16th IST Mobile and Wireless Communications Summit, 2007, 1-5
[20] SimPack Project Page, http://www.ifi.uzh.ch/ddis/research/semweb/simpack/
[21] Toninelli, A.; Corradi, A. & Montanari, R., Semantic-based discovery to support mobile context-aware service access, Computer Communications, vol. 31, n° 5, March 2008, 935-949.
[22] Weiser, M. The computer for the 21st century, Scientific American, vol. 66, 1991.
[23] Yau, S. S. & Liu, J., Incorporating situation awareness in service specifications, In: Lee, S.; Brinkschulte, U.; Thuraisingham, B. & Pettit, R. (Eds.), 9th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC 2006), 2006, 287-294
[24] Zarras, A.; Fredj, M.; Georgantas, N. & Issarny, V., Engineering reconfigurable distributed software systems: issues arising for pervasive computing, In: Butler, M.; Jones, C.; Romanovsky, A. & Troubitsyna, E. (Eds.), LNCS 4157, Rigorous Development of Complex Fault-Tolerant Systems, Spring, 2006, 364-386

Annex VIII

Paper

Najar, S.; Kirsch-Pinheiro, M.; Souveyet, C. & Steffenel, L. A., “Service Discovery Mechanisms for an Intentional Pervasive Information System”. *Proceedings of 19th IEEE International Conference on Web Services (ICWS 2012)*, Honolulu, Hawaii, 24-29 June 2012, pp. 520-527.

2012 IEEE 19th International Conference on Web Services

Service Discovery Mechanism for an Intentional Pervasive Information System

Salma Najar, Manuele Kirsch Pinheiro, Carine Souveyet
Centre de Recherche en Informatique - Université Paris1
90 rue de Tolbiac, 75013 Paris – France
Salma.Najar@malix.univ-paris1.fr,
{Manuele.Kirsch-Pinheiro, Carine.Souveyet}@univ-paris1.fr

Luiz Angelo Steffenel
CReSTIC – Syscom Team
Université de Reims Champagne-Ardenne
51100 Reims - France
Luiz-Angelo.Steffenel@univ-reims.fr

***Abstract*—Pervasive Information System (PIS) provides a new vision of Information System available anytime and anywhere. The users of these systems must evolve in a space of services, in which several services are offered to him. However, PIS should enhance the transparency and efficiency of the system. We believe that a user-centric vision is needed to ensure a transparent access to the frequently changing space of services regardless of how to perform it. In this paper, we propose a new approach of PIS, both context-aware and intentional called IPIS. In this approach, services are proposed in order to satisfy user's intention in a given context. Then, we propose a context-aware intentional service discovery mechanism. Such mechanism is based on an extension of OWL-S taking into account the notion of context and intention. We present in this paper IPIS platform. Then, we detail the proposed service discovery mechanism and present experimental results that demonstrate the advantage of using our proposition.**

Keywords-component; pervasive information system; service orientation; intention; contexte-aware service;

I. INTRODUCTION

Nowadays, with the development of mobile technologies, we observe a shift in Information Systems. Instead of having Information Technology in the foreground, triggered and manipulated by users, we witness that IT gradually resides in the background, monitoring user's activities, processing this information and intervening when required [8]. In other terms, we are observing the emergence of a *Pervasive Information System* (PIS) that intends to increase user's productivity by making IS available anytime and anywhere.

Contrarily to traditional IS, whose interaction paradigm is the desktop, PIS deal with a multitude of heterogeneous devices, providing the interaction between the user and the physical environment [8]. We believe that PIS should evolve in a *space of services* offering to users a set of heterogeneous services helping them to accommodate their needs.

Weiser [20] suggests that pervasive environment will be characterized by its transparency and homogeneity. Twenty years later, we can notice that this pervasive environment meant to be an invisible or unobtrusive one, represents a technology-saturated environment combining several devices highly present and visible. PIS have to face such an environment, in which a rapidly evolving and increasing number of services are available, with multiple implementations. However, details concerning such implementation are still too complex for the user, who wants just a service that satisfies his needs. This complexity requires considerable effort from the user in order to understand what is happening around him and in order to select the service that best fulfills his needs.

In order to reach such vision of PIS, these systems should enhance its transparency. We believe that this will only be possible through a user-centric vision that ensures a transparent access to the space of services without any technical details concerning how to perform it in a given context. Thus, we propose a service discovery mechanism guided by user's intention and context in order to hide implementation complexity, and consequently achieving the transparency promised. First, the notion of intention can be seen as the goal that we want to achieve without saying how to perform it [7] or as a goal to be achieved by performing a process presented as a sequence of intentions and strategies to the target intention [3]. In other words, an intention represents a *requirement that a user wants to be satisfied without really care about how to perform it or what service allows him to do so*. Then, the context concept represents *any information that can be used to characterize the situation of an entity (a person, place, or object)* [4].

By adopting this vision, we propose to improve the transparency by considering, on the one hand, the intention service allows user to satisfy, and on the other hand, context on which this intention emerges. Thus, we advocate that the selection of the service that satisfies user's intention, in PIS, is valid in a given context, and a context influences the manner to satisfy user's intention and the execution of the service that support it. We believe this relation can be explored for service discovering purposes: by observing intention and context, we can obtain a more precise service discovery mechanism, offering most suitable services.

In this paper, we present an *intentional pervasive information system* platform (IPIS), which enhances the transparency of the system by proposing a service discovery mechanism guided by user's intention and context. The service discovery, based on these two concepts (context and intention), will help users by discovering for them the most appropriate service, without carrying about any technical details concerning how to perform it in a given context.

This paper is organized as follows: Section II presents an overview on related works. Section III introduces our IPIS platform, while section IV details our proposed service discovery process. The section V presents the implementation of the service discovery, while the section VI presents the experiments results. Finally, we conclude in the section VII.

978-0-7695-4752-7/12 $26.00 © 2012 IEEE
DOI 10.1109/ICWS.2012.84

II. RELATED WORK

In the last decade, an important change has been performed on IS. Those systems become, with the growth of information technology, accessible for users on many different circumstances and too complex for them. These changes lead to the notion of PIS. *Pervasive Information System* (PIS) constitute an emerging class of IS, in which information technology is gradually embedded in the physical environment, capable of accommodating user needs and wants when desired [8]. This author points out, as main characteristics of PIS, not only the heterogeneity of device types, but also the property of *context-awareness*, as a result of the artifacts capabilities to collect, process and manage environmental information. Therefore, when regarding the literature, one may observe two complementary tendencies proposing a new vision of the information system: context-aware approaches and intention-based approaches.

Toninelli et al. [19], consider that, in pervasive scenarios, users require context-aware services that are tailored to their needs, current position, execution environments, etc. These authors propose a personalized semantic-based service discovery that aims at integrating semantic data representation and matchmaking support with context management and context-based service filtering.

Similar to Toninelli *et al.* [19], Ben Mokhtar *et al.* [2] propose a context aware semantic matching of services. They propose, for service discovery purposes a language for semantic specification of functional and non-functional service properties named EASY-L and a corresponding set of conformance relations named EASY-M.

Xiao *et al.* [21] consider context-aware services and the dynamicity of pervasive environment. They use ontologies to enhance the meaning of a user's context values and automatically identify the relations among different context values. Based on these relations, they discover and select the potential services that the user might need.

These authors [2][19][21] emphasize the importance of the context on service discovery. They focus on the technical level that is still too complicate for users who would prefer just a service that satisfy their needs.

A different point of view is given by works such as [1] [9][12][17], which pointed out the importance of considering user's requirements on service orientation. Among these works, [7][17] propose a service oriented architecture based on an intentional perspective. Such architecture proposes the notion of *intentional service*, which represents a service focusing on the intention that it allows satisfying it rather than the functionality it performs. Such service represents an alternative for bridging the gap between low level, technical software-service descriptions and high level, strategic expressions of business needs for services.

Aljoumaa *et al.* [1] propose, based on the Intentional Services Model (ISM) proposed by [17], an ontological based solution to help matching user's needs formulated in business terms as goals with the intentions of services published in an extended registry.

Mirbel et al. [9] propose using semantic models in order to enrich needs description and to propose means of reasoning for intention based service discovery. This approach express user's requests using semantic Web technologies and help such users to find available services that fit these requests.

Olson *et al.* [12] believe that by using goals, services can be described on any arbitrary and useful level of abstraction. According to these authors, through a goal refinement algorithm, goals can be used not only for describing services, but also for improving the performance of service discovery.

None of these works considers the notion of context, contrary to *Bonino et al.* [3], which propose a goal-based dynamic service discovery framework that uses context information. This approach is centered on the use of context information for filtering the input of the user's request. Besides, These authors identify in advance, for each domain, a set of specific intentions and the different tasks allowing the fulfillment of them, which is quit restrictive since it cannot satisfy an intention that is not identified in advance.

All these works advocate for a more user-centric view of IS, by proposing either an intentional or a contextual approach. However, we believe that such user-centric vision can only be reached if one considers the closely relation between the notion of context and intention. We advocate that a service discovery mechanism based on user's context and intention is needed in order to provide answers to questions such as "*why a service is useful in a given context?*" or "*in which circumstances a service need raises?*". For us, *an intention is valid in a given context and a context has an influence on the satisfaction of this intention*. However, as far as we know, none of the previously works proposes a pertinent service discovery mechanism taking into account context and intention.

III. TOWARDS AN INTENTIONAL PERVASIVE INFORMATION SYSTEM

In this paper, we present a user-centric view of PIS called *Intentional Pervasive Information System* (IPIS). IPIS purpose is to propose to users the most appropriate service in a transparent way. It intends to enhance service discovery, on a space of services, according to the user's context and intention, without an explicit help from the user.

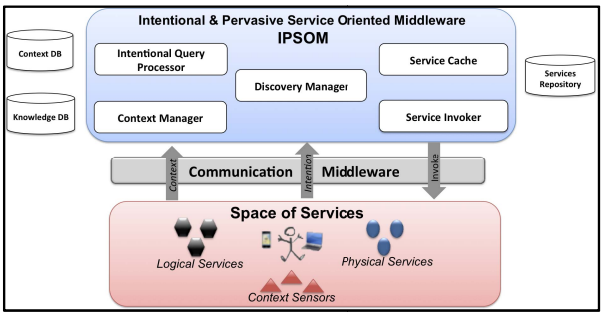


Figure 1. IPIS: Global Architecture

We advocate that understanding user's intention can lead to a better understanding of the real use of a service and consequently the selection of a suitable service that satisfies

user's needs. On the other hand, contextual information plays a central role since it influences the selection of the best strategies of the intention satisfaction.

We present in this section a global vision of IPIS platform as illustrated in Figure 1. IPIS is composed of three main layers: space of services (section III.A), communication layer (section III.B) and the Intentional Pervasive Service-Oriented Middleware (section III.C).

A. Space of services

The evolution of IS into PIS lead us to consider PIS as more than a set of logical services. Indeed, with the development of pervasive technologies, IT becomes embedded in the physical environment, offering innovative services to the users evolving on this environment. Contrarily to traditional IS, PIS may offer both logical and physical services. For us, in a *pervasive information system,* users evolves in a *space of services* offering them a set of heterogeneous services whose focus is to accommodate user's needs. Thus, we introduce the notion of *space of services* as a way to develop such user-centric view through a space that includes not only *logical services,* representing traditional information systems themselves, but also *physical services* embedded on the physical world. This space allows the representation of knowledge about users and their environment, in order to discover the most appropriate service that satisfies a user's intention in a given context.

B. Communication Layer

The communication layer aims at connecting the space of services with our *Intentional Pervasive Service-Oriented Middleware* 'IPSOM'. User requests, context capture, service invocation, etc. are sent to IPSOM through this layer.

C. IPSOM: Intentional Pervasive Service Oriented Middleware

The Intentional Pervasive Service oriented middleware 'IPSOM' represents the core of IPIS. Its main purpose is to satisfy user's intention by discovering and selecting the most suitable service in a given context. This middleware, presented in Figure 1 contains four main modules: *context manager, intentional query processor, service invoker and service discovery*.

1) Context Manager

The *context manager* purpose is to hide the context management complexity by providing a uniform way to access context information. It receives raw context data from different physical and logical sensors (GPS, RFID...) and interprets such data in order to derive context knowledge represented in a higher level. This knowledge is stored in a knowledge database using a context model.

Based on a previous study of context modeling requirements [11], we decided to rely on semantic modeling in order to represent context knowledge in an extensible context model. This context model is based on a multi-level ontology that provides a vocabulary for representing knowledge and describing context information. This multi-level ontology consists in an upper level, defining general context information, and a lower level, with specific context information. The advantage of using such multi-level ontology is its extensibility. This semantic modeling allows the knowledge sharing and its reuse. Besides, the use of ontology for formalizing context information enables the use of powerful logic reasoning mechanisms [11].

2) Intentional Query Processor

The *Intentional Query processor* (IQP) is in charge of processing user's request. Such request represents user's intention, expressed by a *verb*, a *target* and a set of optional *parameters,* according to a specific template [7][17]. The IQP enriches this request with context obtained from context model. This enriched request, represented in XML format, is then transferred to the discovery module.

3) Service Invoker

When a service is discovered and selected by the *discovery manager*, the URI of this selected service is sent to the *service invoker*, which is in charge of invoking and executing it.

4) Discovery Manager

Discovery manager (DM) is in charge of discovering most suitable services, according to the user's context and intention. This process is based on a semantic description of services and on a matching algorithm detailed in section IV.

The service discovery process is launched when the IQP sends the enriched request to the DM. Then, DM loads the semantic description of the available services and launches the matching process on all the available services.

IV. Context-Aware Intentional Service Discovery

In order to enhance the transparency and efficiency of PIS, we propose a new mechanism for service discovery based on our service description enriched with intentional and contextual information [10].

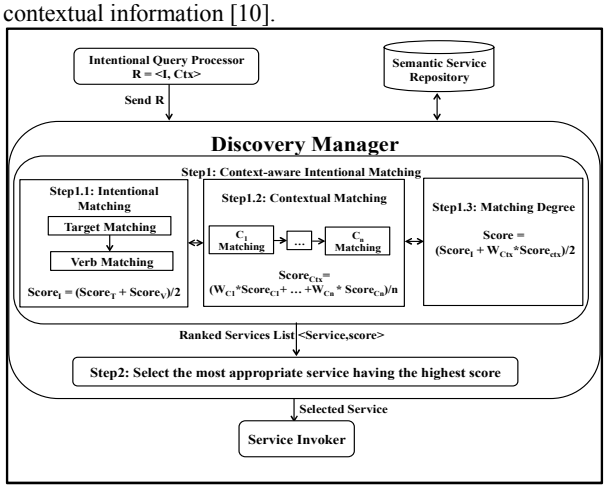


Figure 2. The context-aware intentional service discovery process

The intention concept is used to expose services and to implement a user centric vision of the PIS in a given context. We propose a context-aware intentional service discovery based on a semantic matching algorithm. This matching is a two-steps process illustrated in Figure 2.

First of all, we match the user's intention with the intention that the service satisfies (step 1.1). Second, we match the services context conditions with the user's current context (step 1.2). Finally, we calculate the matching degree between the user's request and the provided service, and we include the service with its obtained score in a list (step 1.3). These steps, detailed in section IV.B, are done for all the available services. Then, from the resulted list, we select the service having the highest score (step 2).

A. Enriched service description

The service description used in the service discovery process is based on our extension of OWL-S, which includes the information concerning both context and intention that characterize a service [10].

Actually, we consider that a user requires a service because he has an intention that the service is supposed to satisfy. Such intention emerges on a given context, which also characterize the service. We enriched OWL-S service description with both intention and context description [10]. Intention is described by adding a new sub-ontology, which refers the intention that a service can satisfy. Context is described by a context element pointing out an external file, which allows service provider to easily update context information related to the service description itself. More details about our extension of OWL-S are presented in [10].

B. Context-aware intentional service discovery algorithm

The service discovery algorithm that we propose in this paper performs a semantic matching process in order to offers the most appropriate service to the user. Once all the available services have been loaded, the matching algorithm may proceed. The goal of this matching algorithm is to rank the available services based on their contextual and intentional information and select the most suitable one for the user. This algorithm compares semantically user's intention with the intention that the service satisfies and user's current context with the service's context conditions.

Algorithm 1 Service Discovery

```
1:  Procedure SERVICEDISCOVERY (C_U, I_U, S)
2:  S_ranked = ø
3:  I^score = 0
4:  C^score = 0
5:  S^score = 0
6:  for ∀s ∈ S do
7:      I^score = IntentionMatching(I_U, I_S)
8:      if I^score > k then
9:          C^score = ContextMatching (C_U, C_S)
10:         S^score = (I^score + (w_i * C^score))/2
11:     end if
12: if S^score > l then
13:     S_ranked.add (s, S^score)
14: end if
15: end for
16: return S_ranked
17: end procedure
```

Figure 3. The context-aware intentional service discovery algorithm

The Figure 3 presents the proposed discovery algorithm. For all the available services, we calculate the matching score S^{score} between the user's request and the service (line 6-11). First, we calculate the intention matching score I^{score} between the user's intention I_U and the service intention I_S (line 7). As we mentioned above, the intention expresses the user's requirement that he wants to be satisfied by the system. It is composed of two mandatory elements: *verb* (V) and *target* (T). The *verb* exposes the action allowing the realization of the intention. Then, the *target* represents either the *object* existing before the satisfaction of the intention or the *result* created by the action allowing the realization of the verb [7][17]. Thus, to define the matching degree between user's intention I_U and service's intention I_S, we calculate: (1) the matching degree between user and service targets (respectively T_U and T_S); (2) the matching degree between user and service verbs (V_U and V_S), and finally (3) the intention score representing the sum of the verb and target matching score. Once intention matching is performed, we proceed with context match, in which we calculate score C^{score} from matching between user's current context and contextual conditions established by service description (line 9). Both scores (I^{score} and C^{score}) are then used to calculate service final score S^{score} (line10). Both matching are detailed in next sections.

1) Intention matching

The intention matching is based on the use of *ontologies*, *semantic matching* and *degree of similarity*. Besides, regarding to the intention formulation, the intention matching is especially based on a *verb* and *target* matching. According to this, we use an *ontology of verb* and a *domain-specific ontology* representing the possible targets in a specific domain. The degree of similarity represents a distance calculated based on the semantic link between two concepts in the ontologies. Thus, the intention matching is calculated based on two relations, *TargetMatch* and *VerbMatch*, used to define the *IntentionMatch* between user's intention $I_U = <V_u, T_u>$ and service's intention $I_S = <V_S, T_S>$ as follows:

$$IntentionMatch(I_U, I_S) = \begin{array}{c} \forall V_U, \exists V_S : VerbMatch(V_U, V_S) \\ And \\ \forall T_U, \exists T_S : TargetMatch(T_U, T_S) \end{array}$$

The target matching relation compares concepts defined in a domain-specific ontology, from required targets T_U and provided target T_S. The evaluation of how well a required target matches a provided target is typically based on the subsumption hierarchy used to find which a provided concept can fulfill a requested concept. In order to perform such matching, our algorithm is based on the matching algorithm proposed by [15], using the following 4 levels:

- ***Exact***: the required concept is equivalent to the provided concept
- ***Plug-In***: the provided concept subsumes the required concept
- ***Subsume***: the required concept subsumes the provided concept
- ***Fail***: there is no subsumption between the two concepts.

Thus, the target matching score is calculated based on the function *getScoreTarget* that takes as input the domain-specific ontology, the user's target and the service's target. The output of this function represents the degree of similarity between these targets calculated based on the distance between them in the domain-specific ontology. The score of the target matching is calculated as illustrates TABLE I.

TABLE I. TARGET MATCHING SCORE

Matching relation	Distance	Score
Exact	0	**1**
Fail	-1	**0**
Plug-in / Subsume	d	**1 / (d+1)**

Then, if the target score is greater than a fixed threshold we start the verb matching step, else we can conclude that the corresponding intentions are not similar and consequently the corresponding service can not satisfy the user's intention.

The verb matching relation is based on a verb ontology, which contains a domain-specific set of verbs, their different meanings and relations. Each verb relation associates a verb with related general verbs, more specific verbs and verbs sharing a common meaning. Thus, we propose to categorize verb matches according to five level of matching:

- ***Exact*:** the required verb is equivalent to the provided verb;
- ***Synonym*:** the required verb shares a common meaning with the provided verb;
- ***Hyponym*:** relationship of subordination between the required verb and the provided verb with a more general sense;
- ***Hypernym*:** relationship of subordination between the required verb and the provided verb with a more specific sense;
- ***Fail***: there is no relation between the verbs.

Such levels are based on a relation *Property (P, C_1, C_2)*, in which C_1 is a required concept, C_2 is a provided concept and P is the related property between C_1 and C_2. For example, *Reserve.hasSynonym = Book*. This relation is defined as follows:

$$Property(P, C_1, C_2) = \forall C_1, C_2, \exists P : C_1.P = C_2$$

Thus, the verb matching score is calculated based on the function *getScoreVerb* that takes as input the verb ontology, the user's verb and the service's verb. The output of this function represents the score of the verb matching degree. This score is calculated based on the relation between the two verbs. First, this function determines the relation property between them. Then, the score of the verb matching is calculated as illustrates TABLE II.

TABLE II. VERB MATCHING SCORE

Relation Property	Score
Exact	**1**
Synonym	**0,9**
Hyponym	**0,7**
Hypernym	**0,5**
Fail	**0**

The final score of the intention matching is represented as the sum of the verb matching score and the target matching score. Thus, if the intention matching score is greater than a given threshold, the second step representing the contextual matching is triggered (line 9).

2) Context matching

The context description for a user (C_U) or a service (C_S) represents a set of observable context elements, in which $C_U=\{c_j\}_{j>0}$ and $C_S= \{c_i\}_{i>0}$. Each context element is described by an *entity* (to which the context element refers), an *attribute* (that characterizes the property that we observe) and a set of observed *values*. In order to define the relation *ContextMatch*, we consider the relation *ContextElementMatch* that matches individually the different context elements constituting the user's and service context descriptions C_U and C_S. This relation is presented as follows:

$$ContextMatch(C_S, C_U) = \forall c_i \in C_S, \exists c_j \in C_U : ContextElementMatch(c_i, c_j)$$

The context element match proceeds as follow: for each c_i and c_j, we (i) match the c_i.entity with the c_j.entity; if the matching score between them is higher than a given threshold then we (ii) match the c_i.attribut with the c_j.attribut; if the matching score between them is higher than a given threshold then we (iii) match the different values one by one.

The observable context elements can be divided into several types. Thus, the context might be represented as a multi-dimensional space. Context information values are distinguished between numerical or non-numerical types. In order to take into account this diversity and the multi-dimensional space representing the context, the relation *ContextElementMatch* identifies the nature of the context element value and accordingly triggers the suitable matching function to compare them. This relation evaluate if the user's context element satisfies the service's context element conditions, based on a specific operator (equal, not-equal, between, higher-than, lower-than).

For example, we have as a service's context condition that the device bandwidth must be higher than 12500. From the user's current context description, if the captured value of the user's device bandwidth is really higher than 12500, then we return an exact match. Thus, the context matching score C^{score} (line 9) is calculated as the sum of the scores of each context element, and represented as follows:

$$C^{score} = \sum_{i=1}^{n} w_i f(type, c_i, c_j)$$

The *weight* that the user allocates to each context attribute and whose value is between 0 and 1, represents the importance of an attribute to a given entity. The purpose here is to highlight the real importance of a context attribute according to user's preferences, and the importance of the attribute is proportional to its weight.

V. SERVICE DISCOVERY MECHANISM

A. Overview

Service discovery algorithm proposed in section IV was implemented using Java language, based on a basic service discovery mechanism [18]. Such basic mechanism is organized around three main interfaces: ServiceManager, PersistenceManager and SearchEngine. The ServiceManager represents the entry point to the application and offers a set of methods allowing ontology management and service discovery and selection. PersistenceFacade interface acts as a façade between the PersistenceManager and the database to access service and ontology descriptions, handling ontologies and service descriptions. SearchEngine is in charge of searching an appropriate service for a given request. It uses a MatcherFacade interface that acts as a facade between the SearchEngine and the API to operate

service descriptions. This *MatcherFacade* allow easily extend discovery mechanism by adding new discovery algorithms. Next sections detail the extensions we made on this basic service discovery mechanism.

B. Service Description Implementation

The context aware intentional service discovery is based on an extension of OWL-S that we presented in a previous work [10] This extension includes the intention that a service satisfy and context conditions in which a service is valid. Thus, based on the OWL-S API Mindswap [13], we develop in Java the OWL-SIC API, which implements the service description according to his intention and context. In order to evaluate the proposed implementation, the service retrieval test collection OWLS-TC2 [14] is used. We chose this test collection because it provides a large number of services from several domains, test queries and relevant ontologies, in spite of containing only basic service descriptions based on OWL-S. The collection is intended to support the evaluation of the performance of OWL-S service matching algorithms.

We extend service descriptions proposed by OWLS-TC2 in order to include on those services intentional and contextual description. These service descriptions are used for evaluating proposed context-aware intentional service discovery mechanism in order to search the most interesting service according to user request and context.

C. Service Discovery Implementation

In our architecture, the *ServiceManager* interface is in charge of supplying the search methods for client applications. Every search method shall propose a list of suitable services with their matching scores and then return the best-ranked service. It decides, based on the calculated scores, the service that should fits the request. This search service is based on a *Matchmaker* interface representing a service matching algorithm. In our implementation, the matchmaker is easily replaceable in order to support multiple discovery processes. The selection of what matchmaker should be launched through a simple properties file.

Two main implementations of matchmaker have been proposed. First, a reference-matching algorithm called *BasicMatchmaker* [18], based on to the input and the output information given by the user. The *BasicMatchmaker* searches and selects the services according to this information [18]. As a second implementation, we propose the context-aware intentional service matching algorithm presented in this paper. This matcher, called *ContextIntentionMatchmaker,* uses OWL-S extended API [10], Jena [6] and the reasoner Pellet [16]. It calculates, for each service in the knowledge base, a score based on the intention and the context from the user's request and from the service description. This matchmaker is based on two classes *ContextMatching* and *IntentionMatching*, which are in charge of calculating respectively the context and the intention scores. By separating score computation, it is possible to easily disable one of these classes in order to evaluate separately the impact of context or intention matching on the core. In order to evaluate the validity of our context-aware service discovery algorithm, we compare the results of these two matchmakers. The experimental results of such evaluation are presented in the next section.

VI. EVALUATION

As mentioned earlier in this paper, we generate a semantic repository containing a set of extended service descriptions based on the extended OWLS-TC2. Among the provided domains, we choose to evaluate our proposed service discovery process in the Travel domain. It represents about 200 service descriptions that we enriched with intentional and contextual information. The evaluation has been performed in Grid'5000 platform [5], which contains a set of heterogeneous clusters, as indicated in TABLE III

TABLE III. GRID'5000 CLUSTERS USED IN THE EXPERIMENTS

Cluster	CPU	Processor	Cores	RAM
Stremi (2011)	2	AMD 1.7 GHz	12	48
Graphene (2010)	1	Intel 2.53 GHz	4	16
Griffon (2009)	2	Intel 2.5 GHz	4	16
Capricorne (2006)	2	AMD 2 GHz	1	2
Bordeplage (2007)	2	Intel 3 GHz	1	2
Bordereau (2007)	2	AMD 2.6 GHz	2	4
Borderline (2007)	4	AMD 2.6 GHz	2	32

The purpose of our experiments is to evaluate the validity of our algorithm and the feasibility of our extended service descriptor. Two main observations emerge from this experiment: (i) *Scalability*: Whether the processing time is reasonable; (ii) *Result quality*: whether the algorithm can effectively select the most appropriate services.

A. Scalability

We measure the scalability of our service discovery algorithm with respect to the number of services and the capabilities of the nodes from different clusters, by measuring the average processing time. The Figure 4 illustrates the experimental results of processing time consumed when running our service discovery algorithm.

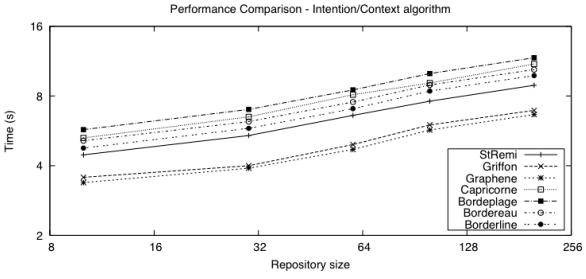


Figure 4. Intention/Context service discovery performance

The results demonstrate that the response time of the service discovery algorithm follows a logarithmic trend line, which allows confirming that the implemented mechanism has a good scalability. Then, we can notice that the use of 200 tested services only for the evaluation, which can be not

significantly large, stills sufficient to demonstrate the logarithmic behavior of our algorithm. Besides, we notice the greater impact of the processor family and the generation of the clusters. Nevertheless, we believe that these performances can be improved by parallelizing tasks on multi-core architectures.

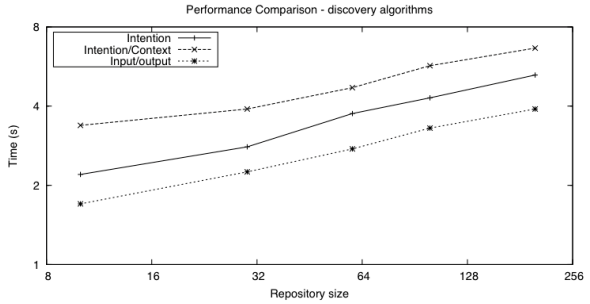


Figure 5. Service discovery performance comparison

The Figure 5 shows a comparison between three types of service discovery: (i) an Input/Output service discovery (*BasicMatchmaker*), (ii) an Intentional service discovery (*IntentionMatchmaker* with the context matching score disabled) and (iii) the Context-aware Intentional service discovery. For these three algorithms named, respectively, ***IO***, ***I*** and ***IC***, we measured response time using the same set of services. For the matter of clarity, we present the comparison of these 3 algorithms on the *Graphene* cluster. This cluster was chosen because its characteristics are closest to current off-the-shelf desktop/workstation, contrarily to the other clusters that are tailored for an HPC environment or have obsolete configurations.

The graph illustrates that the IC algorithm takes longer to process services. However, this difference on execution time does not represent a serious time difference from a user perspective. Actually, despite the fact that the input/output service discovery goes faster for selecting services, the response time of our algorithm still represents a reasonable processing time for selecting the most appropriate service.

This demonstrates the feasibility of our proposition on machines usually used as service repository, despite the fact that the implementation is not optimized to multi-cores architecture. Furthermore, we must consider that the IO algorithm uses a very simplified mechanism having a low cost but, as demonstrates the next section, presents a result quality considerably lower than the IC algorithm.

B. Result Quality

In order to measure the quality of the result, we cover the two most useful quality metrics: *precision* and *recall*. These two measures are defined in terms of a set of retrieved item and a set of relevant items. The *precision* represents how well a system retrieves only the relevant services, while the *recall* measures the ability of a system to retrieve all the relevant service [21]. The definition of recall and precision measures are defined as follows:

$$\textit{Precision} = \frac{|\{\textit{relevant items}\} \cap \{\textit{retrived items}\}|}{|\{\textit{retrieved items}\}|} \qquad \textit{Recall} = \frac{|\{\textit{relevant items}\} \cap \{\textit{retrived items}\}|}{|\{\textit{relevant items}\}|}$$

In order to evaluate these two measures, we formulate five user's requests relatives to the travel domain. These request are represented by the user's intention and his current context, as illustrates TABLE IV. In this scenario, the user is looking for surfing or hiking sport. Thus, he searches a destination where he can practice such sports. Then he wants to reserves a hotel or a Bed-and-Breakfast for the period. As a first step, we choose to evaluate our proposed service discovery algorithm using a simple scenario taking into account different or similar intentions in different contexts. In fact, this scenario represents a simple one, but it is sufficient to illustrate our approach.

TABLE IV. USER'S INTENTION IN A GIVEN CONTEXT

Intention	Context
Reserve Hotel	- Age >=18
Reserve BedAndBreakfast	- Age >= 18 - Season = Summer
Locate Sport Destination	-Age >= 18 - Season = Summer - City = Germany
Search Surfing Destination	-Age >= 18 -Surfing-Level=Beginner -Season=Summer -Weather=notDisturbed
Search Hiking	-Age >= 18 - Hiking Level=Confirmed -Weather=not disturbed -Health = Good

Through the experiments, we observe that the precision and recall are interesting factors when considering the intention and context in the service discovery. The result presented in Figure 6 shows that we obtained the highest precision percentage compared to IO algorithm. We obtain about 99% of precision, while the IO algorithm obtains about 50% of precision. This indicates that our service discovery algorithm has a greater chance to retrieve the most appropriate service according to user's intention and context.

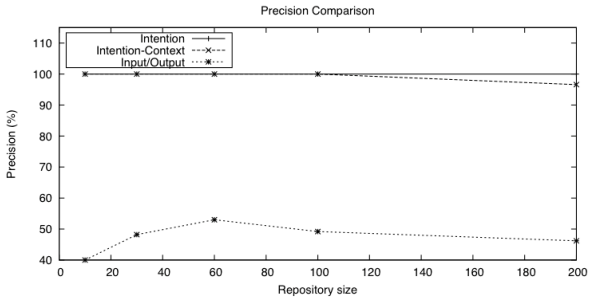


Figure 6. Precision Results

Besides, the results illustrated in Figure 7 show that we obtain an interesting recall about 95% compared to the IO algorithm, which reaches about 93.2%. However, the results obtained by the IO algorithm are circumstantial since this algorithm is not able to select a service adapted neither to user's context nor to his intention. In fact, the IO algorithm can only return all the services related to the request with a high rate of "false-positive" (indicated by its *precision*). This demonstrates that our IC algorithm is able to find all or almost services that can fulfill user's intention in a given context, with the lowest rate of "false-positive". These properties are parts of parameters, which contribute in the quality of the user's experience.

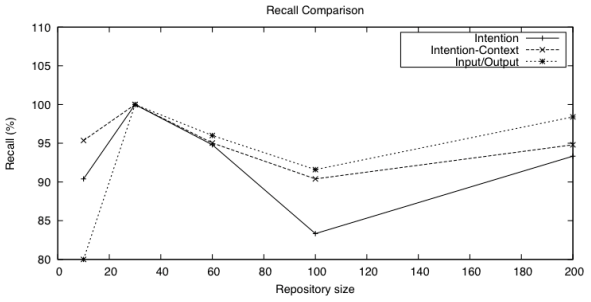


Figure 7. Recall Results

The analysis of these results, illustrated in Figure 6 and Figure 7, demonstrates that our proposition presents a more interesting result with a higher quality. We believe that our service discovery mechanism allows selecting the most appropriate service. And that is due to both its intentional approach, which is more transparent to users, and contextual approach that limits services to those that are valid.

VII. CONCLUSION

With the development of pervasive technologies, information systems become pervasive (PIS). Unfortunately, PIS provides several implementations of services that still too complex for user, who just requests a service satisfying his need. In order to enhance the transparency, we should select to the user the service that satisfies his need, without having to learn more details about the implementation or the constraints of used devices.

In this paper, we propose a user-centric vision of PIS that considers both user's intention and context. This vision allows us to obtain a more precise service discovery offering the most suitable service for the user. Thus, we implement a service discovery mechanism guided by user's intention and context. This service discovery mechanism is evaluated according to two aspects. The first one is the *scalability,* which allows analysing the feasibility of the proposed mechanism especially with respect to the growing of the service repository. The second one is the *result quality* based on two measures in services field called *precision* and *recall*. These two measures are used to analyse whether our mechanism really reached its goal. The experiments demonstrate that our algorithm is able to find, in a reasonable time, all or almost services that can fulfill user's intention in a given context, with the lowest rate of "false-positive".

Future works will be concentrated on the improvement of our experiments results. We expect to evaluate our service discovery mechanism in a more interesting real world scenario. Besides, these experiments will be tested on a more important number of services. Moreover, we opt to improve the implementation of the service discovery algorithm in order to reduce the processing response time. As a next step, our efforts will be focused also on the service prediction mechanism. Given the large amount of existing services and user's needs, our purpose is to help users by offering them, without their demand, personalized services that can interest them.

ACKNOWLEDGEMENTS

Experiments presented in this paper were carried out on Grid'5000 experimental testbed (*https://www.grid5000.fr*).

REFERENCES

[1] K. Aljoumaa, S. Assar, and C. Souveyet, "Reformulating User's Queries for Intentional Services Discovery Using an Ontology-Based Approach," 4th IFIP Int. Conf on New Technologies, Mobility and Security (NTMS), Paris, France, pp. 1-4, 2011.

[2] S. Ben Mokhtar, D. Preuveneers, N. Georgantas, V. Issarny, and Y. Berbers, "EASY: Efficient semAntic Service discoverY in pervasive computing environments with QoS and context support," Journal of Systems and Software 81(5), pp.785-808, 2008.

[3] L.O. Bonino da Silva Santos, G. Guizzardi, L.F. Pires, and M. Van Sinderen, "From User Goals to Service Discovery and Composition," ER Workshops, pp. 265-274, 2009.

[4] A. Dey, "Understanding and using context," Personal and Ubiquitous Computing, Vol. 5 n°1, pp. 4-7, 2001.

[5] Grid5000, https://www.grid5000.fr/ , 2011.

[6] Jena Semantic Web Framework, http://jena.sourceforge.net/.

[7] R.S. Kaabi, and C. Souveyet, "Capturing intentional services with business process maps," RCIS, pp. 309-318, 2007.

[8] P.E. Kouruthanassis, and G.M. Giaglis, "A design theory for pervasive systems," IWUC, pp. 62-70, 2006.

[9] I. Mirbel, and P. Crescenzo, "From end-user's requirements to Web services retrieval: a semantic and intention-driven approach," J.-H. Morin, J. Ralyte, M. Snene, "Exploring service science," First Int Conf, IESS, Springer, pp. 30-44, 2010.

[10] S. Najar, M. Kirsch-Pinheiro, and C. Souveyet, "The influence of context on intentional service," 5th Int. IEEE Workshop on Requirements Engineerings for Services (REFS'11) - IEEE Conference on Computers, Software, and Applications (COMPSAC'11), Munich, Germany, pp. 470 – 475, 2011.

[11] S. Najar, O. Saidani, M. Kirsch-Pinheiro, C. Souveyet, and S. Nurcan, "Semantic representation of context models: a framework for analyzing and understanding," J. M. Gomez-Perez, P. Haase, M. Tilly, and P. Warren (Eds), 1st Workshop on Context, information and ontologies CIAO, European Semantic Web Conference ESWC'2009, ACM, pp. 1-10, 2009

[12] T. Olsson, M.Y. Chong, B. Bjurling, and B. Ohlman, "Goal Refinement for Automated Service Discovery," 3rd Int. Conf on Advanced Service Computing, Service Computation, Rome, Italy, pp. 46-51, 2011.

[13] OWL-S API: http://www.mindswap.org/2004/owl-s/api/.

[14] OWLS-TC: http://www.semwebcentral.org/projects/owls-tc/ .

[15] M. Paolucci, T. Kawmura, T. Payne, and K. Sycara," Semantic matching of web services capabilities," in: Proceedings of the First International Semantic Web Conference, Springer LNCS 2342, Sardinia, Italy, 2002.

[16] Pellet, http://www.mindswap.org/2003/pellet/ index.shtml.

[17] C. Rolland, M. Kirsch-Pinheiro, C. Souveyet, "An Intentional Approach to Service Engineering," IEEE Transactions on Service Computing, Vol.3 n°4, pp. 292-305, 2010.

[18] S. Schulthess, "Construction of a registry for searching web service," Master Thesis, EFREI, Engineering School Paris, 2011.

[19] A. Toninelli, A. Corradi, and R. Montanari, "Semantic-based discovery to support mobile context-aware service access," Computer Communications, vol.31 n° 5, pp. 935-949, 2008.

[20] M. Weiser, "The computer for the 21st Century, " Sci Am 265(3), pp. 94–104, 1991.

[21] H. Xiao, Y. Zou, J. Ng, and L. Nigul, "An Approach for Context-aware Service Discovery and Recommendation," IEEE Int. Conf on Web Services (ICWS), Miami, pp.163-170, 2010.

Annex IX

Paper

Najar, S. ; Kirsch-Pinheiro, M. & Souveyet, C., “A context-aware intentional service prediction mechanism in PIS”, In: David De Roure, Bhavani Thuraisingham & Jia Zhang (Eds.), *IEEE 21st International Conference on Web Services (ICWS 2014)*, 27 June - 2 July 2014, Anchorage, Alaska, USA, IEEE CS, pp. 662-669. DOI : 10.1109/ICWS.2014.97

2014 IEEE International Conference on Web Services

A context-aware intentional service prediction mechanism in PIS

Salma Najar, Manuele Kirsch Pinheiro, Carine Souveyet
Centre de Recherche en Informatique - University Paris1 Panthéon-Sorbonne
90 rue de Tolbiac, 75013 Paris – France
Salma.Najar@malix.univ-paris1.fr,
{Manuele.Kirsch-Pinheiro, Carine.Souveyet}@univ-paris1.fr

***Abstract*—Pervasive Information System (PIS) represents a new generation of Information Systems (IS) available anytime, anywhere in a pervasive environment. In this paper, we propose to enhance PIS transparency and efficiency through a context-aware intentional service prediction approach. This approach allows anticipating user's future needs, offering and recommending him the most suitable service in a transparent and discrete way. We detail in this paper our service prediction mechanism and present encouraging experimental results demonstrating our proposition.**

Keywords-pervasive information system; service orientation; intention; context-aware; prediction; clustering; classification

I. INTRODUCTION

Pervasive Information Systems (PIS) represent an evolution of Information Systems (IS) towards the integration of pervasive environments. Currently, pervasive environments are mainly reactive. Decisions are taken solely based on the current context. Indeed, research in the anticipatory and proactive behavior on pervasive environment, notably by the prediction of the user's future situation, is hardly done. By avoiding focusing on the prediction, current systems lack an important element in the search for transparency and homogeneity. Besides, this research does not consider the user's intentions behind each service request. Consequently, many opportunities can be offered to the user, even if he/she is not always able to understand what is proposed to him and why. Thus, PIS remains too complex for the users, who are just interested in satisfying their needs (and not on how it is done).

We believe that in order to achieve transparency necessary to handle such pervasive environments, the PIS must reduce the user's understanding effort. They must hide the complexity of the multiple available services. This will be possible thanks to a user-centered vision. This vision can be achieved through a service prediction mechanism capable of anticipating future user's needs, improving system proactivity, and thereby contributing to improving the transparency necessary for PIS.

Our purpose is to predict the user's future intention based on his/her context, in order to offer him the most suitable service considering his/her incoming needs. This approach considers PIS through the notion of *intention* that can be seen as the goal that we want to achieve without saying how to perform it [10]. An intention represents a requirement that a user wants to be satisfied without really care about how to perform it or what service allows him to do so. An intention emerges on a giving context. The *context* represents any information that can be used to characterize the situation of an entity [5].

Based on this information, we propose a context-aware intentional service prediction mechanism. The main purpose of such approach is to provide to the user a service that can fulfill his/her needs in a fairly understandable and non-intrusive way, reducing user's understanding effort. This prediction mechanism is based on the assumption that, even in a dynamic and frequently changing Pervasive Information System, common situations can be found. Based on this assumption, this prediction mechanism considers a set of time series representing observed user's situations. Thus, we are able to track and store these situations in a history, after each successful discovery process. By analyzing this history, a prediction mechanism can learn user's behavior in a dynamic environment, and therefore deduce his/her future intention and the most appropriate service that satisfy it.

This paper is organized as follows: the *section II* introduces our vision of an user-centered contextual PIS, while *section III* details the proposed context-aware intentional service prediction mechanism. The *section IV* presents the implementation and the experiments results of the service prediction. The *section V* presents an overview on related works. Finally, we conclude in the *section VI.*

II. A USER-CENTERED CONTEXTUAL VISION OF PIS

In this paper, we introduce our new vision of Pervasive Information Systems [11] [15]. This user-centered vision of PIS allows focusing on the user's needs through an intentional approach. It considers the PIS and their elements both in terms of IS and of pervasive service systems, observing their control, intentionality and context-awareness requirements. This is in order to ensure the necessary transparency and understanding for the design and for the development of PIS.

This vision is characterized by its *context orientation,* which allows a better management of the heterogeneity and dynamics that characterize the pervasive environment. Moreover, in the perspective to better satisfy user's needs and to be at her/his level, our vision is based on an *intention orientation*. Thus, PIS can, on the one hand, better

978-1-4799-5054-6/14 $31.00 © 2014 IEEE
DOI 10.1109/ICWS.2014.97

understand the user's needs, and on the other hand, better meet her/his needs in the most appropriate manner.

Thus, we exploit, in our vision, the close relationship between the notions of intention, context and service, shown in Figure 1. We consider that the satisfaction of the user's intention in a PIS depends on the context in which this user is. For us, the environment directly impacts how to meet the intentions, and so the choice of services to be performed. Therefore, by combining intentional and contextual approaches in service orientation, we propose a new user-centered vision of transparent and non-intrusive PIS that is understandable to the user.

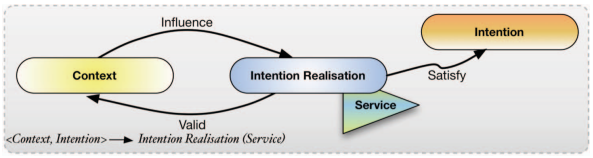


Figure 1. The close relation between context, intention and service

In this paper, we present a context-aware intentional prediction mechanism, presented in the next section, as a part of this vision. This is in order to enhance the PIS transparency, efficiency and proactivity through a user-centered contextual vision of PIS, hiding technical details.

III. CONTEXT-AWARE INTENTIONAL SERVICES PREDICTION MECHANISM

In this paper, we propose a new an approach for predicting the future user's intention (I_U) in a given context (C_{XU}). This approach intends for proactively provide a service (S_{Vi}) that can fulfill the user's future needs. Indeed, this approach is based on the assumption that common situations (S) can be detected, even in a dynamic and frequently changing Pervasive Information System. Based on this assumption, this prediction mechanism considers a set of time series representing a user's observed situation. These observations, represented by the triplet <intention, context, service>, are time stamped and stored in a database after each services discovery process. Thus, by analyzing this history ($\mathcal{H}$), the prediction mechanism can learn the user's behavior model ($\mathcal{M}c$) in a dynamic environment and thus deduce its next intention.

Two main processes compose this intention prediction mechanism: the *learning process* and *the prediction process*, as illustrated in Figure 2. In the learning process, similar situations (S) are grouped into clusters, during the phase of clustering. These clusters are organized as states of a state machine, by the classification phase. It aims at representing, from the recognized clusters, the user's behavior model ($\mathcal{M}c$) based the observed clusters. By interpreting situation changes as a trajectory of states, we can anticipate his/her future needs. Therefore, the intention prediction process is based on the user's behavior model ($\mathcal{M}c$), on the current user's intention (I_U) and the current user's context (C_{XU}). Based on this information, the prediction process allows predicting the user's future needs.

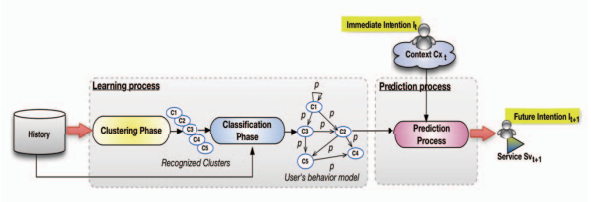


Figure 2. Service Prediction Mechanism

Before detailing these processes, we should describe the structure of the history used by these processes. This represents the trace management, described in next section.

A. Trace Management

The history $\mathcal{H}$ is composed by all the results obtained by the service discovery process [15], thereby forming the traces used for prediction. The service discovery process compares the current user's intention (I_U) and context (C_{XU}) with those proposed by the available services, in order to propose user the most appropriate service (S_{Vi}). We define the notion of situation (S_i) as the user's intention (I_U), in a given context (C_{XU}), satisfied by a specific service (S_{Vi}).

$$S_i = \{< I_{Ui}, C_{XUi}, S_{Vi} > \mid \forall\, i \in [1, n],\ I_{Ui}, C_{XUi}, S_{Vi} \in H \wedge TimeStamp\,(I_{Ui}, C_{XUi}, S_{Vi}) = ti\} \quad (1)$$

The prediction mechanism is based not only on the current user's situation, but also on its previously observed situations. These observations are saved for future needs. We refer to time series of observed situations as the user's history ($\mathcal{H}$). Each time series represents a time stamped observed situation, as illustrate the Table I.

TABLE I. THE STRUCTURE OF THE USER'S HISTORY

Time/Date	Intention	Context	Service
t_1	I_{U1}	C_{XU1}	S_{V1}
t_2	I_{U2}	C_{XU2}	S_{V2}
…	..	..	..
t_i	I_{Ui}	C_{XUi}	S_{Vi}
…	..	..	..
t_n	I_{Un}	C_{XU}	S_{Vn}

Whenever a service is selected, the user's situation is registered on the user's history in order to keep a trace of the user's past situations. The intention (I_U) is represented as an XML schema containing two mandatory elements, namely the verb and the target. Both verb and target are described by ontologies representing respectively significant actions made available by PIS and the objects considered by the actions [14]. The context (C_{XU}) is also represented as an XML schema containing the context description. Such descriptions follows an ontology-based context model on which context elements are described by an entity (corresponding to the entities whose context is observed) and a scope representing what is observed (location, memory, etc.) [14]. Finally, the service (S_{Vi}) represents the name of the service selected to satisfy this intention in this context. Services are described using an extension of OWL-S we have proposed in [14], in

which intention satisfied by the service and the context in which this intention is considered are described.

In the history, the traces represent user's situations (S_i) recorded at a given time. We introduce the notion of observation (O_{Si}), representing a situation of the user S_i observed at the time t_i.

$$O_{Si} = \{<S_i, t_i> \mid \forall i \in [1, n],\ S_i \in \mathcal{H} \wedge TimeStamp\ (S_i) = t_i\} \quad (2)$$

Then, we define the history $\mathcal{H}$ as a set of all the observed situations O_{Si} ordered according to their time of occurrence.

$$\mathcal{H} = \{O_{Si}\},\ i \in [1,n],\ with\ n\ the\ history\ size \quad (3)$$

Thus, maintaining the trace of the user's observed situations helps the learning process in order to deduce the user's behavior model. This learning process will be explained in the following section.

B. Learning Process

To realize anticipatory and proactive behavior of PIS, we need first to dynamically learn about the user and his/her behavior in a frequently changing environment. This represents an important step for the prediction mechanism.

The learning process is based on the analysis of the history ($\mathcal{H}$). We proceed by grouping the different observed situations (O_{Si}) into clusters ($\mathcal{CL}$) of similar situations and then, learn the user's behavior model. It is responsible for dynamically determining the user's behavior model ($\mathcal{M}c$) (*classification*), which illustrates the user's habits, from the recognized clusters (*clustering*). It is with that this process is triggered independently of the prediction process, and may be seen as a background task.

1) Clustering

The first phase of our prediction mechanism is the *clustering* of user's traces. As the user interacts daily with PIS, some of his/her situations may be recurrent. These recurring situations can be expressed with similar contexts and intentions. The role of *clustering* here is to consider the relevance of these situations, which can be grouped by similarities in clusters. A *cluster* represents then a set of situations, sharing some similarity between their intentions and contexts. It gathers recurring and similar situations. The cluster analysis allows a better representation of the user's habits, because they are more relevant to treat than separate situations.

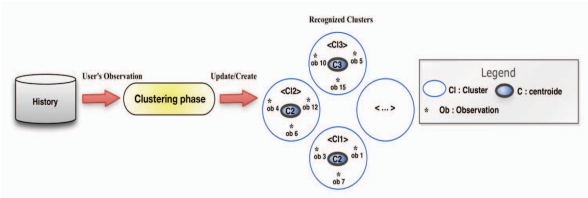


Figure 3. The clustering phase

The input of this phase corresponds to vectors representing user's situations stored in the history (Table I) The main role of clustering, as shown in Figure 3, is to detect recurrent observations among all situations previously observed and grouped in a cluster. A *cluster* consists of a *centroid* and a set of *observations*. The *centroid* represents the identifier of the cluster which symbolizes the observation the most similar to all the observations grouped in this cluster. The *centroid* is defined by the triplet $<I_{\mathcal{CL}}, C\chi_{\mathcal{CL}}, Sv_{\mathcal{CL}}>$.

In fact, the clustering is responsible for determining the situation that is the closest to a set of situations corresponding to highly similar intentions in quite similar context. This provides us with a powerful mechanism to evaluate the user's intention. A user can express the same intention in a slightly different way by using verbs and targets that are semantically similar enough. Based on verb and target ontologies, we perform a semantic matching between two intentions in order to determine their degree of similarity. On the other hand, the user's context represents highly heterogeneous data. Thus, to compare two context descriptions, we combine a semantic matching between the context elements (scope and entity should be semantically similar) and similarity measures that compare the values of context element. Therefore, the clustering will help to find these situations and represent them by one common situation that is closest to all the members of the same cluster.

In order to fully represent a situation, we attach the selected service for the couple <*Intention*, *Context*>. We are aware that this represents a strong constraint (the concept situation is necessarily coupled to a particular service), but it opens a significant performance advantage, since it is not required to launch the service discovery mechanism during the prediction process. Thus, it is important to regularly update the clusters in order to have the service that best meets the couple intention and context of the situation.

Once the clustering is completed, recognized clusters are then interpreted as states of the user's behavior model. This is the classification phase, presented in the next section.

2) Classification

A user's behavior model intends to reflect the interaction between a user and the PIS and its dynamics. Nonetheless, the user cannot be accurately described in advance. Therefore, a dynamic user's behavior model is necessary. It must be able to adapt to user's change and take into account the probabilistic nature of his behavior.

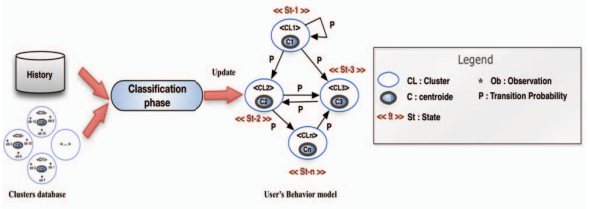


Figure 4. The classification phase

From the clusters recognized in the clustering phase and the history, the classification step determines and maintains a user's behavior model, as illustrated in Figure 4. This phase represents the user's behavior as a set of states with a transition probability. Each *state* is represented by the centroid of a cluster. Each *probability* is calculated based on the history and determines the probability of moving from

one state to another (i.e. the probability that a situation belonging to a cluster A will be followed by a situation belonging to a cluster B).

Several classification techniques exist: Bayesian network (BN) [7], Markov Chain [6], Hidden Markov Model (HMM) [19], Support Vector Machines (SVM) [3], etc. Similar to [12][20], we consider Markov chains [6] as the more suitable method for context classification thanks to its unsupervised and online characteristic. Moreover, Markov chains are able to classify multidimensional and heterogeneous data, which is necessary for classifying intention-context cluster as these are proposed on this paper.

Therefore, Markov [6] chains are the most suitable candidates for PIS. It is a well-known method for representing a stochastic process in discrete time with discrete state space. We represent the Markov chains model ($\mathcal{M}c$) as the doublet $\mathcal{M}c = (\mathcal{S}t, p)$, with $\mathcal{S}t$ representing the different states and $p \in [0,1]$ the transition probability.

At a given time t, the user is in a state $\mathcal{S}t_i$. In PIS, the user's intention and his context may change. Therefore, the user moves from the state $\mathcal{S}t_i$ to $\mathcal{S}t_j$. The state $\mathcal{S}t_j$ represents the successor of $\mathcal{S}t_i$ with a certain probability p. This transition probability represents the ratio of the transition from $\mathcal{S}t_i$ to $\mathcal{S}t_j$ divided by the number of all the possible transitions from $\mathcal{S}t_i$. This probability is represented in (4).

$$p_{St_iSt_j} = P(X_{t+1} = St_j | X_t = St_i) = \frac{N_{St_iSt_j}}{N_{St_iSt_k}} \quad (4)$$

The prediction process, described in the next section, is mainly based on the results of the classification to predict the next user's intention.

C. Prediction Process

The purpose of this prediction process is to predict the future user's intention in order to propose him the next service. This way, the user would not have to actively request it. This process is triggered when a service discovery process is performed successfully.

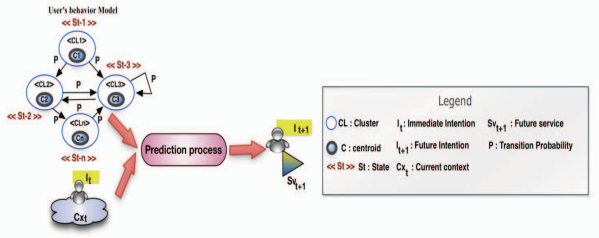


Figure 5. The prediction process

As illustrated by Figure 5, the services prediction process is based on the user's behavior model ($\mathcal{M}c$), which is updated during the previous phase of *classification*. The prediction process is, then, responsible to find the state ($\mathcal{S}t_i$) from the model ($\mathcal{M}c$), which is the closest (i.e. semantically similar) of the current user's situation, and to deduce the following situation, which is the most probable. More specifically, and as shown in Figure 6, this process compares semantically, for each cluster represented as a state in the model $\mathcal{M}c$, the immediate intention of the user ($I_{\mathcal{U}}$) with the intention of the state (centroid of a cluster) (I_{St_i}). Then, it compares semantically the current user's context ($Cx_{\mathcal{U}}$) with the context of the state (Cx_{St_i}). If the final score (degree of the contextual and intentional matching) is acceptable (above a certain threshold), then the state is selected as a candidate. Once this processing is done on the set of states in the model $\mathcal{M}c$, and then the state having the highest score is retained. The future service to be offered to the user represents the service of the state, which is held at the end.

Algorithm 1 Context-aware Intentional Service Prediction

```
1.  Procedure ServicePrediction (I_U, Cx_U, Mc)
2.  Result = ∅, State_ranked = ∅
3.  State_Observed = ∅
4.  State_Successor = ∅
5.  Sv_state = ∅, I_state = ∅, Cx_state = ∅
6.  Score = 0
7.  ID = GetStatesID (Mc)
8.  For each id ∈ ID do
9.      Score = Match (I_U, Cx_U, id)
10.     State_ranked.add(id, score)
11. End for
12. If State_ranked is not empty Then
13.     State_Observed = MaxScore(S_ranked)
14.     State_Successor = FindNextState(State_Observed, Mc)
15.     Sv_state = GetService (State_Successor)
16.     I_state = GetIntention (State_Successor)
17.     Cx_state = GetContext (State_Successor)
18.     Result.add(Sv_state, I_state, Cx_state)
19. End If
20. Return Result
21. End procedure
```

Figure 6. The service prediction algorithm

The Figure 6 details the proposed algorithm for predicting the future user's intention and consequently the most appropriate next service. The line 9 of Figure 6 shows the first step of the prediction process. It illustrates the semantic matching between the intention and context of each state of the model with the user's immediate intention and context. First, this step is based on a semantic matching between the user's intentions and the intention of the state. As mentioned above, an intention consists of a verb and a target. The semantic matching of intentions is therefore based on ontologies describing these elements in order to calculate the matching score between them. Then, the algorithm performs a semantic matching between the user's context description and the context descriptions of the different states of the model. This matching is based on a domain-specific ontology and on similarity measures between the values of context (see [15] for more details).

The final matching score represents the sum of the intention matching score and the context matching score. This information is stored with the state identifier. Going through all the states of the model, we can determine the state the most similar to the current user's situation (line 13). If a state is identified, the algorithm select the next state

based on the transition probabilities (line 14). This transition probability must exceed a certain threshold. If several successor states are retrieved, then the one having the highest transition probability is chosen (if there is more than one next state having the highest transition probability, then we choose arbitrary one of them). By this choice, we derive the successor state, which represents the future user's intention in a given context. Thus, we anticipate the user's future needs by offering him the most appropriate service that can interest him.

IV. IMPLEMENTATION AND EVALUATION

We present in this section the implementation of our context-aware intentional service prediction. Then, we discuss our experimental results.

A. Implementation

The services prediction mechanism, proposed in section IV, is based on a history that contain user's traces, the recognized clusters and the user's behavior model. We feed this database by a set of observations stored as traces. These observations represent a set of intentions that the services of the test collection OWLS-TC2 [16] are able to satisfy in the field of travel. Then, we add to these intentions, different contextual descriptions and the service identifier that can meet this intention in the context of use. We identified 10 different context descriptions. Then, we assigned arbitrarily this context description to the defined intentions. These traces were created fictitiously because it was difficult to convince companies to provide us their real data (respect and protection of privacy). The recognized clusters and the user's behavior model are maintained from this history.

This services prediction mechanism was implemented using Java language. It follows the same implementation structure like the service discovery mechanism [15], where the implementations of the processes are organized around three main implementations of interfaces. The *Manager* interface (*IclusteringManager, IClassificationManager* and *IservicePredictionManager*) represents the entry point of the processes. Then, the *PersistenceManager* interface (only used in the clustering and prediction process) acts as a facade between the component that implements the *manager* interface and the ontology directory, which allows access and loading ontologies. Finally, the *Engine* interface (*IclusteringEngine, IclusteringEngine* and *Iprediction Engine*) proposes the necessary methods to implement the different proposed algorithms, such as (i) the classification of the recognized clusters, (ii) the clustering of the user's situations; and (iii) the prediction of the user's future intention and the most appropriate service.

The clustering and prediction algorithms implementations are based on our OWL-S extended API [14], Jena [9] and the reasoner Pellet [18]. These impelentations use two classes, namely *"ContextMatching"* to determine the contextual matching score and the *"IntentionMatching"* class to determine the intentional matching score.

B. Evaluation

As part of our experiments, we deployed our algorithms on a machine Intel Core i5 1.3 GHz with 4 GB memory. As mentioned earlier in this paper, the evaluation of these algorithms has been performed on a semantic directory containing a set of domain-based ontologies and on the constructed history.

The purpose of our experiments is to evaluate the validity of our algorithms and their feasibility. Two main observations emerge from this experiment: (i) *Scalability*: whether the processing time is reasonable; (ii) *Result quality*: whether the algorithm can effectively select the most appropriate services.

In order to evaluate these two measures, we formulate 7 user's requests relatives to the travel domain. These request are represented by the user's intention and his current context, as illustrates in Table II.

TABLE II. USER REQUESTS : AN INTENTION EMERGED IN A GIVEN CONTEXT

Requête	Intention	Current Context
Req 1	- Reserve Five Star European Hotel	**Context User 1 :** -DateTime.Season = Winter, -Profile.Age = 21, -DateTime.Time = Morning, -Location.City = France -Resource.Screen = 16, -Device = Intel Core 2 duo -Resource.Memory = 2048, -Resource.Network = Ethernet
Req 2	- Book-up Lodge	**Context User 2 :** -DateTime.Season = Summer, -Profile.Age = 23 -DateTime.Time = Evening, -Location.City = Tanzania -Resource.Network = Wifi, -Device = Intel Core 2 duo -Resource.Memory = 1024, -Resource.Screen =15
Req 3	- Search BedAndBreakfast	**Context User 1**
Req 4	- Find Contact	**Context User 3** -DateTime.Season = Winter, -DateTime.Time = Night -Profile.Age = 16, -Profile.Expertise = Low -Profile.Role = Student, -Location.City = Mexique -Location.Country = USA, -Resource.Network = Wifi -Device = Iphone 4, -Resource.Memory = 16 -Resource.Screen = 3
Req 5	- Search Destination	**Context User 1**
Req 6	- Locate Surfing Destination	**Context User 4** -DateTime.Season = Summer, -Profile.Age = 24 -DateTime.Time = Morning, -Profile.Expertise = High -Profile.Role = Student, -Location.City = Hawai -Location.Country = USA, -Resource.Network = 3G -Device = Ipad 2, -Resource.Memory = 32 -Resource.Screen = 9
Req 7	Look for Surfing Course	**Context User 5** -DateTime.Season = Summer, -Location.City = Germany -DateTime.Time = Evening, -Profile.Expertise = High -Profile.Role = Student, -Device = Intel Core 2 duo -Profile.Age = 23, -Resource.Network = Ethernet, -Resource.Memory = 2048, -Resource.Screen = 16

These requests are formalized in different ways. We described situations where the elements of the user's intention are not described in the intention ontologies. Nevertheless, there are a set of clusters and states of the user's behavior model that can satisfy this intention in the current user's context (*Req2* and *Req7*). In addition, we described situations where the elements for the user's intention are described in the intention ontologies. These situations are specified in order to demonstrate that the threshold setting, defined by the system designer in the clustering and prediction algorithms, can eliminate some clusters and states even if they are able to satisfy the user's situation (*Req1*, *Req5* and *Req6*). And finally, we evaluated the requests *Req3* and *Req4*, which describe intentions that

can be satisfied by a set of clusters and states in the current user's context, which represents a complex context.

We measure the scalability of our algorithms with respect to the number of clusters, observations in the history and states of the user's behavior model, by measuring the average processing time.

1) Scalability

We measure the scalability of our algorithms with respect to the number of clusters, observations in the history and states of the user's behavior model, by measuring the average processing time.

The execution time of the clustering algorithm was measured by varying the number of clusters already recognized in the database between 7 and 186 clusters. This time represents the average execution time taken by the clustering algorithm to determine which cluster an observation belongs. As illustrated in Figure 7, the execution time following a polynomial trend of degree three varying from 3,6 *s* for 7 clusters to 6,3 *s* for 186 clusters. However, even if this time is a higher, we can observe that despite the fact that we have increased the number of clusters over twenty six times, the response time has only increased by a little more than one and half times. In addition, one of our perspectives is to improve the execution time by optimizing our development code by using the Java threads, for example.

Concerning the execution time of the classification algorithm, it is measured by varying the number of observations already grouped in clusters in the database between 10 and 200 observations. This time represents the average execution time taken by the classification algorithm to dynamically update the user's behavior model. As illustrated in Figure 7, the execution time follows a polynomial trend ranging from 39 *ms* for 10 observations to 398 *ms* for 200 observations. However, even if this algorithm does not take much time to dynamically update the Markov chains, we can notice that rose almost 10 times by increasing the number of observations of twenty times. Thus, introducing the parallel processing in the algorithm can also optimize this.

Finally, the execution time of the prediction algorithm is measured by varying the number of states in the user's behavior model, stored in the database, between 7 and 168 states. This time represents the average execution time set to predict the next service that satisfies a future user's intention according to his immediate intention and current context. As illustrated in Figure 7, the execution time following a polynomial trend of degree three from 2,04 *s* for 7 states to 5,28 *s* for 168 states. We increased the number of states over twenty five times, while the execution time has only increased by two and half times. This allows us to validate the feasibility and scalability of our algorithm. However, these results can be optimized, such as the last two algorithms of clustering and classification.

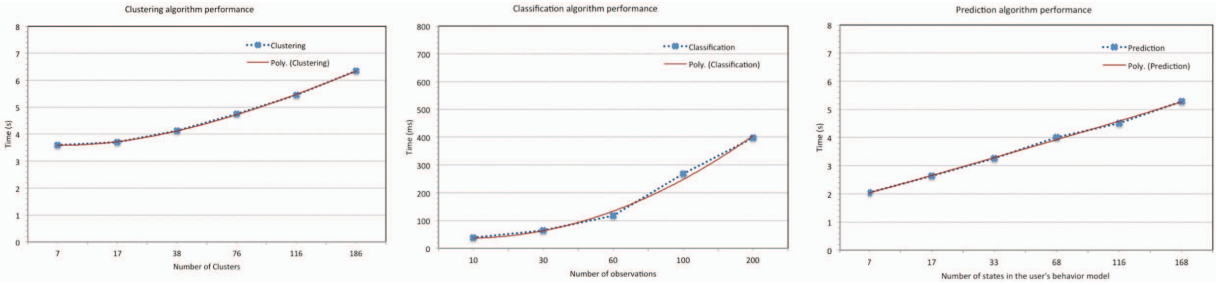


Figure 7. Clustering, classification and prediction algorithms performances

These results allow us to deduce that our algorithms provide a good scalability. However, the performance of our algorithms, especially the clustering and prediction is rather average. But, we remain confident since these algorithms can be optimized in order to improve the execution time. These optimizations are one of our short-term perspectives.

2) Result Quality

In order to measure the quality of the result, we cover the two most useful quality metrics: *precision* and *recall*. These two measures are defined in terms of a set of retrieved item and a set of relevant items. The *precision* represents how well a system retrieves only the relevant services, while the *recall* measures the ability of a system to retrieve all the relevant service [21]. The definition of recall and precision measures are defined as follows:

$$Precision = \frac{|(relevants\ items) \cap (retrived\ items)|}{(retrived\ items)}, Rappel = \frac{|(relevants\ items) \cap (retrived\ items)|}{(relevants\ items)}$$

We consider that the precision and recall metrics are important factors in the analysis of the learning (clustering and classification) and prediction mechanisms. The results presented in Figure 8 indicate that the three algorithms have a satisfying level of precision. The mean precision of the clustering and prediction algorithms is around 88%, whereas the classification algorithm is 100%.

These results indicate that the clustering algorithm is more likely to recognize the right cluster that represents the

user's observation. The classification algorithm represents exactly the dynamics of the user. The prediction algorithm, meanwhile, has a higher chance to select the most appropriate service that satisfy the future user's intention in a similar context to his current context. However, this good results of precision are accompanied by less interesting results (but still satisfaying) on recall, as illustrated in Figure 8. We observe that the average rate of the recall, for the clustering and prediction algorithms, is around 75%.

These results can be explained, for example, by the evaluation of situations where the elements of intention are not described in ontologies, while it exists in the database clusters/states similar to that intention in the user's context. In this case, clustering algorithms and prediction algorithms returns any results. This affects the results quality and return a recall of 0%. In addition, when situations are described by intentions whose verbs and/or targets are fairly generic or specific, some clusters/states that can respond to the trace or to the immediate user's intention in the current context will not be selected. Thus, in some cases we obtain reminders that are below 25%.

The analysis of these results, illustrated in Figure 8, demonstrates that our proposition presents a more interesting result with a higher quality. However, it is important to note that we cannot get that good results only if the system designer: 1) establishes a rich and complete description the different ontologies used; and 2) choose the best threshold setting. We believe that our service prediction mechanism allows selecting the most appropriate future service. And that is due to both its intentional approach, which is more transparent to users, and contextual approach that limits clusters and states to those that are valid.

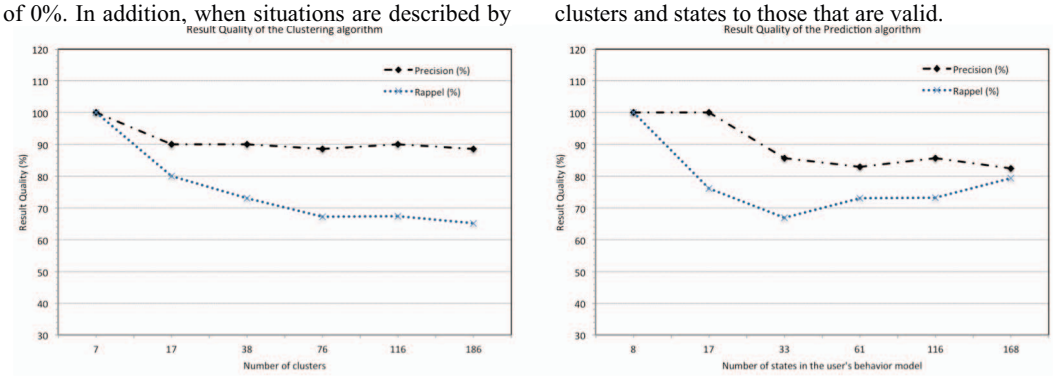


Figure 8. Quality of result of the Clustering and prediction algorithms

V. RELATED WORK

In order to supply users with the desired services, different research on *context prediction* and *context based recommendation systems* are proposed. Mayrhofer [12], Sigg et al. [20] and Meiners et al. [13], for example, propose major contributions towards generic context prediction. Mayrhofer [12] proposes an unsupervised classification, which tries to find previously unknown classes from input data. Sigg et al. [20] provide an alignment method based on typical pattern and on alignment technique in order to deduce missing low-level context information. Moreover, Meiners et al. [13] present a generic and structured context prediction approach based on (1) the incorporation of the domain knowledge at design time and (2) the selection of multiple exchangeable prediction techniques.

These context prediction approaches try to predict user's next context based on the user's current context and history. However, none of these works consider the services a user invokes on a given context, neither the real need behind it. Against these work, we focus first on the real user's needs and try to anticipate it by predicting his future intentions in similar context. This is in order to propose him the most suitable service that can interest him.

Concerning the second aspect, we have many contributions on recommendation systems, which aim to propose services based on the user's context. Adomavicius & Tuzhilin [1] and Panniello et al. [17] propose to categorize recommendation approaches into three categories: 1) *pre-filtering*, where the contextual information is used to filter out irrelevant ratings before they are used for computing recommendations; (2) *post-filtering*, where the contextual information is used after the standard non-contextual recommendation methods; and (3) *contextual modelling*, where the contextual information is used inside the recommendation algorithms with the user and item data.

For example, Baltrunas and Ricci [2] introduce a technique for context-aware collaborative filtering called "Item Splitting". In this approach, items experienced in two alternative contextual conditions are "split" into two items, which are then used in the rating prediction algorithm. Moreover, Cremonesi et al. [4] propose a technique that relies on classical and post-filters recommendation based on contextual information. This technique use association rules to identify the most significant correlations between context and item. These rules are then used to filter the predictions performed by traditional recommender systems. Recently, Hussein et al. [8] introduces a software framework for the development of context-aware and hybrid recommenders. They introduce the Hybreed framework based on Dynamic Contextualization approach.

All these works try to anticipate user's needs in order to offer him more transparency. Hence, most recommendation systems propose a next service to users based solely on their

context information, without considering the user's requirements behind a service, i.e., its intentions. They propose available services to the user, ignoring why this service is needed.

VI. CONCLUSION

Nowadays, our environment is characterized by the evolution of pervasive technologies. However, PIS that derived from using IS on pervasive environments still complex, requiring an important user's understanding effort.

Therefore, we propose a user-centered vision of PIS based on a context-aware intentional prediction approach, in order to hide PIS complexity. This approach allows us to anticipate the future user's needs. By this approach, we believe contributing to the improvement of PIS transparency and productivity through a user-centered view.

Thus, we propose an intentional prediction mechanism guided by the context. This prediction mechanism allows: (i) clustering similar user's situations in a set of clusters, (ii) learning the user's behavior model according to recognized clusters and user's history (iii) deducing the user's future intention based on his behavior model and on his current context and intention.

The, we implement the service prediction mechanism. This service prediction mechanism is evaluated according to two aspects. The first one is the *scalability,* which allows analysing the feasibility of the proposed mechanism especially with respect to the growing of the service repository. The second one is the *result quality* based on two measures in services field called *precision* and *recall*. These two measures are used to analyse whether our mechanism really reached its goal. The experiments demonstrate that our algorithms can be able to find, in a reasonable time, the most appropriate results that can fulfill user's intention in a given context, with the lowest rate of "false-positive". But, to in order to have a good result, the system designer should describe the different ontologies used in a rich and complete manner and choose the best threshold setting.

This intention prediction mechanism highlights the anticipatory and proactive behavior of our proposed vision of PIS. We strongly believe that an intentional prediction approach can answer to transparency and homogeneity requirements, necessary for fully acceptation of PIS. Moreover, evaluating the user acceptation of the proposal requires applying it in a real case study. Such evaluation should consider the final user's point of view. It should consider the user acceptance, considering the prediction mechanism, as well as the level of transparency perceived by these users. As a future work, we expect to evaluate our approach in a large-scale in order to validate its usefulness and compare it with the existing techniques.

REFERENCES

[1] G. Adomavicius, and A. Tuzhilin, "Context-aware recommender systems," In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, Recommender Systems Handbook, Springer, pp. 217-253, 2011

[2] L. Baltrunas, and F. Ricci, "Experimental evaluation of context-dependent collaborative filtering using item splitting,". User Modeling and User-Adapted Interaction: Special issue on Context-Aware Recommender Systems, 2013.

[3] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery," 2(2), pp. 121–167, 1998

[4] P. Cremonesi, P. Garza, E. Quintarelli and R. Turrin, "Top-N recommendations on Unpopular Items with Contextual Knowledge," In 3rd Workshop on Context-Aware Recommender Systems (CARS) at ACM Recommender Systems, Chicago, USA, 2011

[5] A. Dey, "Understanding and using context," Personal and Ubiquitous Computing, Vol. 5 n°1, pp. 4-7, 2001

[6] W. Feller, "An Introduction to Probability Theory and its Applications," New Jersey: Wiley, 1968

[7] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian Network Classifiers," Machine Learning, 29(2-3), pp. 131- 163, 1997

[8] T. Hussein, T. Linder, W. Gaulke, and J. Ziegler, "Hybreed: A software framework for developing context-aware hybrid recommender systems," User Modeling and User-Adapted Interaction, 24(1-2), pp.121-174, 2014

[9] Jena Semantic Web Framework, http://jena.sourceforge.net/

[10] R.S. Kaabi, and C. Souveyet, "Capturing intentional services with business process maps," 1st IEEE Int Conference on Research Challenges in Information Science (RCIS), pp. 309-318, 2007

[11] M. Kirsch Pinheiro, B. Le Grand, C. Souveyet, and S. Najar, "Espace de Services : Vers une formalisation des Systèmes d'Information Pervasifs," XXXIème Congrès INFORSID 2013: Informatique des Organisations et Systèmes d'Information et de Décision, Paris, 2013

[12] R. Mayrhofer, An Architecture for Context Prediction. PhD thesis, Johannes Kepler University of Linz, 2004

[13] M. Meiners, S. Zaplata, and W. Lamersdorf, "Structured context prediction: A generic approach". In R. Kapitza F. Eliassen (Ed.), Proceedings of the 10th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS), pp. 84–97 IFIP, 6: Springer, 2010

[14] S. Najar, M. Kirsch-Pinheiro, and C. Souveyet, "The influence of context on intentional service," 5th Int. IEEE Workshop on Requirements Engineerings for Services (REFS'11) - IEEE Conference on Computers, Software, and Applications (COMPSAC'11), Munich, Germany, pp. 470 – 475, 2011

[15] S. Najar, M. Kirsch Pinheiro, C. Souveyet, and A. Steffenel, "Service Discovery Mechanism for an Intentional Pervasive Information System," International Conference on Web Services (ICWS), Honolulu : United States, pp. 520-527, 2012

[16] OWLS-TC: http://www.semwebcentral.org/projects/owls-tc/

[17] U. Panniello, A. Tuzhilin, and M. Gorgoglione, "Comparing context-aware recommender systems in terms of accuracy and diversity: Which contextual modeling, pre-filtering and post-filtering methods perform the best," User Modeling and User-Adapted Interaction. Special issue on Context-Aware Recommender Systems, 2013

[18] Pellet, http://www.mindswap.org/2003/pellet/ index.shtml.

[19] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE, 77(2), pp. 257–286, 1989

[20] S. Sigg, S. Haseloff, and K. David, "An Alignment Approach for Context Prediction Tasks in UbiComp Environments," IEEE Pervasive Computing, 9(4), pp. 90-97, 2010

[21] H. Xiao, Y. Zou, J. Ng, and L. Nigul, "An Approach for Context-aware Service Discovery and Recommendation," IEEE Int. Conf on Web Services (ICWS), Miami, pp.163-170, 2010

Annex X

Paper

Najar, S.; Kirsch Pinheiro, M.; Le Grand, B. & Souveyet, C., “A user-centric vision of service-oriented Pervasive Information Systems”, *8th International Conference on Research Challenges in Information Science (RCIS 2014)*, IEEE, 2014, 359-370

A user-centric vision of service-oriented Pervasive Information Systems

Salma Najar, Manuele Kirsch Pinheiro, Bénédicte Le Grand, Carine Souveyet
Centre de Recherche en Informatique, Université Paris 1 – Panthéon Sorbonne
Paris, France
{First-Name.Last-Name }@univ-paris1.fr

***Abstract*—Information Systems (IS) have massively adopted service orientation by exposing their functionalities as services. With the evolution of mobile technologies (smartphones, 3G/4G networks, etc.), such systems are now confronted with a new pervasive environment for which they were not originally designed. Indeed, pervasive environments are characterized by their heterogeneity and dynamicity due to their evolving context and their need for transparency. None of these features are particularly considered in traditional IS designed for stable and controlled office environments. In our new vision for service-oriented Pervasive Information Systems (PIS), the user becomes the center of these systems. This paper presents a user-centric service-oriented vision for PIS based on a context-aware intentional approach, which considers the user intention and the context in which this intention arises as a guiding principle for service description, discovery, prediction and recommendation.**

Keywords—Service-Oriented Architecture; context-aware systems; user intention; Information Systems; Pervasive Computing.

I. INTRODUCTION

Weiser [42] proposed a new computing vision in which computers seamlessly integrate the environment. In this pervasive computing environment, users interact with computers in a transparent way: by having the machines fit the human environment instead of forcing humans to enter theirs [42]. We believe that such pervasive environments are already a reality through the seamless integration of multiple devices in our everyday life. According to Bell & Dourish [6], computation is embedded into the technology and practice of everyday life; we continually use computational devices without thinking of them as computational in any way. Indeed, we are continuously interacting with devices such as smartphones and tablets without any cognitive effort.

These new technologies (smartphones, RFID tags, wireless networks, etc.) have expanded the frontiers of Information Systems (IS) outside the enterprise. The BYOD (Bring Your Own Device) concept illustrates quite well this tendency: employees bring their own devices to the office and keep using them to access the IS even when they are on the move. The consequence of such technological evolution is that IS now have to cope with a pervasive environment, and in the future, they will have to integrate physical elements as well as logical and organizational ones. Indeed, pushed by the users, IT departments must make IS evolve towards these new trends (mobility, technology, transparency, etc.) to help them work efficiently anytime and everywhere. The shift of IS from traditional to pervasive is a specific trend requiring to find a trade-off between a centralized and controlled IT environment and a more dynamic and open environment adapting their support according to user environment.

A new generation of IS is then emerging, the *Pervasive Information System* (PIS). PIS intend to increase user productivity by making IS services available anytime and anywhere. Such systems shift the interaction paradigm from desktop computing to new technologies, evolving from a fully controlled environment (the office) to a dynamically pervasive one. Contrary to traditional IS, PIS have to support a multitude of heterogeneous device types that differ in terms of size and functionality (mobile phones, portable laptops, sensors and so on), by providing continuous interaction that moves computing from local presence to constant presence [19].

Designing Pervasive Information Systems is a challenge for which IT departments have no help. We argue that the user must be the center of this new generation of IS, since those systems should be designed for helping the user to better satisfy his/her goals according to the environment he/she belongs to. Moreover, new aspects characterizing PIS should now be considered: their need for *transparency*, as well as the *heterogeneity* and *dynamicity* of pervasive environments, in addition to the *goals* they must satisfy from IS point of view. We propose in this paper an innovative user-centric vision for PIS, based on a service-oriented context-aware intentional approach. Notice that the proposed approach assumes that a PIS is not built from scratch but that IS already exists as a collection of application services. The intentional and contextual layer is derived in a bottom-up fashion from the IS services.

The *intentional approach* allows us to consider services from a user requirements perspective, focusing on why a service is needed, and not only on how it is executed. Actually, we consider a service as a way to satisfy a user's intention in a given context [28]. An *intention* can be seen as the goal that we want to achieve without saying how to perform it [17]. It is formulated in a given *context* that can be defined as any information that can be used to characterize the situation of an entity (person, place, or object) considered as relevant to the interaction between a user and an application [11]. Combining these two aspects, it is possible to propose more relevant services to the user. By focusing on these aspects, the

transparency of the PIS is improved because, on the one hand, the user does not care about how it will fulfill the intention; on the other hand, an intention can be satisfied in many ways, especially considering the different contexts.

This innovative user-centric vision of PIS guided us into the definition of a new conceptual framework, named *Space of Services*, which can be applied to understand the main concepts of PIS without describing the way to implement them. Based on this conceptual point of view, we consider the user's point of view by analyzing the mechanisms necessary for offering the appropriate services. This results in a new functional point of view, which proposes mechanisms for *service discovery* and *prediction* according to user's intention and context. These mechanisms are incorporated in a suitable architecture. This architectural point of view considers system architecture aspects required for building and managing PIS according to this user-centric vision. Finally, in order to design new PIS based on this conceptual framework, a fourth point of view, focusing on the system designer's point of view is proposed. This support point of view provides a methodological guidance for IT management, guiding from the conceptual point of view to the architectural one.

The paper is organized as follows: first, we present the notion of Pervasive Information System, and review related works mainly in the area of pervasive systems. Then, we introduce our user-centric vision and its four points of views and describe each point of view's goals and components. Next, we present an evaluation of the presented mechanisms. The final section is dedicated to discussions and conclusion.

II. PERVASIVE INFORMATION SYSTEMS REQUIREMENTS

Pervasive Information Systems (PIS) have to cope with pervasive environments, without leaving behind the fact that they remain Information Systems (IS). PIS have to deal with heterogeneity that characterizes pervasive environments. In such environments, different kinds of devices co-exist and communicate with each other, forming a highly complex and dynamic environment. Such a rich environment offers new opportunities for services that could be integrated as part of the IS. Nevertheless, this complexity makes such environments difficult to understand by end users and IT management. This difficulty may limit user adoption of the system.

According to Dey [12], when users have difficulty forming a mental model about how applications work, they are less likely to adopt and use them. This is also true for PIS. Users need to understand PIS actions without necessarily understanding the technological environment around them. Transparency is then needed in order to hide this heterogeneity of devices, infrastructures and services. Such transparency is even more necessary because of the strategic position of IS in any company business. These systems are designed to help users reach business goals (*goal oriented*). Consequently, when using such systems, users must focus on their own tasks and not on the technology itself. Without transparency, any PIS will not be able to successfully fulfill its IS role.

Besides, according to Hagras [16], the dynamic and ad hoc nature of pervasive environments means that the environment has to adapt to changing operating conditions and changing user preferences and behaviors in order to enable more efficient and effective operation, while avoiding system failure. A context-aware approach helps deciding how to execute and adapt services in highly dynamic environments. PIS should supply users with the most appropriate service according to the user's current context. *Context-awareness* becomes then a key aspect for PIS, promoting system pro-action based on environmental stimuli [19]. By observing the user's context, it is possible to propose services that better cope with the real conditions in which they are invoked. Thus, PIS should provide context-aware capabilities in order to cope with dynamic changes of the environment and improve user efficiency.

Nevertheless, PIS must also behave like traditional IS, managing services according to both user and business goals. Indeed, IS are supposed to be built in order to fit business strategy and goals, and to allow users to accomplish their mission within this business strategy. PIS represents the next generation of IS and they must also cope with this IS role. Due to their strategic role, PIS cannot be designed as "normal" pervasive systems. PIS should be "controllable". In other words, they should be managed and controlled by company's IT management, since the inappropriate exposition of an internal service may have important consequences for the company's business. Thus, the unpredictable characteristic of the pervasive environment is not allowed in a PIS. Indeed, exploratory and opportunistic behaviors as those proposed by [18][34] cannot be fully accepted by IT management. They represent a risk for IS and what it represents for the companies.

Pervasive Information systems have to conciliate two completely different worlds. They must behave as pervasive systems, handling dynamic and *heterogeneous* environments. Nevertheless, they remain an Information System and as such, they must keep a *predictable* and expected behavior, despite such dynamic environments. From this analysis, a set of requirements applying to PIS can be delineated as follows:

- [R1] ***Heterogeneity***: PIS should handle heterogeneity of devices and services integrating pervasive environment.
- [R2] ***Transparency***: PIS should hide the heterogeneity and complexity of the pervasive environment that should become transparent to the users.
- [R3] ***Context-awareness***: PIS should be able to observe changes in the execution environment and adapt its behavior accordingly.
- [R4] ***Goal oriented***: PIS should be designed in order to satisfy user and business goals.
- [R5] ***Predictability***: PIS should be able to satisfy user's goals in a predictable and expected manner.

Unfortunately, the design of new PIS respecting these requirements remains an open challenge, although numerous researches on pervasive systems have proposed some insights concerning some of these requirements. The next section summarizes some of these related works.

III. RELATED WORKS

Context-awareness (R3) has become a key element for supporting pervasive environments. It can be defined as the

ability of a system to detect changes in the environment and to react to those changes, adapting its behavior in consequence [5][21][29]. During the last decade, a lot of research has been conducted on pervasive systems, mainly on context-aware services [8][9][21][24][39][41].

According to Eikerling et *al.* [13], context-awareness is necessary for providing adaptable services, for instance, when selecting the best service according to the relevant context or when adapting the service during its execution according to context changes. Such adaptation capabilities are often based on a semantic description of such services. Context-aware services can then be defined as services whose description is associated with contextual properties, *i.e.*, services whose description is enriched with context information indicating the situations to which the service is adapted to [41].

Different proposals for semantic description of context-aware services can be found in the literature [24][38][39][41]. Most of them are used for service discovery [25][39] or composition [25]. For [24][39], context is seen as a non-functional aspect of service. For others, such as [41], context is seen as condition for the execution of a service. In both cases, a semantic matchmaking is performed between context information related to the service and the one related to user or execution environment. For [24], this matchmaking is based on subsume and plugin relationships, while for [41], it is essentially based on similarity measures.

In most of these approaches, the user's context information is compared to the context information provided by service semantic description. The user is in the center of these approaches through the observation of the context. They enforce requirements R1 (*Heterogeneity*) and R3 (*Context-awareness*) mentioned above, but they fail on handling requirements R4 (*Goal oriented*) and R5 (*Predictability*) since goals behind user actions and requests are neglected.

Contrary to these approaches, intentional approaches such as [17][23][30] consider user's goals as a central aspect for service definition. For instance, [17][35] propose a methodological guidance for defining new services based on the intentions these services are supposed to satisfy. These authors assume that such an intentional-driven approach should avoid the current mismatch of languages between low-level service expressions such as WSDL statements and business perceived services [35]. Similarly, [17][23] also consider an intentional-driven process. They focus on service discovery, promoting a guiding process based on Web semantic technologies. This process intends to help users from an expert community to discover services responding to their intention.

Both [2][23] are also based on Web semantic technologies. [23] focus on user expressing intentional-based requests. Service descriptions on SAWSDL (Semantic Annotations for WSDL) are then enriched with semantic annotation describing intentional aspects of services, allowing a semantic matching between user's requests and those enriched services. [30] also associate service description with the intentions those services are supposed to fulfill. Similar to [23], they also consider the decomposition of intentions on refined low-level intentions. They advocate that such a refining process can be used to improve service discovery mechanism. WSML (Web Services Modeling Language) [15] also focus on a semantic description of user's goals, describing service capabilities, with their pre- and post-conditions, with the corresponding mediators necessary to reconcile requester and supplier representations.

Works such as [23][30][35] satisfy requirement R4 (*Goal oriented*), but they fail considering requirement R3 (*Context-awareness*), since they do not consider execution context. On the other hand, [36] consider both user's intentions and execution context on GSF framework (Goal-based Service Framework). User's intentions are associated to tasks, which are then associated with services. Context information is used only as input information during service discovery process.

In addition, [2][20] propose goal-oriented requirement engineering (RE) modeling approaches. These approaches use the notion of context in order to identify and model domain variability in goal models. The notion of context is restricted as a set of assertions, which may be integrated in the model concepts in order to express what part of the specification is available only under these conditions. The context exploitation at the requirement analysis allows defining the perimeter of the system to develop and its costs. They are not considered at all the execution phase, it is why the metamodel of the user context are not defined as well as its capture process with sensors and its evolution process are not taken into account. However, the relation <goal, context> remains useful during the execution phase as the "end-means impact", since the context can influence the choice of the activity to perform in order to achieve the goal of an actor. It is why in our proposition the service (activity) is described including the intention (goal) it supports and the context required to achieve the intention.

Most existing works are merely reactive. Decisions are taken in response to a user's request, and no anticipatory and proactive behavior is proposed. Current systems do not focus on the prediction of user's future situation, and therefore lack an important element in the search for transparency. In other terms, none of the previous works is able to fully satisfy R5 (*Predictability*) and R2 (*Transparency*) requirements.

An anticipatory behavior has been proposed by some works on context prediction [22][37][40] and on context-aware service recommendation [1][43]. Context prediction works try to anticipate user's next context [22][40] or fulfill missing context information [37]; while recommendation works try to proactively propose services to the user [1][43]. Both are based on the analysis of user's history in order to identify common patterns enabling the anticipation of user's next situation.

Even if these works provide a proactive behavior, they do not consider user's goals emerged from context situations or behind services requisitions. These works endorsed requirement R3 (*Context-awareness*) and R5 (*Predictability*), but they do not handle requirement R4 (*Goal oriented*).

We may observe that despite numerous works on service oriented pervasive systems, designing Pervasive Information Systems that fulfill requirements listed previously remains a complex task. In order to help IT management in this difficult task, we propose in this paper a new vision for Pervasive Information System as explained in the following section.

IV. A User-Centric Context-Aware Intentional Vision of PIS

Our innovative user-centric vision of PIS is based on a close relationship between the notions of *Intention*, *Context* and *Service*. This vision allows, on the one hand, to focus more on the user's real needs through an intentional approach, and on the other hand, to manage the heterogeneity and dynamics of the pervasive environment through a contextual approach. Indeed, we consider the PIS and their elements both in terms of IS and pervasive systems, observing their control, intentionality and context-awareness requirements. This is in order to ensure the necessary transparency and understanding for the design and the development of PIS.

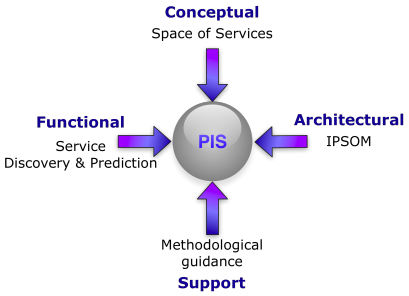


Fig. 1. Four points of views of a Pervasive Information System

This user-centric vision is decomposed into four complementary points of views, as represented in Fig. 1:

- The ***conceptual point of view*** proposes a conceptual framework, named *Space of Service*, intended to help IT management to better conceptualize such systems and its elements (*i.e.*, the service they offer and the observed context elements).
- The ***functional point of view*** supplies *service discovery* and *prediction* mechanisms using this dual intentional and context-aware approach.
- The ***system architecture point of view*** proposes a middleware named *IPSOM* that integrates previous mechanisms and represents the vision architecture.
- The ***support point of view*** provides a *methodology* guiding PIS design from the conceptual framework to the description of the proposed services within the system architecture.

Each point of view is detailed in the following sections.

A. Conceptual Point of view: Space of Services

The conceptual point of view focuses on understanding and defining Pervasive Information Systems (PIS). It aims at helping IT management better conceptualize such systems and their elements, notably the offered service and the observed context elements. In order to do so, we propose a conceptual framework, named *Space of Service*s. As illustrated by Fig. 2, the user interacts with the IS through a *Space of Services*, which defines the new PIS. Through this space, user interacts with services offered by the system and the user's context is observed by a set of sensors, in a transparent way.

Services are the central element of this framework. They represent the functionalities exposed by the PIS to the users, without defining how those will be implemented. Seeing PIS as service-oriented pervasive systems allows us to manage the heterogeneity of services that PIS may offer, which contributes to both **R1** (*Heterogeneity*) and **R2** (*Transparency*) requirements. Indeed, the nature of services proposed by a PIS can vary significantly, from traditional Web services to services integrated to the physical environment.

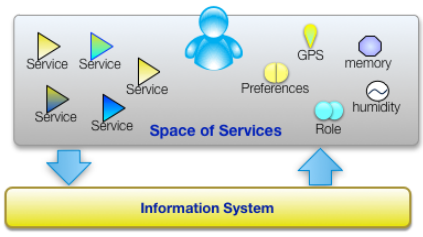


Fig. 2. Space of Services representation

Both services can be seen through the functionalities they offer rather than by the technologies used for their implementation, as stated by definition (1).

A service sv_i is characterized by a set of functionalities $\mathcal{F}$. Each functionality f_j is defined as a function of inputs in_j and outputs out_j expected by the service clients.

$$\mathcal{F} = \{ f_j (in_j , out_j) \} \quad (1)$$

Besides, we consider that a service offered by a PIS is proposed in order to satisfy a given user's goal, corresponding to user's needs. In other words, in order to fulfill requirement **R4** (Goal oriented), services should be associated to the intentions they allow users to satisfy, as stated in definition (2).

A service sv_i is proposed in order to satisfy a set of intentions $\mathcal{I}$. Each intention $I_t \in \mathcal{I}$ is defined by a verb v characterizing its action, a target tg over which action takes place and a set of optional parameters par.

$$I = \{< v , tg , par >\} \quad (2)$$

We believe that user intentions emerge in a given context, which should be observed in order to fully satisfy such an intention (**R3**). We advocate that an intention is meaningful when considering it in a given context. For us, an intention is not a simple coincidence. It emerges because a user is under a given context. As a consequence, a user does not require a service just because he is located in a given place or under a given context. He does require a service because he has an intention that a service can satisfy in this context.

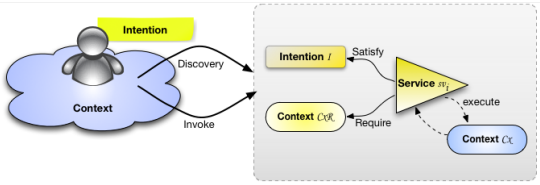


Fig. 3. Relationship among services, intentions, and context information

As illustrated by Fig. 3, a service sv_i belongs to a given context Cx (see definition (3)). This context indicates the conditions under which the service is executed by the provider.

It also characterizes the position of this service in the space of services ξ. Moreover, we consider that a service sv_i may have a required context CXR, which represents a set of contextual conditions under which the service is more likely to reach its goals. Therefore, the better the matching between the observed user context and the required context CXR is, the higher the chances of adapting it to the situation and of satisfying the user.

A service sv_i corresponds to a set of functionalities F provided by this entity sv_i in a context Cx in order to satisfy a set of intentions I. The satisfaction of these intentions depends on a favorable context, described as a required context CXR for the good operation of the service.

$$sv_i = <I, F, Cx, CXR> \quad (3)$$

Various models of context exist [7][29]. Despite their differences, some common key elements may be identified. We may therefore reduce a context model to the observation of one or several subjects (users, devices, etc.) for which a set of context elements is collected (location, activity, available memory, etc.). For each concept, the values associated to the metadata are captured (representation, quality indices, etc.). From these observations, we define the notion of observation made by a sensor, as presented in definition (4).

Each observation refers to the sensor cp_i for which a context element eo has been observed for the subject s_j. Each observation is thus a tuple composed of the subject s_j, the context element eo, and the value v observed at time t and described by the set of metadata M.

$O_{cpi} = \{ <ob_j, t_j> \}$, where $ob_j = <s_j, eo, v, M>$, in which (4)

- s_j is the observed subject;
- eo is an element of the context ontology;
- v is a value observed for this concept;
- t represents the instant when this observation is made;
- M is the set of metadata m and their value d

A sensor provides the IS and users with a set of contextual information that correspond to values observed in the environment. Sensors thus feed the PIS with contextual information it will use to adapt its service offer to users and their needs in the observed context.

These various types of sensors allow observing elements that characterize not only the physical environment (GPS, temperature, etc.), but also the logical environment (available device memory, user preferences, etc.) and the organizational environment (user role, execution state of a process, etc.). A sensor cp_i is defined by the set of its observations O_{cpi}, and also by its context Cx. The context itself is also described by a set of observations of context elements related to a given subject. This position is formalized in definition (5).

A sensor cp_i is defined by the set of its observations O_{cpi} and by the context Cx, also described by a set of observations.

$$cp_i = \{ O_{cpi}, Cx \} \quad (5)$$

Thanks to all elements identified above, we propose to formalize the *space of services* as a set of elements called entities, which surround the user in his/her physical, logical and organizational environments.

A space of services ξ is a pervasive environment composed of a set of entities e_i.

$$\xi = \{ e_i \mid e_i \in \mathcal{A} \vee e_i \in \mathcal{P} \}, \text{ in which} \quad (6)$$

- $\mathcal{A} \in \{sv_i\}$ is the set of active entities, which represent services,
- $\mathcal{P} \in \{cp_i\}$ is the set of passive entities, *i.e.*, available sensors in space ξ.

The *space of services* therefore comprises two types of entities: *active entities* ($\mathcal{A}$), capable of offering users one or several services, and *passive entities* ($\mathcal{P}$), which can inform users and the system of the environment. Active entities can have an action on the environment, whereas passive entities feed the PIS with information about the environment.

An entity e_i ($e_i \in \mathcal{A}$ or $e_i \in \mathcal{P}$) is characterized in the space of services ξ, by a context Cx, made of a set of observations. Each observation is related to the observed entity e_i and contains a value v for a context element eo observed at instant t, together with the set of meta-data M that characterize this observation.

$$Cx = \{ <ob_j, t_j> \}, ob_j = <e_i, eo, v, M> \quad (7)$$

The notion of *space of services* allows designers to better imagine the optimal, controlled and yet dynamic environments of the user and the system. This allows them to describe their PIS as multiple spaces of services, which are *permeable*, as illustrated in Fig. 4.

In other words, a *space of services* is not a closed space completely disconnected from the other spaces. On the contrary, it is a space that has no clear boundaries that prevent it from communicating with other spaces. It remains accessible. These spaces can share some common entities (see Fig. 4). Thus, the active or passive entities of a space of services may then exist in other spaces. In addition, the user evolves between these multiple spaces that overlap and change over time.

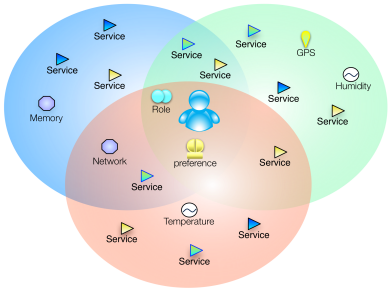


Fig. 4. Multiple spaces of services: Permeability

Thus, in order to allow the coexistence of the static and dynamic vision in a harmonious way, we consider the *state* of the space of services, in addition to its static definition, described above. In fact, a space of services may evolve over time, with appearing, disappearing or unavailable entities. The *state* of a space of services ξ at instant t, noted ξ^t, thus corresponds to active and passive entities actually available in the space ξ at that time. An entity e_i has therefore also a state at instant t. This entity state noted $e_i^{\xi,t}$, shows the availability of the entity in space ξ at instant t, as defined in (8).

The state of a space of services ξ at instant t, noted ξ^t, is the set of the states of entities e_i available in the space:

$$\xi^t \subseteq \xi, \xi^t = \{ e_i \mid e_i \in \xi \wedge e_i^{\xi,t} \} \quad (8)$$

where $e_i^{\xi,t}$ is the state of entity e_i at instant t.

B. *Functional Point of view: Service Discovery & Prediction*

Based on the space of service definitions, we could define a new semantic service description. We enrich the OWL-S (Semantic Markup for Web Ontology Language) service description in order to include information about the context and the intention that characterizes a service. More detailed explanation of this extension can be found in [27]. Nevertheless, proposing services based on the notions of intention and context is not enough. It is also necessary to propose to the user the appropriate services based on his/her current intention and context.

A service discovery mechanism based on these notions is then needed. Besides, in order to fulfill requirements R2 (*Transparency*) and R5 (*Predictability*), a proactive behavior is necessary. For more transparency, a PIS should be able to anticipate user's needs in a predictable way. A service prediction mechanism is then needed. Functional point of view considers these mechanisms for offering users the appropriate services considering their current and future needs and context.

1) Service Discovery

In our user-centric vision, we propose a *service discovery* mechanism in order to hide implementation complexity, and consequently to achieve the transparency promised by PIS. This service discovery, intends helping users discover the most appropriate service for them, *i.e.*, the service that satisfies the immediate user's intention in a given context. It is based on the semantic service description mentioned above [27] and on a semantic service discovery algorithm. The goal of this algorithm is to rank the available services based on their contextual and intentional information. It semantically compares the user's intention with the intention that the service satisfies and user's current context with the service's context conditions (corresponding to the required context ($C\mathcal{X}\mathcal{R}$). Then the service with the highest matching score is selected. It represents the most appropriate service that satisfies user's immediate intention in his/her current context.

More specifically, as illustrated in Fig. 5 the semantic matching algorithm is a two-step process: *intention matching* and *context matching*, presented in detail in [28].

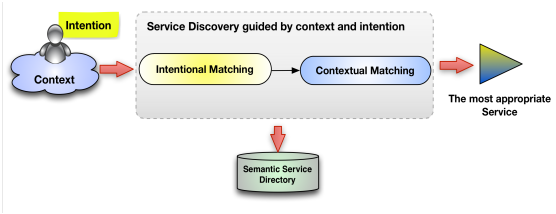


Fig. 5. Schematic view of the Services Discovery mechanism

In the first step, the *intention matching* is based on the use of ontologies and a semantic matching. *Intention matching* is calculated based on two relations, *TargetMatching* and *VerbMatching*, which are used to define the relation *IntentionMatching* between the user's required intention $I_{\mathcal{U}} = <\mathcal{V}_{\mathcal{U}}, \mathcal{T}_{\mathcal{U}}>$ and the service's intention $I_{svi} = <\mathcal{V}_{svi}, \mathcal{T}_{svi}>$.

For the *verb matching*, we use an ontology of verbs, which contains a domain-specific set of verbs, their different meanings and relations. The degree of similarity is calculated based on the existence of a semantic link between these two verbs in the verb ontology: (1/ L + 1), where L represents the number of links between two concepts in the ontology. We define five levels of similarity, inspired from the levels defined by Paolucci et *al.* [32], as illustrated in TABLE I.

TABLE I. VERB MATCHING RELATIONS

Matching Relation	Explanation	Link	Score
Exact	Required verb is equivalent to the provided verb	0	1
Synonym	Required verb share a common signification with the provided verb	-	0,9
Hyponym	Required verb is more specific than the provided one	L	1/(L+1)
Hypernym	Required verb is more generic than the provided one	L	1/(L+1)
Fail	No relation between the two verbs	-1	0

Similarly, for the *target matching*, we use a domain-specific ontology. This ontology represents the possible targets that are made available through the PIS. We compare the required target $\mathcal{T}_{\mathcal{U}}$ and the provided target $\mathcal{T}_{svi}$ using a degree of similarity also based on the distance between these concepts in the target ontology. This semantic similarity, based on [32], uses four levels: *exact*, *plugin*, *subsume* and *fails*. The *plugin* is similar to the *hyponym* in the verb matching, while the *subsume* is similar to the *hypernym*.

The second step, *i.e.*, the *context matching*, is based on a context ontology and a set of similarity measures. It matches individually, the different context elements constituting the user ($C\mathcal{X}_{\mathcal{U}}$) and service context descriptions ($C\mathcal{X}\mathcal{R}_{svi}$). The context description for a user ($C\mathcal{X}_{\mathcal{U}}$) represents a set of context observations $C\mathcal{X}_{\mathcal{U}}=\{cx_j\}\mid j>0$ and the context description for a service ($C\mathcal{X}\mathcal{R}_{svi}$) represents a set of context conditions $C\mathcal{X}\mathcal{R}_{svi} =\{cx_i\}\mid i>0$, both concerning subject and context elements.

The *context matching* score Cx_{score} is calculated as the sum of the scores of each context condition, as illustrated in (9)

$$\boldsymbol{Cx_{score}} = \sum_{i=1}^{n}(\boldsymbol{w} * \boldsymbol{ContextConditionMatching}\ (\boldsymbol{cx_i}, \boldsymbol{cx_j})) \quad (9)$$

Thus, in order to define the relation *ContextMatching*, we consider the relation *ContextConditionMatching* that matches individually the different context observations ($C\mathcal{X}_{\mathcal{U}}$) and context conditions ($C\mathcal{X}\mathcal{R}_{svi}$). The context matching proceeds as follows: for each observation *cxi* and *cxj*, we first match the corresponding subjects, using the context ontology. If the matching score is higher than a given threshold, then we match the corresponding context elements. This last matching also takes into account the weight assigned to it. The final score of the context element matching is equal to the weight assigned to it multiplied by the score of matching between them. Then, if the matching score between them is higher than a given threshold, only at this moment we evaluate the satisfaction of the context condition cxi.cd with respect to the user's context observations value, one by one. More details about the discovery mechanism are presented in Najar et al. [28].

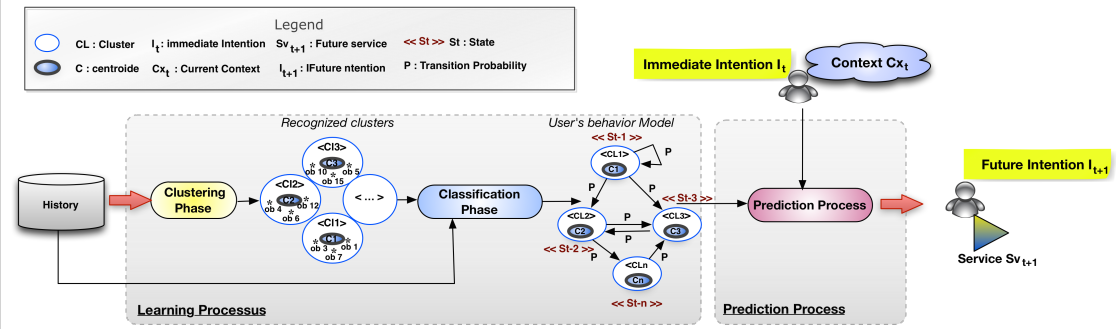


Fig. 6. Schematic view of the services prediction mechanism

Besides, the weight (w) that the user allocates to each context attribute (whose value is between 0 and 1) represents the importance of an attribute to a given entity. The purpose here is to highlight the real importance of a context attribute according to user's preferences, and the importance of the attribute is proportional to its weight.

2) Service Prediction

We propose an approach to predict the user's future intention. This approach recommends proactively a service that can fulfill user's future needs. It is based on the assumption that common situations (S_i) can be detected, even in a dynamically and frequently changing PIS. Based on this assumption, this prediction mechanism considers a set of time series representing the user's observed situation. We define the notion of situation (Si) as the user's intention (I_U), in a given context (CX_U), satisfied by a specific service (sv_i) resulting from a previous discovery process, as presented in (10).

$$S_i = < I_U, CX_U, sv_i > \quad (10)$$

These situations are time-stamped (*observations*) and stored in a database after each service discovery process. Thus, by analyzing the history ($\mathcal{H}$), the prediction mechanism can learn the user's behavior model ($\mathcal{M}c$) in a dynamic environment, and thus deduce its immediately coming intention.

Two main processes compose this intention prediction mechanism: the *learning* and the *prediction* processes, as illustrated in Fig. 6. Thus, to realize anticipatory and proactive behavior of PIS, we need first to dynamically learn about the user and his behavior in a frequently changing environment. This represents the *learning process* where similar situations are grouped into clusters, during the *clustering phase*. In the next step, these clusters are interpreted as states of a state machine. The transition probabilities from one state to another are then calculated based on the history. This step, called *classification phase*, aims at representing, from the recognized clusters, the user's behavior model ($\mathcal{M}c$) based on their situations (S_i). By interpreting situation changes as a trajectory of states, we can anticipate their future needs. This process consists in estimating the probabilities of moving from one situation to other possible future situations. Therefore, the *prediction process* is based on the user's behavior model ($\mathcal{M}c$), on the current user's intention (I_U) and on the current user's context (CX_U). Based on these, the prediction process allows predicting the user's future needs. It provides them a service that can meet their next needs in a fairly reasonable way.

The main task of the *clustering phase* is to detect recurrent situations (S_i) from all the observed situations before. It determines for a given situation, the closest set of situations corresponding to highly similar intentions in quite similar context. This provides us a powerful mechanism to evaluate the user's intention. Indeed, a user can express the same intention in a slightly different ways by using verbs and targets that are semantically similar enough. Based on verb and target ontologies, we perform a semantic matching between two intentions in order to determine their degree of similarity. The same applies to context information, since an intention may rise on similar contexts.

From the recognized clusters and the history, the *classification phase* determines and maintains a user's behavior model. This model represents the user's behavior as a set of states with a transition probability, representing the probability of moving from one state to another. The Markov chain [14] represents one of the well-known methods for representing a stochastic process in discrete time with discrete state space. We represent the Markov chains model ($\mathcal{M}c$) as the doublet $\mathcal{M}c = (St, p)$, with St representing the different states and $p \in [0,1]$, the probability of transition from one state to another.

In our case, at a given time t, the user is in a situation (state) St_i representing its intention in a given context. In a PIS, the intention of the user and their context may change. Therefore, the user moves from the situation St_i to the situation St_j. The situation St_j is the successor state of St_i with a certain probability p. This transition probability represents the ratio of the transition from St_i to St_j divided by the number of all the possible transitions from St_i. It is represented in (11).

$$p_{St_i St_j} = P(X_{t+1} = St_j | X_t = St_i) = \frac{N_{St_i St_j}}{N_{St_i St_k}} \quad (11)$$

The prediction process is based on the results of the classification to predict the user's next intention. Its purpose is to predict the user's future intention in order to propose to him/her the next service that can answer the future intention.

This prediction process is based on the semantic matching between the intention and context of each state of the model with the user's immediate intention and context. Similar to the discovery process, the semantic matching of intentions is based on ontologies describing these elements in order to calculate the score between them. Similarly, the matching between the user's context description and the context descriptions of the different states of the model is also based on a domain-specific

ontology and on similarity measures between the values of context (see [26] for more details on the different ontologies).

The final matching score represents the sum of the intention matching score and the context matching score. This information is stored with the state identifier. And by going through all the states of the model, we can determine the state that is the most similar to the current user's situation. Subsequently, if a state is identified, then the next state is selected based on the transition probabilities. This transition probability must exceed a certain threshold. If several successor states are retrieved, then the one having the highest transition probability is chosen. By this choice, we derive the successor state, which represents the user's future intention in a given context. We anticipate the user's future needs by offering them the most appropriate service that can interest them.

C. *Architectural Point of view: IPSOM Architecture*

Realizing the vision proposed by the space of services demands an architecture integrating its concepts and the functional point of view presented above. We propose, in this paper, an architecture named IPSOM (Intentional & Pervasive Service Oriented Middleware), whose goal is to allow the proposal of PIS that satisfies the requirements enounced earlier in this chapter. IPSOM, presented in Fig. 7, contains five main modules: context manager, intentional request processor, service discovery manager, learning manager, and service prediction manager. These modules are detailed below.

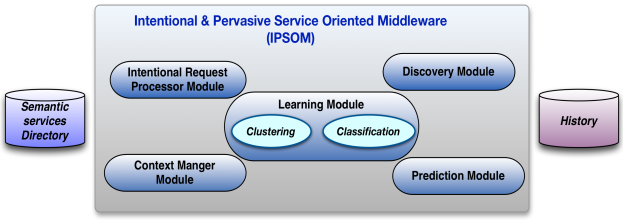


Fig. 7. Intentional & Pervasive Service Oriented Middleware Architecture

The purpose of the *context manager* (CM) is to provide a uniform way to access context information. The *Intentional Request Processor* (IRP) is in charge of processing user's request, expressed as an intention, and enriches it with current user's context. This enriched request is send to the *discovery manager* (DM) that is in charge of discovering most suitable services, by implementing the discovery mechanism as presented in section IV.B.1). Next, the learning manager (LM) is in charge of grouping the different observed situations into clusters of similar situations (clustering) and dynamically learns the user's behavior model (classification). This is used by the prediction manager (PM) that is in charge of the prediction mechanism (cf. section IV.B.2).

D. *Support Point of view: Methodological Guidance*

In order to support the creation of the space of services, we propose a methodological guidance for the conception of this space, intended for PIS designers. Our goal is to help them specify their system's expected features, together with the information that will be captured to guarantee adaptability. The challenge is to control the definition of the system and its services while taking into account a dynamic environment.

The proposed methodology combines bottom-up and top-down approaches, as shown on Fig. 8, and relies on both business and functional aspects in order to define the space of services and its business and technical components.

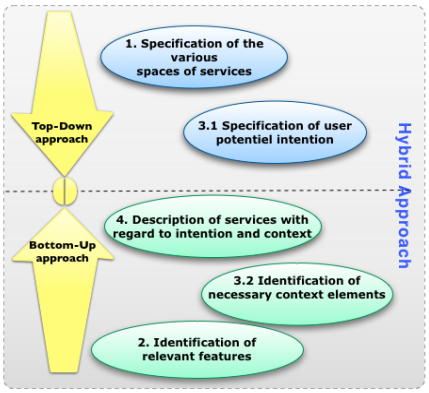


Fig. 8. Methodological guidance for PIS design

The design of a PIS with the notion of space of services starts with the definition of the multiple spaces that the users will be involved in. Each space is composed of active entities, corresponding to the services offered to users, and of passive entities (sensors), which allow observing the environment. Services are defined with regard to the intention they must satisfy and to the context of these intentions. Sensors are in charge of collecting contextual information required for system for adaptation purposes. They define boundaries for the notion of context by specifying relevant information. The goal is to take the context into account in order to provide users with well-chosen services.

Our methodology is divided into five steps, illustrated in Fig. 8. We illustrate these steps by an example, which represents the conception of the PIS for a mobile network company. In this company, workers are mobile, working at home, moving from client to client in order to sell and maintain their products. They move through multiple spaces with different devices, networks, configurations, profiles, etc. The five steps of our methodology are described below:

- *Specification of the various spaces of services*: This step analyzes and specifies the user's main workspaces. We consider here a mobile user, and the goal is to identify the conceptual spaces within which the user will interact with the PIS. The result of this first step is a list of spaces of services that are relevant for the user and required for the PIS. For example, in our network company, different category of users, with different roles (technician, sell man, etc.) and mobility behaviors can be observed. Analyzing these categories allow us identifying four spaces of services, representing the user when he/she is in the ***company***, at ***home***, at the ***customer*** and ***outside***.
- *Identification of relevant features*: This step identifies the various relevant functionalities of the PIS. This bottom-up approach starts from the existing IS, at the functional level, and identifies the functionalities that

should be exposed as services by the various spaces. For instance, considering our mobile network company, its existing IS offers features such as the *access to the customer view*, *the edition of sell proposals*, *the consultation of the commands*, etc. For each of these features, we should determine the corresponding available technical services. From these features, we have identified services such as ***Access Client View VPN_Service*** (consultation of the client view with VPN), ***View Command_Service*** (consultation of a specific command), ***Request Low Quality Audio Conference_Service*** (organization of a low-Level audio conference), etc.

- *Specification of potential user intention*: This step analyzes in depth user's potential needs in each space, in order to better understand the required PIS operation from the user's point of view. These needs are then specified as intentions. In the proposed example, we identify five relevant intentions: (1) consulting the client view; (2) editing a sell proposal; (3) consulting a command; (4) organizing a conference or a meeting; and (5) searching a certified restaurant. These intentions are described according to the Prat's model [33]. For example, the user's need *consulting the client view* is described as follow: *<intention> <verb> consult </verb> <target> <object> Client View </object> </target> </intention>*.

- *Identification of necessary context elements*: This step aims at identifying and modeling the various context elements that are relevant with regard to the services proposed in each space. The objective is also to spot available technologies to observe and capture necessary context elements. For this, we must identify relevant context subjects and elements that can characterize the space of services. For our mobile network company, we could identify two relevant context subjects: the ***user*** and the ***device***. For each of these context subjects, we identify the context elements that can be collected. For example, for the ***user*** we identify three static context elements, the user's ***age***, ***role*** and ***expertise***, and a dynamic one, which is the user's localization described as GPS coordinates. For the ***device*** subject, we identify four dynamic context elements, which are the ***type of the used device***, the ***memory***, the ***network type*** and the ***network name***. Finally, following the identification of these subjects and context elements, we are able to complete the multi-level context ontology.

- *Description of services with regard to intention and context*: This step consolidates the definition of each space of service. The services proposed in these spaces are described semantically by adding to their technical description the intentions that they must satisfy and their required execution context, as well as their execution context. A semantic description has thus been proposed as an OWL-S extension [27]. The resulting descriptions are then used for service discovery and execution as a function of user's intention and current context by the IPSOM platform. We identify, for our illustrative example, 17 services descriptions enriched with contextual and intentional information. TABLE II illustrates four of them.

At this step, we have determined all the intentional and contextual services. These services, with the intentions they meet, are well integrated into multiple spaces of services, reflecting the permeable nature of our notion of space services. For example, the service ***Sv1*** in Table II is integrated in two spaces of services: *home* and *outside*. The service ***Sv2***, which satisfy the same intention as the service ***Sv1*** but in a different required execution context, is integrated also in two spaces of services: *home* and *company*.

TABLE II. EXTRACT OF THE INTENTIONAL AND CONTEXTUAL SERVICE DESCRIPTIONS

Service	Intention	Required Execution Context	Service realization
Service Sv1	I1 Consult client view	- Device.Network.type ≠ Ethernet - User.location.name ≠ Company - User.Role = Commercial - User. Profile.Expertise = Low	- ***AccessClientViewVPN*** (TunnelVPN, SSLAuthentification, DataEncryption, ClientListPage, ViewClientWS)
Service Sv2	I1 Consult client view	- Device.Network.type ≠ Ethernet - User. Profile.Expertise = High - User. Role = Commercial	- ***AccessClientView*** (SSLAuthentificationWS, FindClientByAddressWS, DetailedClientView)
Service Sv4	I2 Edit propoal	- Device.Network.type = Ethernet - User.location.name = company - Device.memory > 512	- ***ProposalEditionFax*** (ProposalEditionWS, FaxService)
Service Sv12	I4 Organize conference	- User.Role = Commercial - User.location.name ≠ Company - Device.Memory > 512 - Device.Network.type ≠ Ethernet	- ***RequestVideoConference*** (RequestVirtualRoom, RequestBordWidth, StartVideoConfWS)

V. EVALUATION

In order to evaluate the feasibility of the proposed vision, we have implemented our service discovery mechanisms using Java technologies, notably Jena [4], an open source Java framework for Semantic Web, and Pellet [10], an OWL reasoner for Java. We evaluated this mechanism based on a set of 400 service descriptions from OWL-TC2 data set [31], which we enriched with intentional and contextual description. Then, we implemented our prediction mechanism using the same technologies as used in our proposed service discovery mechanism. We evaluated it on a pre-defined database simulating user's traces and the service description base used for evaluating the service discovery algorithm.

More specifically, we deployed our algorithms on a machine Intel Core i5 1.3 GHz with 4 GB memory. The purpose of these experiments is to evaluate the validity of our algorithms and their feasibility. Thus, we formulate a set of user's requests relatives to the travel domain. These request are represented by the user's intention and his current context. These requests are formalized according to three different distributions. The first distribution considers requests that are very similar to the service (*service discovery*) or the state (*service prediction*). Then, the second distribution illustrates situations where (i) the elements describing the intention and/or the context are not described in ontologies while there is services or clusters that are similar to this request and (ii) the elements describing the intention and the context are described in ontologies while there is no service or state that are similar to this request. Finally, the third distribution shows the influence of the threshold by presenting in this distribution requests that are within the limits of the threshold and others that are beyond the threshold.

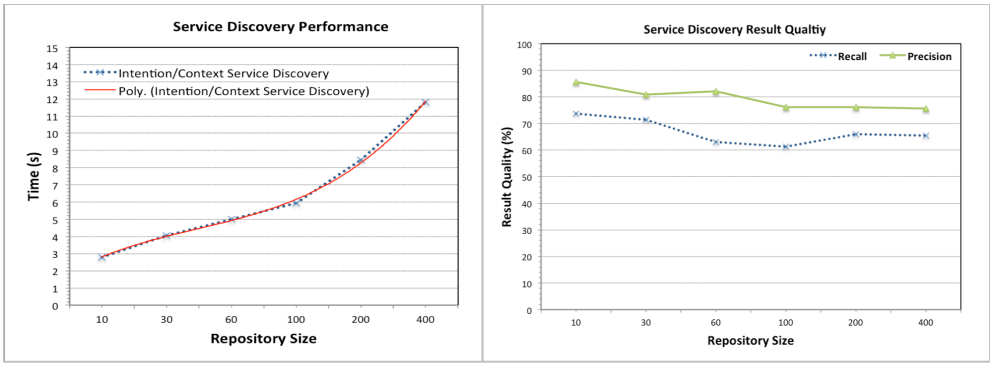


Fig. 9. a) the Service Discovery performance, b) The service discovery result quality

Our first experiments concern the *evaluation of our service discovery mechanism*. We measure the performance of this mechanism by varying the number of services between 100 and 400. We measure the average execution time taken by the algorithm to determine which services are the most appropriate to the user's request. As illustrated in Fig. 9.a, the execution time follows a polynomial trend of degree three However, even if the execution time still higher, we can observe that despite the fact that we have increased the number of services over forty times, the response time has only increased by four times. Thus, the tests that we conducted, have demonstrated the good scalability of our mechanism, even when the number of available services increases. In addition, we can improve the execution time by using the Java threads in the implementation.

Besides, in order to measure the quality of the result, we cover the two most useful quality metrics: *precision* and *recall* [43]. Through the experiments, we observe that the precision and recall are interesting factors when considering the intention and context in the service discovery. The result presented in Fig. 9.b shows that we obtained a higher precision percentage, about 80%. This indicates that our service discovery algorithm has a greater chance to retrieve the most appropriate service according to user's intention and context.

However, the good results of precision are accompanied by less interesting results concerning the recall, as illustrated in Fig. 9.b. We can observe that the average recall approximates 67%. These results can be explained by the evaluation of certain situations that can harm the results quality. For example, we have described some user's request where the elements of the intention are not described in ontologies, while it exists in the service repository a set of services able to satisfy this intention in the current user's context.

Our second experiments concern the *evaluation of our service prediction mechanism*. We measure the performance of our algorithms with respect to the number of clusters, observations in the history and states of the user's behavior model, by measuring the average processing time. For example, the execution time of the prediction algorithm is measured by varying the number of states in the user's behavior model, between 8 and 168 states. This time represents the average execution time set to predict the next service that satisfies a future user's intention according to his immediate intention and current context. As illustrated in Fig. 10.b, the execution time follows a polynomial trend of degree three, like the service discovery algorithm, from 1,63 s for 8 states to 4,16 s for 168 states. We increased the number of states over twenty five times, while the execution time has only increased by two and half times. This allows us to validate the feasibility of our prediction algorithm. However, these results can be optimized, such as the service discovery algorithm.

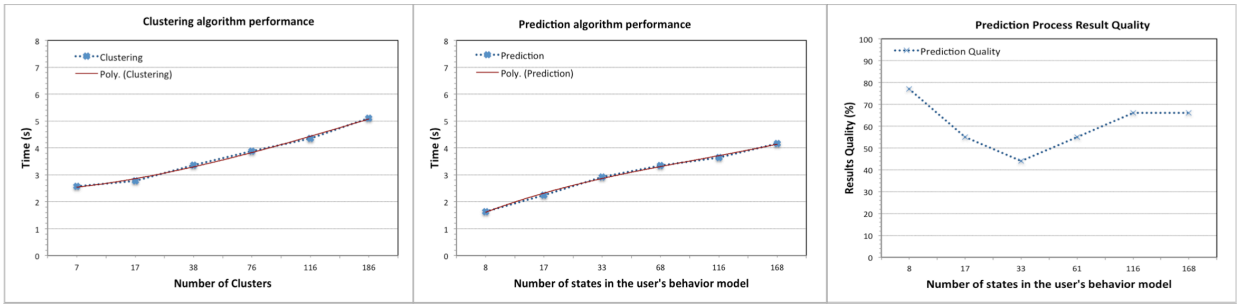


Fig. 10. a) The clustering performance, b) The prediction performance, c) The prediction process results quality

Besides, in order to measure the result quality, we use a quality metric (inspired from the precision metric used to evaluate the service discovery) used to check whether the predicted service is the one that is expected or not. We determined previously the service that must be returned by the algorithm. We compared, thereafter, this service with the returned service. We illustrate in Fig. 10.c the quality percentage achieved by the algorithm by varying the number of states in the behavior model. This percentage represents the average quality obtained for a set of requests. The results presented in Fig. 10.c indicate that the prediction algorithm has a good quality that is around 60%. These results can be explained by the evaluation, for example, of situations where the intention's verb and/or target are not described in ontologies, whereas it exists in the database clusters or states that are similar to the user's intention in his current context. In this case, the prediction algorithm does not return a result. This contributes to the degradation of the results quality thereby obtaining a quality of 0 %. In addition, in the case where situations are described by intentions where the verb and/or the target are fairly generic or specific, we obtain a quality in some cases below 45%. Thus, when the system designer sets very high threshold settings, some clusters or states that can meet the immediate user's intention in his current context will not be selected, and this contributes to the degradation of the quality.

The analysis of these results shows the importance of the discovery and prediction mechanisms in our user-centric view. We believe that the proposed mechanisms allow really the selection of the service that fulfills the user's immediate needs and the anticipation of his future need. This is thanks to both its intentional approach, which is more transparent to the user, and its contextual approach that restricts services to those that are valid. However, it is important to note that we cannot get that good result if the system designer does not establishes from the beginning a rich description of the available services and the different ontologies and the most appropriate threshold setting.

VI. CONCLUSION

In this paper, we have proposed a new user-centric vision for PIS, based on a service-oriented and context-aware intentional approach. This new vision is needed in order to hide the complexity of these systems and to achieve the transparency required by theirs users. The core of this vision is the notion of 'Space of Services', which is a conceptual framework for understanding key concepts of PIS. According to this framework, a PIS is defined as a set of permeable spaces in which services offered to the users and sensors collecting context information coexist. The dynamics of this space is captured by the context observed for the services and sensors.

Based on this conceptual point of view, three other points of views have been proposed. The functional point of view presents service discovery and prediction mechanisms. These mechanisms allow us to not only offer the user the most suitable services given his current intention and context, but also to anticipate the user's future needs in order to propose a service that can interest him in a fairly understandable and less intrusive way. The support point of view offers a complementary view, for the designer. It proposes a methodological guidance allowing IT management to apply the Space of Service and then to specify the expected functionalities of their system as well as context information that will be captured by it for a better adaptation. This allows keeping control over the definition of the system, while allowing the inclusion of a highly dynamic environment. Finally, the architecture point of view proposes a new architecture, named IPSOM, which integrates service discovery and prediction mechanisms mentioned above.

All these points of views form a technology-independent coherent package, which can be applied to any context or service technology. Indeed, the proposed points of views are generic and can be implemented by (i) any service technologies (Web services described in OWL-S or in WSML), and (ii) any context models and acquisition technologies. Thanks to this independence, our solution is open and can be applied on the top of any IS. In order to prove its feasibility, we have performed a first implementation considering services described in OWL-S [27]. We have implemented IPSOM architecture using Java technologies, and notably Jena [4] and Pellet [10]. We have evaluated the service discovery and prediction mechanisms using this implementation, proving the feasibility of the proposed mechanisms.

By this vision, we believe we are contributing to the improvement of PIS transparency and productivity. Evaluation of the user acceptance of the proposal requires applying it in a real case in an enterprise. Such evaluation should consider both the user's and designer's points of view. The first one should consider the user acceptance of the prediction mechanism as well as the level of transparency provided by the mechanism, while the second should consider the use of the proposed methodology for designing new PIS in terms of whether designers will be able to understand the notion of Space of Services and be able to easily apply the proposed methodology.

REFERENCES

[1] S. Abbar, M. Bouzeghoub, and S. Lopez, "Context-Aware Recommender Systems: A Service- Oriented Approach," In Proceedings of the 3rd International Workshop on Personalized Access, Profile Management, and Context Awareness in Databases, Lyon, France, 2009.

[2] R. Ali, F. Dalpiaz, and P. Giorgini, Paolo, "A goal-based framework for contextual requirements modeling and analysis," Requirements Engineering, 15(4), 2010, pp. 439–458.

[3] K. Aljoumaa, S. Assar, and C. Souveyet, "Reformulating User's Queries for Intentional Services Discovery Using an Ontology-Based Approach," In Proceedings of the 4th IFIP International Conference on New Technologies, Mobility and Security, 2011, pp. 1-4.

[4] Apache, "Apache Jena," Retrieved from http://jena.apache.org/, 2013

[5] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," International Journal of Ad Hoc and Ubiquitous Computing, 2(4), 2007, 263-277.

[6] G. Bell, and P. Dourish, "Yesterday's tomorrows: notes on ubiquitous computing's dominant vision," Personal and Ubiquitous Computing, 11, 2007, pp. 133-143.

[7] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," Pervasive and Mobile Computing, 6, 2010, pp. 161–180.

[8] J. Brnsted, K. Hansen, and M. Ingstrup, "Service Composition Issues in Pervasive Computing," Pervasive Computing, 9(1), 2010, pp. 62-70.

[9] T. Chaari, F. Laforest, and A. Celentano, "Adaptation in context-aware pervasive information systems: the SECAS project". Journal of Pervasive Computing and Communications, 3(4), 2007, pp. 400-425.

[10] Clark & Parsia. "Pellet: OWL 2 Reasoner for Java," Retrieved from http://clarkparsia.com/pellet, 2013

[11] A.K. Dey, "Understanding and using context". Personal and Ubiquitous Computing, 5(1), 2001, pp. 4-7.

[12] A.K. Dey, "Intelligibility in ubiquitous computing systems," In A. Ferscha (Ed.), Pervasive Adaptation: Next generation pervasive computing research agenda, 2011, pp. 68-69.

[13] H.J. Eikerling, P. Mazzoleni, P. Plaza, D. Yankelevich, and T. Wallet, "Services and mobility: the PLASTIC answer to the Beyond 3G challenge". Retrieved from http://plastic.paris-rocquencourt. inria.fr/promotion-material/white_paper_plastic_v1-3.pdf, 2007

[14] W. Feller, "An introduction to probability theory and its applications," J. Wiley & Sons.

[15] D. Fensel, F.M Facca, E. Simperl, and I. Toma, "Semantic Web Services," Springer, 2011

[16] H. Hagras, "Intelligent pervasive adaptation in shared spaces," In A. Ferscha (Ed.), Pervasive Adaptation: Next generation pervasive computing research agenda, 2011, pp. 16-17.

[17] R.S. Kaabi, and C. Souveyet, "Capturing intentional services with business process maps," In Proceedings of the 1st IEEE Int Conf on Research Challenges in Information Science, 2007, pp. 309-318.

[18] M.U.Khan, "Unanticipated Dynamic Adaptation of Mobile Applications," PhD Thesis. University of Kassel, 2010.

[19] P.E. Kourouthanassis, and G.M Giaglis, "A Design Theory for Pervasive Information Systems," In Proceedings of IWUC, 2006, pp. 62-70.

[20] A. Lapouchnian, and J. Mylopoulos, "Modeling Domain Variability in Requirements Engineering with Contexts," In Laender, A.; Castano, S.; Dayal, U.; Casati, F. and de Oliveira, J. (Eds), Conceptual Modeling - ER 2009, LNCS vol. 5829, Springer, 2009, pp.115–130.

[21] Z. Maamar, D. Benslimane, and N.C. Narendra, "What can context do for web services?," Communication of the ACM, 49(12), 2006, 98-103.

[22] R. Mayrhofer, "An Architecture for Context Prediction," PhD Thesis. Johannes Kepler University, 2004.

[23] I. Mirbel, and P. Crescenzo, "From end-user's requirements to Web services retrieval: a semantic and intention-driven approach,' Lecture Notes in Business Information Processing, 53, 2010, pp. 30-44.

[24] S.B Mokhtar, A. Kaul, N. Georgantas, and V. Issarny, "Efficient Semantic Service Discovery in Pervasive Computing Environments," ACM/IFIP/USENIX 7th International Middleware Conference, 2006.

[25] S.B. Mokhtar, D. Fournier, N. Georgantas, and V. Issarny, "Context-Aware Service Composition in Pervasive Computing Environments," LNCS vol. 3943, 2006, pp. 129-144. Springer.

[26] S. Najar, M. Kirsch-Pinheiro, and C. Souveyet, "The influence of context on intentional service," 5th Int. Workshop on Requirements Engineerings for Services, IEEE Conference on Computers, Software, and Applications, Munich, Germany, 2011, pp. 470 – 475.

[27] S. Najar, M. Kirsch-Pinheiro, and C. Souveyet, "Enriched Semantic Service Description for Service Discovery: Bringing Context to Intentional Services". International Journal on Advances in Intelligent Systems, 5(1, 2), 2012, pp. 159-174.

[28] S. Najar, M. Kirsch-Pinheiro, C. Souveyet, and L.A Steffenel, "Service Discovery Mechanism for an Intentional Pervasive Information System," 19th Int. Conference on Web Services, 2012, pp. 520 -527.

[29] S. Najar, O. Saidani, M. Kirsch-Pinheiro, C. Souveyet, and S. Nurcan, "Semantic representation of context models: a framework for analyzing and understanding". In J.M. Gomez-Perez, P. Haase, M. Tilly, & P. Warren (Eds.), 1st Workshop on Context, information and ontologies, European Semantic Web Conference, 2009, pp. 1-10, ACM.

[30] T. Olsson, B. Bjurling, M. Chong, and B. Ohlman, "Goal Refinement for Automated Service Discovery," In Proceedings of the 3rd International Conf on Advanced Service Computing, Rome, Italy, 2011, pp. 46–51.

[31] OWL-TC, "SemWebCentral: OWL-S Service Retrieval Test Collection," Retrieved from http://projects.semwebcentral.org/projects/owls-tc/, 2013.

[32] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara, "Semantic Matching of Web Services Capabilities". LNCS vol. 2342, 2002, pp. 333-347. Springer.

[33] N. Prat, "Goal Formalisation and Classification for Requirements Engineering," In Proceedings of the third International Workshop on Requirements Engineering : Foundation for Software Quality, 1997.

[34] D. Preuveneers, and Y. Berbers, eContext-driven migration and diffusion of pervasive services on the OSGi framework," Int Journal of Autonomous and Adaptive Communication Systems, 3(1), 2010, 3-22.

[35] C. Rolland, C. Souveyet, and M. Kirsch-Pinheiro, "An Intentional Approach to Service Engineering," IEEE Transactions on Service Computing, 3, 2010, pp. 292-305.

[36] L.O. Santos, E.G da Silva, L.F. Pires, and M. van Sinderen, "Towards a Goal-Based Service Framework for Dynamic Service Discovery and Composition," 3rd Int. Conference on Information Technology: New Generations, 2009, pp. 302-307. IEEE Computer Society.

[37] S. Sigg, S. Haseloff, and K. David, "An alignment approach for context prediction tasks in ubicomp environments," IEEE Pervasive Computing, 9(4), 2010, pp. 90-97.

[38] V. Suraci, S. Mignanti, and A. Aiuto, "Context-aware Semantic Service Discovery," 16th IST Mobile and Wireless Communications Summit, 2007, pp. 1-5.

[39] A. Toninelli, A. Corradi, and R. Montanari, "Semantic-based discovery to support mobile context-aware service access," Computer Communications, 31(5), 2008, pp. 935-949.

[40] Y. Vanrompay, "Efficient Prediction of Future Context for Proactive Smart Systems," PhD Thesis. Katholieke Universiteit Leuven, Faculty of Engineering, Leuven, Belgium, 2011.

[41] Y. Vanrompay, M. Kirsch-Pinheiro, and Y. Berbers, "Service Selection with Uncertain Context Information". In S. Reiff-Marganiec, & M. Tilly (Eds.), Handbook of Research on Service-Oriented Systems and Non-Functional Properties: Future Directions. IGI Global, 2011.

[42] M. Weiser, "The computer for the 21st century," Scientific American, 265(3), 1991, pp. 66-75

[43] Y. Xin, and W. Hao, "A review of Web service selection process based on Qos," In Proceedings of the International Conference on Computer Science and Automation Engineering, 2012, pp. 486-490. IEEE.

Annex XI

Paper

Steffenel, L.A. & Kirsch-Pinheiro, M., "Improving Data Locality in P2P-based Fog Computing Platforms", *9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2018)*, Leuven, Belgium, November 5-8, 2018. DOI: doi:10.1016/j.procs.2018.10.151

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 141 (2018) 72–79

Procedia Computer Science

www.elsevier.com/locate/procedia

The 9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2018)

Improving Data Locality in P2P-based Fog Computing Platforms

Luiz Angelo Steffenel[a], Manuele Kirsch Pinheiro[b]

[a] *Université de Reims Champagne-Ardenne, Reims, France*
[b] *Université Paris 1 Panthéon-Sorbonne, Paris, France*

Abstract

Fog computing extends the Cloud Computing paradigm to the edge of the network, relying on computing services intelligently distributed to best meet the applications needs such as low communication latency, data caching or confidentiality reinforcement. While P2P is especially prone to implement Fog computing platforms, it usually lacks important elements such as controlling where the data is stored and who will handle the computing tasks. In this paper we propose both a mapping approach for data-locality and a location-aware scheduling for P2P-based middlewares, improving the data management performance on fog environments. Experimental results comparing the data access performances demonstrate the interest of such techniques.

Keywords: fog computing; P2P overlay; data locality; big data; distributed hash table

1. Introduction

The multiplication of computational devices around us (smart-phones, tablets, nanocomputers) and Internet of Things (IoT) devices raises several challenges on how to coordinate these devices and how to manage the information in such heterogeneous and dynamic environments (sometimes referred as "pervasive environments"). Information produced in such environments are often handled using a client-server model, in which the aggregation and data analysis are performed over distant facilities on the *cloud* (e.g. [14, 11]). This is mostly explained by the flexibility and the computing power offered by these platforms. In spite of these advantages, relying solely on cloud infrastructures has also several drawbacks. For example, the data transfer to a remote cloud facility may induce non-negligible delays and slow down the data processing and decision-making. In addition, applications that exclusively rely on distant services can freeze or fail if the network connection is unstable or faulty.

* Corresponding author. Tel.: +33 326-91-32-18.
E-mail address: angelo.steffenel@univ-reims.fr

10.1016/j.procs.2018.10.151

Therefore, we need to re-imagine how to transmit, store and analyze data in such environments. Several alternatives for the "all to the cloud" philosophy exist, including pervasive grids [17], (mobile) edge computing [8, 10, 19] or fog computing [3]. All share the same basic principle: to exploit the computing power from devices in the edge of the network in order to accomplish tasks usually delegated to a distant facility like a cloud or a datacenter. However, new challenges raise from these alternatives, and among them the data locality and context-awareness. Indeed, in order to be efficient, fog computing platforms should ensure that the services are performed closed to the end devices and that the computing resources are assigned according to their capabilities at runtime. To make this possible, these two principles (data locality and context-awareness) become a necessity on these platforms.

Besides, P2P middlewares are especially prone to implement fog computing platforms as they are inherently distributed, fault-tolerant and easy to deploy, but usually they lack important elements such as controlling where the data is stored and who will handle the computing tasks. Indeed, most P2P storage overlays are designed with hashing functions that uniformly distribute and replicate data across the network, which it is usually accompanied by a loss of information about the data locality [30], making hard any tentative of optimization.

In this paper we propose a job-based mapping approach for data-locality and a location-aware scheduling mechanism for opportunistically placing jobs on available resources on the edge. Experimental benchmarks conducted over our mobile cloud platform CloudFIT [23] demonstrate that, through these approaches, it is possible to obtain interesting performances for fog platforms using available resources in a reasonable and opportunistic way.

The rest of the paper is organized as follows: Section 2 introduces related works on fog and edge computing. Section 3 analyses the challenge of bringing data locality to fog platforms and introduces our proposal for data locality and for context-aware scheduling. Section 4 discusses some experimental results, while Section 5 presents our conclusions.

2. Related Works

The dissemination of nearby devices with non-negligible computational capabilities promotes the production of large amounts of data as well as the integration of these devices into the data processing, an approach opposed to the *pure cloud* model in which all storage and data processing is accomplished on remote servers. All naturally, several initiatives like pervasive grids [17], mobile edge computing [8, 10], cloudlets [19], or fog computing [3] have been proposed in order to move some applications and services closer to the end user.

While these works sometimes differ, mostly they share the same definitions [25]. Indeed, fog computing has been defined as a paradigm that extends the cloud computing and its services to the periphery of the network [5], while (mobile) edge computing aims to turn nearby base stations into intelligent service centers capable of providing highly personalized services [25]. Both refer to an infrastructure in which computing tasks are distributed to locations that best meet the applications needs, such as low communication latency, higher bandwidth or confidentiality requirements. Similarly, pervasive grids [17] and cloudlets [19] aim at distributing tasks to devices surrounding the users. The term *fog* itself is used to express the idea of services surrounding users and data sources, in opposition to the remote resources from the *clouds* [3]. This is illustrated in Fig. 1, which depicts the relationship between IoT, *fog* and *cloud* [32]. For the sake of simplicity, all through this paper we use the term *fog computing* indistinctly to represent the challenges of fog computing, edge computing, cloudlets, pervasive grids as well as any other weakly coupled network characterized by the heterogeneity of resources and tasks.

As noticed by Yi et al. [31], the challenges on fog are many and include the deployment, the orchestration, the migration of tasks/services, the management of the networks, and so on. Hence, several works in the literature focus on proposing architectures to deploy services and applications on the fog, using different approaches based on virtualization [19], micro-clouds [9], micro-services [26] or workflows [15]. Besides, industrial solutions often rely on specific hardware (e.g., proprietary routers and switches from Cisco [4]) or depend on a centralized server to coordinate the resources (e.g. [29]). The lack of standardization is also a major issue, and while initiatives like the *Open Fog Consortium* [6] may help technology maturation, they may also impose a complex software dependency or a non-negligible memory footprint that may prevent the deployment in low-end devices.

The possibility of using low-end devices is a key aspect on fog computing, since those are considered for data processing, preventing transferring large amounts of data to cloud infrastructures. Frequent or heavy data transfers may negatively impact the network and the application performance (as we demonstrate on [22]). Deploying applications (i.e. jobs) close to the production sites or the end clients being one of the promises of fog computing, the notion of

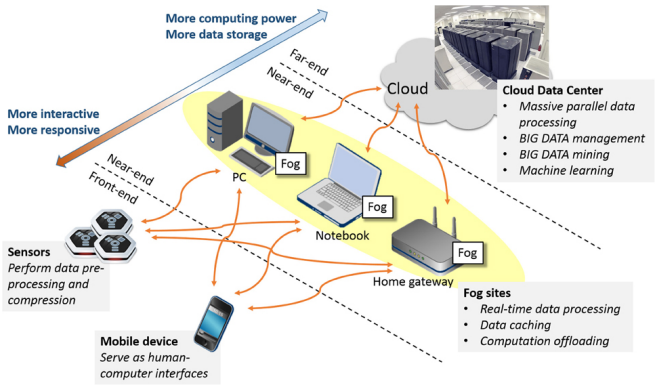


Fig. 1. Conceptual architecture from a *fog/cloud* infrastructure [32]

data locality becomes a key factor when considering job placement on fog infrastructures. In the case of a P2P-based platform, whose loose coupling is particularly interesting on fog environments, data storage can be provided through the use of Distributed Hash Tables (DHT), but this comes with a non-negligible drawback: most DHTs are designed with hashing functions to uniformly distribute and replicate data across the network. While this approach contributes to load balance and avoiding losing data in the case disconnections, it is usually accompanied by a loss of information about the data locality [30], making hard any tentative of storage or data transfer optimization.

Finally, due to the heterogeneity that characterize fog environments, data and service placement must also consider runtime conditions on each node in order to consume resources efficiently, preferring nodes that have available resources instead of nodes that are already overloaded by other tasks. Context may also consider other parameters such as the remaining battery lifetime, which lead to a string requirement for automatic and intelligent reconfiguration of the topological structure and assigned resources within the workflow [27]. In order words, context-awareness, which can be seen as the capability of system to adapt its behavior to changes in its execution context [1], becomes a central aspect for fog platforms, such as underlined for pervasive grids [21].

3. Optimizations for the Fog

When processing data on the fog, it is important to consider the data access and data management performance, since most operations involve the collection, transformation and analysis of data. Indeed, the effects of data transfers can be particularly important on fog environments. We could observe such effects in [22], in which frequent transfers of data penalized the overall performance. When looking at Big Data solutions, which are confronted to similar situations, we observe that platforms such as Apache Hadoop [28] use the concept of *data locality* to prioritize computing on nodes that hold a copy of the data to be processed. Besides, the heterogeneity of fog environments can also impact the application deployment since devices may have very different capabilities and constraints. In [23] we observed the effects of the heterogeneity on the application performance, highlighting the importance of observing the nodes status before assigning jobs for execution. Thus, we identified two key points to be addressed by fog computing middlewares: (*i*) the support for *data locality* and (*ii*) its integration with the task scheduling.

3.1. Per-Job Data Locality

The first factor to consider when handling data in the fog is where data lies (the *data locality*). While some applications rely on external storage servers (e.g., a cloud storage service), this solution is not always adapted since it incurs extra latencies or transfer fees. Ideally, a fog computing solution must be able to control where data is stored.

This is currently not the case of most P2P-based platforms. As stated before, these platforms use to uniformly distribute and replicate their data across the network. While some non-uniform hashing functions exist [13], the typical DHT hashing uses MD5 or SHA to compute resource and node IDs uniformly distributed over the network. Although these solutions prevent data to be lost when clients disconnect, it also complexifies the data transfer optimization since data locality information is actually lost [30]. Recently we observed some works on ”controlled” hashing, as the recent

development of geographical databases and NoSQL boosted the research for methods to improve the access to co-located data [2]. Most of these works concern specific data types like those found on geographical databases, whose queries can benefit from grouping data in a given area. Hence, one can express the data location as a set of coordinates, used to generate a *geohash* key [12] or to identify zones like in CAN (Content Addressable Network) [18], quad-tree hashing [24] or in a Voronoi overlay network [16].

Unfortunately, these solutions are often globally applied, i.e., all nodes in the network become responsible for a given zone in spite of being directly related to that data or not. In our view, fog applications must be able to control both elements, the nodes performing the tasks and where the data is located. To do this, we need to extend the concept of data locality to accommodate the principle of group membership. Indeed, we need to ensure that data segments from an application are preferentially distributed among the nodes that will execute that application.

The group membership (or "community") represents a subset of the system nodes, and may be statically or dynamically defined. In the first case, a previous knowledge of the network and the application requirements drives the assignment of a group of nodes to perform some specific role. Indeed, a community may include the nodes covering a specific area (in order to host a proximity service), nodes that present similar computing performances (same CPU type), etc. In the second case, dynamic factors such as the nodes status and context, or even the evolution of the network (nodes joining or leaving), are used to form temporary associations between nodes in order to execute an application. In both cases, different communities may be created to fulfill specific requirements (nodes capabilities, location or security requirements, etc.) and a node may belong to one or more communities.

As both static and dynamic communities require data locality to perform correctly, we propose to manipulate the hash key so data is not randomly spread among all nodes but specifically attached to a community. While a typical DHT uses a simple mapping where the same hashing function computes the data hash (the content key) and the node ID (the location key), we rely on a double hashing function that decouples the location key and the content key for a resource: in a first moment, the content key is obtained through a traditional hashing method. Later, the location key is computed to map only among the community nodes.

To illustrate this approach, Fig. 2 shows an example of a mapping that reinforces the data proximity. In a traditional P2P storage with a single location key, a resource r_3 could be stored in any node in the network, depending on the hash result (like for example $hash(r_3) = kl_7$). By using a location key and a content key with a community-aware hash function $hash_{ca}()$, we can restrict the location key to the nodes in the community, all while properly identifying a resource. Hence, for a resource r_3 and a community C_1 composed by nodes with IDs kl_2, kl_3 and kl_4, we can compute a community-aware location key kl' that points to a node from the community. This way, the primary copy of the resource r_3 will be located in the node kl' with the content key kc_3. As a consequence, this mapping improves the probability that the primary copy of a resource is stored in a node from the group. In addition, fault-tolerance is ensured as the storage overlay can continue to perform replication to other nodes, even those outside the community.

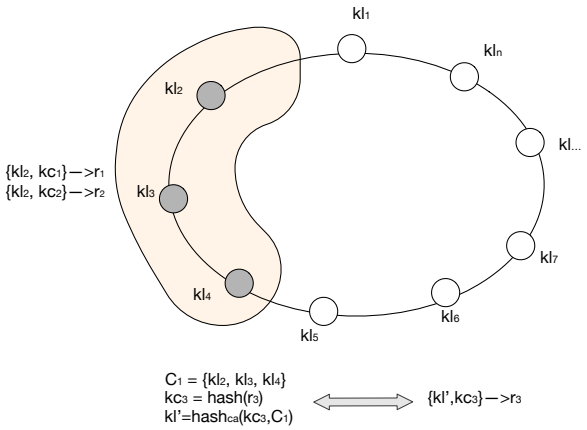


Fig. 2. Location key remapping to reinforce *data-locality*

To retrieve a resource, two strategies can be applied. If the community is stable (for example, a predefined set of nodes) one can rely on the same $hash_{ca}()$ function to find the resource. If the community membership can vary with

the time (nodes joining or leaving the community), an index file can be used to keep a track of the resources' location, in spite of the eventual mapping skew when the community changes.

This mapping can be implemented on any DHT overlay through the use of specific hashing functions. In order to validate this approach, we implemented it using the TomP2P[1] overlay over our fog computing platform CloudFIT. TomP2P has the advantage of using several hash keys to identify a resource, instead of a single key. While a typical P2P storage uses a simple mapping where the data is indexed by the node with the closes ID to the hash key of the data, TomP2P identifies resources with up to four keys $\{k_l, k_d, k_c, k_v\}$, namely:

- k_l - location key, which determines the node ID closest to the hash key;
- k_d - domain key, used for namespacing;
- k_c - content key, which identifies different resources stored in the same location; and
- k_v - version key, which allows the managing of different versions of the same resource.

Communities are managed by a context manager from CloudFIT, which controls a set of predefined communities based on the node's specifications and status. Nevertheless, new communities may be created to respond to specific application requirements. Together with a context-aware scheduler, presented in the next section, we implemented our data-locality mapping as following:

- nodes joining a community sign in their ID in a *membership index table*. This file is visible only to the community members (using the domain key) and conflicts are prevented by using a version key too;
- when writing a file, a content key is computed and a data locality hash function is applied over this key to assign its location; this content-location tuple is written in a *resource index table*;
- when reading a file, a node first computes the content key and then the location key using the data locality function. If a membership change modifies the hash mapping, then it can check the resource index table.

Using this "community-aware" approach, it is then possible to manage the data placement in the network and to reinforce data locality inside the community.

3.2. Scheduling with context-awareness

As the previous technique helps to implement data locality, we can now address another issue related to the heterogeneity of the fog. Indeed, as the computing performance varies from device to device, context information such as processing power, available memory and storage space or current CPU load can be useful to improve the execution performance, as demonstrated by [8]. Therefore, efficiently matching the tasks with the resources capabilities and their locations is a key element on the optimization of performance-sensitive fog applications [20].

In the case of data-locality aware schedulers, the presence of a dataset on the local storage of a node can be used as a factor to determine the tasks assignments. Indeed, most DHT offer the possibility to *lookup* for a given resource on the local cache, so we can set dependencies to data resources, prioritizing tasks that have the required datasets locally available. Still, we should go further in the support to context-awareness by distributing data among the nodes according to their relative capacities. We can use a strategy similar to the geohash *precision* property [12], so that the zone covered by a node depends on a "zoom" level. By assigning larger or smaller DHT zones relative to the node's capacities, one can drive the scheduler to assign tasks accordingly.

Thus, we propose and experimented a scheduling mechanism on the CloudFIT platform, using a two-layer scheduler. The first layer is the "job" scheduler, which is enriched with a context manager that helps verifying requirements expressed by each job. Context information, which can be defined as any relevant information that can be used to characterize the situation of an entity [7], is used to establish if a node is currently able to handle a job or a task. This context information is composed by nodes current status (idle or working, CPU usage, available memory), the nodes location and the node specifications (CPU type and speed, total memory, disk space). Node specifications are used to create predefined communities corresponding to typical application requirements, so a job can also express

[1] https://tomp2p.net/

its needs in terms of ”required communities”. Once validated, the application is deployed over one or more nodes that currently match the requirements, otherwise it remains in a waiting queue until resources become free. While the primary objective of the job scheduler is to guarantee resources that fit the job requirements, it also allows load balancing, QoS and concurrency management. By distributing jobs according to the nodes current execution context, i.e., if two communities match the minimal requirements, the scheduler will deploy the job on the community that is less overloaded even if it is not the ”most powerful” community.

On the second layer we found the ”tasks” scheduler, a per-job scheduler that coordinates the tasks execution on each node. The basic scheduler algorithm in CloudFIT considers tasks as fully independent, so each node randomly starts a task and then diffuses the tasks status to the other nodes (*in execution*, *completed*), updating the list of tasks still available to be executed. However, as the tasks scheduler algorithm is inherent to each job, it can be extended in several ways and applied according to the job's requirements. Therefore, in order to implement a data-locality aware scheduling, the presence of a data resource in the local storage is used as a factor to determine the tasks' execution order. By assigning dependencies between tasks and data resources, we instrument the scheduler to prioritize tasks that have the required datasets already at hand. Additional context elements such as the current available memory, CPU load and disk space also permit the task scheduler to decide at runtime if a task can be started, avoiding penalizing an application because of an overloaded node.

4. Experimental Results

4.1. Description

In order to conduct our experiments, we deployed CloudFIT over a network composed by 5 nodes in a local cluster, and 5 nodes in the Google Cloud Platform (GCP). The local machines are identical (AMD Opteron 6164 HE, 12 cores, 48GB RAM) and interconnected by a Fast Ethernet network. GCP machines are of type *n1-standard-2* (2vCPU, 7.5 GB RAM) and located at the *us-central1-c* zone.

The experiments performed write and read operations on the DHT, with file sizes of 1kB, 10kB, 100kB, 1MB, 10MB and 100MB. Three scenarios were considered to verify the impact of data locality: operations in the same machine, in the same network and over a distant node lying in the cloud. The first case represents the situations where data locality is able to deliver data to the machine that will handle it. The same network scenario represents the intermediate scenario where the data is close to the machine, like in the case of a low-entry device that avoids storing data locally. Finally, the last scenario represents the situations where the data lies far from the node, like when data-locality is not enabled. At each run, the CloudFIT platform was shut down and the files removed. Also, the order of the files was shuffled to prevent *pipeline* effects on the network, and at least 10 runs were executed for each scenario.

4.2. Analysis

Table 1 represents the average and the percent deviation for each combination of operation (write/read), data size and scenario (same node, same network, on the cloud). These averages are also depicted in Fig. 3. We can observe that both local and network scenarios present similar performances for the reading of messages smaller than 1MB. For larger messages, however, the reading time between two nodes becomes much more important, scaling to levels almost comparable to those of accessing remote nodes. This slowdown evidences not only the network limitations (a 100Mbps network) but the overhead of requesting a resource through the DHT.

If we consider the writing operation, an interesting effect occurs: for most of the small messages, it is more expensive to write files in the local node than in another node in the same network. This could also be explained by the DHT management overhead, as the request for a local file needs to activate two different routines in the same machine (resource lookup and transmission), which can impact on the overall execution performance. Of course, the DHT management overhead is present in the cloud scenario, but less prominent as the communication is mainly dominated by the latency and the network bandwidth.

These observations encourage the use of data locality to improve the performance of fog applications. Indeed, if we have access to the job membership, it is cheaper to distribute the data directly to the computing nodes, instead of disseminating the data to unrelated nodes or storing it in one node to serve the others later.

Table 1. DHT Read/Write performance according to the data locality (average in milliseconds and percent deviation)

		1 kB	10 kB	100kB	1 MB	10 MB	100 MB
Same Node	**Read**	32.86 (55,1%)	30.47 (36.5%)	34.18 (5.9%)	53.29 (18.9%)	207.64 (4.9%)	2122.52 (2.0%)
	Write	618.64 (7.4%)	602,15 (27.2%)	534,55 (15.2%)	692,06 (16.7%)	742.06 (15.1%)	3140,70 (2.9%)
Same Network	**Read**	25.37 (8%)	28.44 (5.4%)	45.88 (5.7%)	160.2 (1.9%)	1057.4(0.6%)	16289.77 (0.6%)
	Write	217.22 (14.2%)	194.88 (24.7%)	201.85 (26%)	284.22 (12%)	1272 (3.6%)	17475.66 (1.6%)
On the Cloud	**Read**	431.25 (0.4%)	574.75 (0.6%)	860 (0.4%)	1588 (1%)	4323 (6.2%)	30494.5 (0.04%)
	Write	454.66 (1.2%)	735.25 (2%)	1169.16 (1%)	1622.33 (0.8%)	3046.42 (1.1%)	29605 (5.6%)

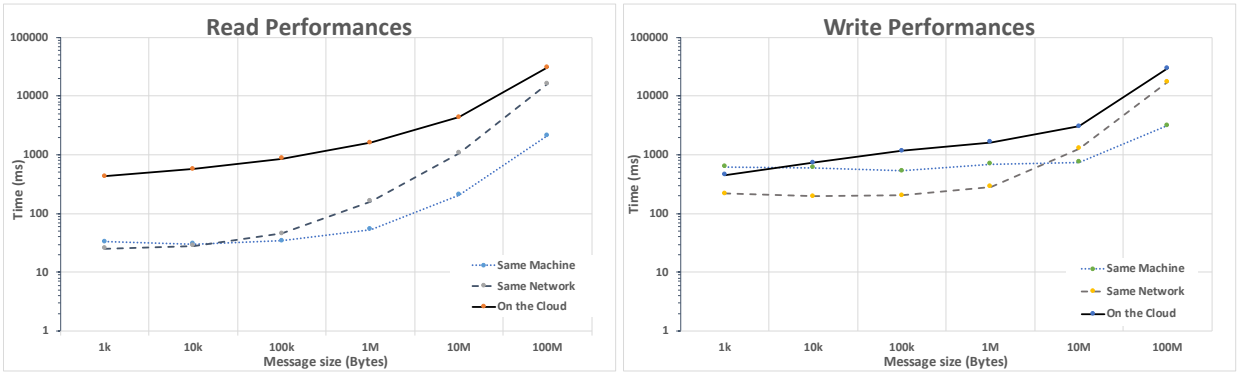


Fig. 3. Performance comparison when reading/writing data from the DHT

5. Conclusion

This work focused in innovative solutions for the implementation of data locality in P2P-based fog computing platforms. On the one hand, such platforms are a real alternative to existing platforms like [4] and [29], as it allows a better decentralization of the infrastructure and more fault-tolerant management of the fog. On the other hand, the use of DHTs as storage support usually suffers from the lack of data locality information, which penalizes most big data and data analysis applications that require low latency to perform correctly.

In this work, we discuss different strategies to bring data locality to fog computing platforms. Through the observation of the behavior from applications and devices, we designed new strategies to cope with the main performance bottlenecks in the fog: the difficulty to place the data where it needs to be, the awareness about data locality on the execution scheduling and, finally, the reduction of the data management overhead in low end nodes that characterize most fog computing proximity devices. We believe that these techniques help to improve the performance of data-intensive applications all while respecting the main aspects from fog computing such as decentralization, heterogeneity and context-awareness. Future works include the experimentation with data-intensive applications and the comparison with other fog platforms.

References

[1] Baldauf, M., Dustdar, S., Rosenberg, F., 2007. A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing 2, 263–277. doi:10.1504/IJAHUC.2007.014070.

[2] Ben Brahim, M., Drira, W., Filali, F., Hamdi, N., 2016. Spatial data extension for cassandra nosql database. Journal of Big Data 3, 11. URL: https://doi.org/10.1186/s40537-016-0045-4, doi:10.1186/s40537-016-0045-4.

[3] Bonomi, F., Milito, R., Zhu, J., Addepalli, S., 2012. Fog computing and its role in the internet of things, in: Proceedings of the 1st MCC Workshop on Mobile Cloud Computing, ACM, New York, NY, USA. pp. 13–16. doi:10.1145/2342509.2342513.

[4] Cisco, . Cisco iox. URL: https://developer.cisco.com/site/iox/docs/.

[5] Cisco Research Center Requests for Proposals (RFPs), . Fog computing, ecosystem, architecture and applications. http://www.cisco.com/web/about/ac50/ac207/crc_new/university/RFP/rfp13078.html.

[6] Consortium, O., 2017. Openfog reference architecture. https://www.openfogconsortium.org/ra/.

[7] Dey, A., 2001. Understanding and using context. Personal and Ubiquitous Computing 5, 4–7.

[8] Dey, S., Mukherjee, A., Paul, H.S., Pal, A., 2013. Challenges of using edge devices in iot computation grids, in: Int. Conf. on Parallel and Distributed Systems, pp. 564–569. doi:http://doi.ieeecomputersociety.org/10.1109/ICPADS.2013.101.

[9] Elkhatib, Y., Porter, B., Ribeiro, H.B., Zhani, M.F., Qadir, J., Rivière, E., 2017. On using micro-clouds to deliver the fog. IEEE Internet Computing 21, 8–15. doi:doi.ieeecomputersociety.org/10.1109/MIC.2017.35.
[10] ETSI, . Mobile-edge computing - introductory technical white paper. https://portal.etsi.org/.
[11] Fazio, M., Celesti, A., Puliafito, A., Villari, M., 2015. Big data storage in the cloud for smart environment monitoring. Procedia Computer Science 52, 500 – 506. doi:10.1016/j.procs.2015.05.023. the 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015).
[12] Fox, A., Eichelberger, C., Hughes, J., Lyon, S., 2013. Spatio-temporal indexing in nonrelational distributed databases, in: IEEE International Conference in Big Data, pp. 291–299.
[13] Freepastry, 2012. http://freepastry.rice.edu.
[14] Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M., 2013. Internet of things (iot): A vision, architectural elements, and future directions. Future Generation Computer Systems 29, 1645 – 1660. doi:http://dx.doi.org/10.1016/j.future.2013.01.010.
[15] Hao, Z., Novak, E., Yi, S., Li, Q., 2017. Challenges and software architecture for fog computing. IEEE Internet Computing 21, 44–53. doi:doi.ieeecomputersociety.org/10.1109/MIC.2017.26.
[16] Hu, S.Y., Chen, J.F., Chen, T.H., 2006. Von: a scalable peer-to-peer network for virtual environments. IEEE Network 20, 22–31. doi:10.1109/MNET.2006.1668400.
[17] Parashar, M., Pierson, J.M., 2010. Pervasive grids: Challenges and opportunities, in: Li, K., Hsu, C., Yang, L., Dongarra, J., Zima, H. (Eds.), Handbook of Research on Scalable Computing Technologies. IGI Global, pp. 14–30. doi:10.4018/978-1-60566-661-7.ch002.
[18] Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S., 2001. A scalable content-addressable network. ACM SIGCOMM Computer Communication Review 31, 161–172.
[19] Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N., 2009. The case for vm-based cloudlets in mobile computing. IEEE Pervasive Computing 8, 14–23.
[20] Shekhar, S., Gokhale, A., 2017. Dynamic resource management across cloud-edge resources for performance-sensitive applications, in: 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), pp. 707–710. doi:10.1109/CCGRID.2017.120.
[21] Steffenel, L., Flauzac, O., Charao, A.S., Barcelos, P.P., Stein, B., Nesmachnow, S., Pinheiro, M.K., Diaz, D., 2013. Per-mare: Adaptive deployment of mapreduce over pervasive grids, in: Proceedings of the 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC'13), Compiegne, France.
[22] Steffenel, L.A., Kirsch Pinheiro, M., 2015. Leveraging data intensive applications on a pervasive computing platform: The case of mapreduce, in: The 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015), the 5th International Conference on Sustainable Energy Information Technology (SEIT-2015), Elsevier. pp. 1034 – 1039. URL: http://www.sciencedirect.com/science/article/pii/S1877050915009023, doi:http://dx.doi.org/10.1016/j.procs.2015.05.102.
[23] Steffenel, L.A., Kirsch-Pinheiro, M., 2015. When the cloud goes pervasive: approaches for IoT PaaS on a mobiquitous world, in: Springer (Ed.), EAI International Conference on Cloud, Networking for IoT systems (CN4IoT 2015), Rome, Italy. pp. 347–356.
[24] Tanin, E., Harwood, A., Samet, H., 2007. Using a distributed quadtree index in peer-to-peer networks. The VLDB Journal 16, 165–178. URL: https://doi.org/10.1007/s00778-005-0001-y, doi:10.1007/s00778-005-0001-y.
[25] Vermesan, O., Friess, P., Guillemin, P., Giaffreda, R., Grindvoll, H., Eisenhauer, M., Serrano, M., Moessner, K., Spirito, M., Blystad, L.C., Tragos, E.Z., 2014. Internet of things beyond the hype: Research, innovation and deployment, in: Internet of Things - From Research and Innovation to Market Deployment. River Publishers, pp. 15–118.
[26] Villari, M., Fazio, M., Dustdar, S., Rana, O., Ranjan, R., 2016. Osmotic computing: A new paradigm for edge/cloud integration. IEEE Cloud Computing 3, 76–83. doi:10.1109/MCC.2016.124.
[27] Wen, Z., Yang, R., Garraghan, P., Lin, T., Xu, J., Rovatsos, M., 2017. Fog orchestration for internet of things services. IEEE Internet Computing 21, 16–24. doi:10.1109/MIC.2017.36.
[28] White, T., 2010. Hadoop: The definitive guide. 2nd edition ed., Yahoo Press, O'Reilly.
[29] Willis, D.F., Dasgupt, A., Banerjee, S., 2014. Paradrop: a multi-tenant platform for dynamically installed third party services on home gateways, in: ACM SIGCOMM workshop on Distributed cloud computing, pp. 43–44.
[30] Wu, D., Tian, Y., Ng, K.W., 2005. Aurelia: Building locality-preserving overlay network over heterogeneous p2p environments, in: Proc. of the International Conference on Parallel and Distributed Processing and Applications, Springer. pp. 1–8. doi:10.1007/11576259_1.
[31] Yi, S., Hao, Z., Qin, Z., Li, Q., 2015. Fog computing: Platform and applications, in: IEEE (Ed.), Third IEEE Workshop on Hot Topics in Web Systems and Technologies, pp. 73–78.
[32] Zao, J.K., Gan, T.T., You, C.K., Chung, C.E., Wang, Y.T., Rodríguez Méndez, S.J., Mullen, T., Yu, C., Kothe, C., Hsiao, C.T., Chu, S.L., Shieh, C.K., Jung, T.P., 2014. Pervasive brain monitoring and data sharing based on multi-tier distributed computing and linked data technology. Frontiers in Human Neuroscience 8, 370. URL: https://www.frontiersin.org/article/10.3389/fnhum.2014.00370, doi:10.3389/fnhum.2014.00370.

Annex XII

Publication list

Publication list

Synthesis

Table below synthetises all my publications, organized according the period of my career: before 2008, when I integrated the University Paris 1 Panthéon Sorbonne; between 2008 and 2014, representing the first part of my career in this university; and from 2015 until now.

	Total number of publications	Publications until 2008	Publications 2008 < x ≤ 2014	Publications after 2014	Publications classed ERA[3] A	Publications classed ERA B
Book chapters	9	0	7	2	N/A	N/A
Journal papers	12	3	3	6	1	2
International conference papers	48	14	20	14	4	8
National conference papers	13	3	5	5	N/A	N/A
Reports & dissertations	5	4	1	0	N/A	N/A
Total	87	24	36	27	5	10

Book chapters

2018 — Steffenel, L.A., Kirsch-Pinheiro, M., Vaz Peres, L., Kirsch Pinheiro, D. "Strategies to implement Edge Computing in a P2P Pervasive Grid", In Mehdi Khosrow-Pour et al. (Eds.), **Fog Computing: Breakthroughs in Research and Practice**, IGI Global, pp 142-157, 2018, doi:10.4018/978-1-5225-5649-7.ch006.

2016 — Deneckere, R., Hug, C., Jaffal, A. Kirsch Pinheiro, M., Le Grand, B., Mazo, M., Rychkova, I. "Context Management and Intention Mining for Adaptive Systems in Mobile Environments: from Business Process Management to Video Games?". In **Digital Interfaces in Situations of Mobility: Cognitive, Artistic, and Game Devices**. Common Ground Publishing, 2016.

2014 — Rychkova I., Kirsch-Pinheiro M., Le Grand B., "Automated Guidance for Case Management: Science or Fiction?", In : Ficher, L. (Ed.), **Empowering Knowledge Workers: New Ways to Leverage Case Management**, Series BPM and Workflow Handbook Series, Future Strategies Inc., 2014, pp. 67-78. ISBN : 978-0-984976478

[3] The ERA ranking corresponds to an international recognized publication ranking proposed by the Australian CORE (*Computing Research and Education Association*). It is available at https://www.core.edu.au/conference-portal, but also at http://www.conferenceranks.com, which is a Web site regrouping several other rankings, such as the Brazilian Qualis ou the MSAR, proposed byMicrosoft.

2013

Vanrompay, Y., Kirsch Pinheiro, M., Ben Mustapha, N., Aufaure, M.-A., “Context-Based Grouping and Recommendation in MANET”, In : Kolomvatsos, K., Anagnostopoulos, C., Hadjiefthymiades, S. (Eds.), **Intelligent Technologies and Techniques for Pervasive Computing**, IGI Global, 2013, pp. 157-178. ISBN : 978-1-4666-4040-5.DOI : 10.4018/978-1-4666 4038-2.ch008

Najar, S., Kirsch Pinheiro, M., Vanrompay, Y., Steffenel, L.A., Souveyet, C., “Intention Prediction Mechanism in an Intentional Pervasive Information System”, In : Kolomvatsos, K., Anagnostopoulos, C., Hadjiefthymiades, S. (Eds.), **Intelligent Technologies and Techniques for Pervasive Computing**, IGI Global, 2013, pp. 251-275. ISBN : 978-1-4666-4040-5. DOI: 10.4018/978-1-4666-4038-2.ch014.

2012

Vanrompay, Y., Kirsch-Pinheiro, M., Berbers, Y., “Service Selection with Uncertain Context Information”, In: Stephan Reiff-Marganiec and Marcel Tilly (Eds.), **Handbook of Research on Service-Oriented Systems and Non-Functional Properties: Future Directions**, IGI Global, pp. 192-215, ISBN 978-1613504321, DOI: 10.4018/978-1-61350-432-1

Ait-Ali-Slimane, A., Kirsch-Pinheiro, M., Souveyet, C., “Considering Quality of a Service in an Intentional Approach”, In: Stephan Reiff-Marganiec and Marcel Tilly (Eds.), **Handbook of Research on Service-Oriented Systems and Non-Functional Properties: Future Directions**, IGI Global, pp. 334-351, ISBN 978-1613504321, DOI: 10.4018/978-1-61350-432-1

2010

Kirsch Pinheiro, M., Carrillo-Ramos, A., Villanova-Oliver, M., Gensel, J., Berbers, Y. “Context-Aware Adaptation in Web-based Groupware Systems”, In: JingTao Yao (Ed.), **Web-based Support Systems**, Series: Advanced Information and Knowledge Processing, Springer, mars 2010, ISBN: 978-1-84882-627-4, pp. 3-31. DOI: 10.1007/978-1-84882-628-1_1

2009

Carrillo-Ramos, A., Kirsch Pinheiro, M., Villanova-Oliver, M., Gensel, J., Berbers, Y., “Collaborating agents for adaptation to mobile users”, In: Max Chevalier, Chantal Soule-Dupuy, Christine Julien (Eds.), **Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling**, IGI Global, 2009, ISBN 978-1605663067.

Preuveneers, D., Victor, K., Vanrompay, Y., Rigole, P., Kirsch Pinheiro, M., Berbers, Y., “Context-Aware Adaptation in an Ecology of Applications”, In: Dragan Stojanovic (Ed.), **Context-Aware Mobile and Ubiquitous Computing for Enhanced Usability: Adaptive Technologies and Applications**, IGI Global, 2009, ISBN 978-1605662909.

Journal papers

2019

Kirsch-Pinheiro, M. & Souveyet, C. “Is Group-Awareness Context-Awareness?: Converging Context-Awareness and Group Awareness Support”, **International Journal of e-Collaboration (IJeC)**, *15*(3) , 2019, DOI: 10.4018/IJeC.2019070101.
Classement ERA : B

2018

Kirsch-Pinheiro, M. & Souveyet, C. "Supporting context on software applications: a survey on context engineering" (« Le support applicatif à la notion de contexte : revue

de la littérature en ingénierie de contexte »), **Modélisation et utilisation du contexte**, 2(1), 2018

Steffenel, L.A., Kirsch-Pinheiro, M., Vaz Peres, L., Kirsch Pinheiro, D. "Strategies to implement Edge Computing in a P2P Pervasive Grid", **International Journal of Information Technologies and Systems Approach (IJITSA)**, 11(1), IGI Global, pp 1-15, 2018, doi:10.4018/IJITSA/2018010101.

2016 Cassales, G. W., Charao, A., Kirsch-Pinheiro, M., Souveyet, C., Steffenel, L.A. "Improving the Performance of Apache Hadoop on Pervasive Environments through Context-Aware Scheduling ", **Journal of Ambient Intelligence and Humanized Computing**, 7(3), Springer, pp. 333-345, 2016. doi:10.1007/s12652-016-0361-8.

2015 Najar, S., Kirsch-Pinheiro, M., Souveyet, C. "Service discovery and prediction on Pervasive Information System", **Journal of Ambient Intelligence and Humanized Computing**, vol. 6, issue 4, June 2015, Springer Berlin Heidelberg, pp. 407-423. ISSN: 1868-5137. DOI: http://dx.doi.org/10.1007/s12652-015-0288-5

Engel, T., Charão, A., Kirsch-Pinheiro, M., Steffenel, L.-A., "Performance improvement of data mining in Weka through multi-core and GPU acceleration: opportunities and pitfalls", **Journal of Ambient Intelligence and Humanized Computing**, vol. 6, issue 4, June 2015, Springer Berlin Heidelberg, pp. 377-390. ISSN: 1868-5137. DOI: http://dx.doi.org/10.1007/s12652-015-0292-9

2014 Steffenel, L. A., Flauzac, O., Charao, A. S., Barcelos, P. P., Stein, B., Cassales, G., Nesmachnow, S., Rey, J., Cogorno, M., Kirsch-Pinheiro, M., Souveyet, C. "MapReduce Challenges on Pervasive Grids", **Journal of Computer Science**, vol. 10, issue 11, 2014, Science Publications, pp. 2192-2207. DOI : 10.3844/jcssp.2014.2192.2207. URL: http://thescipub.com/abstract/10.3844/jcssp.2014.2194.2210

2012 Najar S., Kirsch-Pinheiro, M., Souveyet, C. "Bringing Context to Intentional Services: Enriched Semantic Service Description for Service Discovery", **International journal On Advances In Intelligent Systems**, vol. 5, issue 1 & 2, June 2012, pp. 159-174, IARIA Journals / ThinkMind, ISSN: 1942-2679

2010 Rolland, C., Kirsch-Pinheiro, M., Souveyet, C. "An Intentional Approach to Service Engineering", **IEEE Transactions on Services Computing**, vol.3, no.4, pp.292-305, Oct.-Dec. 2010. DOI: http://dx.doi.org/10.1109/TSC.2010.26
Classement ERA : A

2003 Kirsch-Pinheiro, M.; Lima, J.V.; Borges, M.R.S. "A Framework for Awareness Support in Groupware Systems". **Computer in Industry**, vol. 52 n° 1, Elsevier, 2003, pp. 47-57.
Classement ERA : B

2001 Kirsch-Pinheiro, M.; Telecken, T.; Zeve, C.D.C; Lima, J.V.L.; Edelweiss, N. "A Cooperative Environment for E-Learning Authoring". **Documents Numériques : Espaces d'Information et de Coopération**, vol. 5, n° 3-4, Hermes Science, France, 2002, pp. 89-114.
Lima, J.V. de; Kirsch-Pinheiro, M.; Telecken, T.; Edelweiss, N.; Zeve, C.D.C; Borges, T. B.; Maciel, C. B. "O uso de open source e standards para o ensino a distância dentro do projeto CEMT". **Cadernos de Informática**, vol. 2, n° 1, Instituto de Informática – URGS, Porto Alegre, Brésil, pp. 103-105, 2001.

Internation conference & workshop papers

2020

Ben Rabah, N.; Kirsch Pinheiro, M.; Le Grand, B.; Jaffal, A. & Souveyet, C., “Machine Learning for a Context Mining Facility”, **16th Workshop on Context and Activity Modeling and Recognition, (COMOREA 2020), IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops***)*, 2020, pp.678-684

2018

Kirsch Pinheiro, M., Souveyet, C., “Is Group-Awareness Context-Awareness?”, In: Rodrigues, A.; Fonseca, B. & Preguiça, N. (Eds.), **24th International Conference on Collaboration and Technology (CRIWG 2018)**, Lecture Notes on Computer Science, vol. 11001, Springer, 2018, 198-206

Steffenel, L.A., Kirsch-Pinheiro, M., "Improving Data Locality in P2P-based Fog Computing Platforms", **9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2018)**, Leuven, Belgium, November 5-8, 2018.

2017

Kirsch-Pinheiro, M., Souveyet, C. “Quality on Context Engineering Modeling and Using Context”, **10th International and Interdisciplinary Conference, CONTEXT 2017**, Paris, France, June 20-23, Lecture Notes on Computer Science, vol. 10257, 2017, pp. 432-439
Classement ERA : C

Beserra, D., Kirsch-Pinheiro, M., Steffenel, L.A., Moreno, E.D., "Comparing the Performance of OS-level Virtualization Tools in SoC-based Systems: The Case of I/O-bound Applications", **22nd IEEE Symposium on Computers and Communications (ISCC 2017),** Heraklion, Greece, July 3-6, 2017. doi: 10.1109/ISCC.2017.8024598
Classement ERA : B

Charao, A., Hoffmann, G., Steffenel, L.A, Kirsch-Pinheiro, M., Stein, B., "Performance Evaluation of Cloud-based RDBMS through a Cloud Scripting Language", **19th International Conference on Enterprise Information Systems (ICEIS),** Porto, Portugal, April 26-29, 2017. doi: 10.5220/0006350803320337
Classement ERA : C

Beserra, D., Kirsch-Pinheiro, M., Steffenel, L.A., Souveyet, C., Moreno, E.D., "Performance Evaluation of OS-level Virtualization Solutions for HPC Purposes on SoC-based Systems", **IEEE International Conference on Advanced Information Networking and Applications (AINA)**, Taipei, Taiwan, March 27-29, 2017. DOI: 10.1109/AINA.2017.73
Classement ERA : B

2016

Kirsch-Pinheiro, M., Mazo, R., Souveyet, C., Sprovieri, D., “Requirements Analysis for Context-oriented Systems”, **7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016)**, Procedia Computer Science, vol. 83, 2016, Elsevier, pp. 253–261. DOI : http://dx.doi.org/10.1016/j.procs.2016.04.123

Steffenel, L.A., Kirsch-Pinheiro, M., Kirsch Pinheiro, D., Perez, L.V. "Using a Pervasive Computing Environment to Identify Secondary Effects of the Antarctic Ozone Hole", **BigD2M Workshop, ANT/SEIT 2016**, Madrid, Spain. Procedia Computer Science, vol. 83, 2016, Elsevier, pp. 1007-1012, 2016.

2015

Steffenel, L.A., Kirsch-Pinheiro, M., "When the Cloud goes Pervasive: approaches for IoT PaaS on a mobiquitous world", **EAI International Conference on Cloud, Networking for IoT systems (CN4IoT 2015)**, Rome, Italy, October 126-27, 2015.

Cassales, G.W., Charao, A.,Kirsch-Pinheiro, M., Souveyet, C., Steffenel, L.A., "Context-Aware Scheduling for Apache Hadoop over Pervasive Environments", **6th International Conference on Ambient Systems, Networks and Technologies (ANT 2015),** Procedia Computer Science, vol. 52, Jun 2015, Elsevier, pp. 202–209. DOI: http://dx.doi.org/10.1016/j.procs.2015.05.058.

Steffenel, L.A., Kirsch-Pinheiro, M., "Leveraging Data Intensive Applications on a Pervasive Computing Platform: The Case of MapReduce", **BigD2M Workshop, ANT/SEIT 2015**, London, England. Procedia Computer Science, vol. 52, pp. 1034-1039, 2015.

Jaffal, A., Le Grand, B., Kirsch-Pinheiro, M., "Refinement Strategies for Correlating Context and User Behavior in Pervasive Information Systems", **BigD2M Workshop, ANT/SEIT 2015**, London, England. Procedia Computer Science, vol. 52, pp. 1040-1046, 2015.

Steffenel, L.A., Kirsch-Pinheiro, M., " CloudFIT, a PaaS Platform for IoT Applications over Pervasive Networks ", **ESOCC Workshops 2015**: 20-32, Taormina, Italy. pp. 20-32, 2015.

2014

Engel, T.A., Charão, A.S., Kirsch-Pinheiro, M., Steffenel, L.A., "Performance Improvement of Data Mining in Weka through GPU Acceleration", **5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014),** Procedia Computer Science, vol. 32, 2014, Elsevier, pp. 93–100. DOI: http://dx.doi.org/10.1016/j.procs.2014.05.402

Najar, S., Kirsch Pinheiro, M., Souveyet, C., "A new approach for service discovery and prediction on Pervasive Information System", **The 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014),** 2-5 June 2014, Hasselt, Belgium, Procedia Computer Science, vol. 32, 2014, Elsevier, pp. 421–428. DOI: http://dx.doi.org/10.1016/j.procs.2014.05.443

Najar, S., Kirsch Pinheiro, M., Le Grand, B., Souveyet, C. "A user-centric vision of service-oriented Pervasive Information Systems", **8th International Conference on Research Challenges in Information Science**, May 28-30 2014, Marrakesh, Morocco, pp. 359-370
Classement ERA : B

Najar, S., Kirsch Pinheiro, M., Souveyet, C. "A context-aware intentional service prediction mechanism in PIS", In: David De Roure, Bhavani Thuraisingham and Jia Zhang (Eds.), **IEEE 21st International Conference on Web Services (ICWS 2014)**, 27 June - 2 July 2014, Anchorage, Alaska, USA, IEEE CS, pp. 662-669. DOI : 10.1109/ICWS.2014.97
Classement ERA : A

Jaffal, A., Kirsch-Pinheiro, M., Le Grand, B., "Unified and Conceptual Context Analysis in Ubiquitous Environments", In: Jaime Lloret Mauri, Christoph Steup, Sönke Knoch (Eds.), **8th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2014)**, August 24 - 28, 2014 – Rome, Italy, ISBN: 978-1-61208-353-7, IARIA, pp. 48-55.
Classement ERA : C

Cassales, G.W., Charão, A.S., Kirsch-Pinheiro, M., Souveyet, C., Steffenel, L.A., "Bringing Context to Apache Hadoop", In: Jaime Lloret Mauri, Christoph Steup, Sönke Knoch (Eds.), **8th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2014)**, August 24 - 28, 2014 – Rome, Italy, ISBN: 978-1-61208-353-7, IARIA, pp. 252-258.
Classement ERA : C

2013

Garcia, K., Kirsch-Pinheiro, M., Mendoza, S., Decouchant, D., "An Ontological Model for Resource Sharing in Pervasive Environments", **2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT),** volume 1, Nov. 17-20, 2013 Atlanta, GA USA, pp. 179-184. DOI : http://dx.doi.org/10.1109/WI-IAT.2013.27
Classement ERA : C

Garcia, K., Kirsch-Pinheiro, M., Mendoza, S., Decouchant, D., "Ontology-Based Resource Discovery in Pervasive Collaborative Environments", **19th International Conference on Collaboration and Technology, CRIWG 2013**, Lecture Notes in Computer Science, vol. 8224, 30 October - 1 November 2013, Wellington, New Zealand, pp. 233-240. DOI : http://dx.doi.org/10.1007/978-3-642-41347-6_17

Steffenel, L.A., Flauzac, O., Charao, A.S., Barcelos, P.P., Stein, B., Nesmachnow, S. Kirsch Pinheiro, M., Diaz, D., "PER-MARE: Adaptive Deployment of MapReduce over Pervasive Grids", **8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC'13)**, Compiègne, France, Oct. 28-30, 2013, pp. 17-24.

Rychkova I., Kirsch Pinheiro M., Le Grand B., "Context-Aware Agile Business Process Engine: Foundations and Architecture", In: Nurcan, S., Proper, H., Soffer, P., Krogstie, J., Schmidt, R., Halpin, T. & Bider, I. (Eds.), **Enterprise, Business-Process and Information Systems Modeling**, Proceedings of the 14th Working Conference on Business Process Modeling, Development, and Support (BPMDS 2013), Lecture Notes in Business Information Processing, vol. 146, Valence : Espagne, 2013, pp. 32-47.
Classement ERA : C

Kirsch Pinheiro M., Rychkova I., "Dynamic Context Modeling for Agile Case Management", In: Y.T. Demey and H. Panetto (Eds.), **2nd International Workshop on Adaptive Case Management and other non-workflow approaches to BPM (AdaptiveCM 2013)**, OnTheMove Federated Workshop (OTM 2013 Workshops), LNCS 8186, Graz, Austria, 9-13 September 2013, Springer, pp. 144–154, 2013.

2012

Najar, S. Kirsch-Pinheiro, M., Souveyet, C., Steffenel, L. A. "Service Discovery Mechanisms for an Intentional Pervasive Information System", Proceedings of 19th **IEEE International Conference on Web Services (ICWS 2012)**, Honolulu, Hawaii, 24-29 June 2012, pp. 520-527, DOI: 10.1109/ICWS.2012.84
Classement ERA : A

2011

Najar, S., Kirsch Pinheiro, M., Souveyet, C. "Bringing context to intentional services". **3rd Int. Conference on Advanced Service Computing, Service Computation'11**, Rome, Italy, pp. 118-123, 2011. URL: http://www.thinkmind.org/index.php?view=article&articleid=service_computation_2011_5_30_10136 (Best Paper Award)

Najar, S., Kirsch Pinheiro, M., Souveyet, C. “The Influence of Context on Intentional Service”. **5th Int. IEEE Workshop on Requirements Engineerings for Services (REFS’11),** IEEE Conference on Computers, Software, and Applications (COMPSAC’11), Munich, Germany, pp. 470-475, 2011.

Najar, S., Kirsch Pinheiro, M., Souveyet, C. “Towards semantic modeling of intentional pervasive information systems”. **6th International Workshop on Enhanced Web Service Technologies (WEWST’11),** European Conference on Web Services (ECOWS’11), Lugano, Switzerland, pp. 30-34, 2011.

2009

Steffenel, L. A., Kirsch-Pinheiro, M. “Strong Consistency for Shared Objects in Pervasive Grid”. Proceedings of the **5th IEEE International Conference on Wireless and Mobile Computing, Networking and Communication (WiMob'2009)**, Marrakesh, Morroco, 12-14 October 2009. pp. 73-78.
Classement ERA : C

Ait Ali Slimane, A., Kirsch Pinheiro, M., Souveyet, C. “Goal Reasoning for Quality Elicitation in the ISOA approach”, **3rd International Conference on Research Challenges in Information Science (RCIS)**, April 22-24, 2009, Fes, Marocco.
Classement ERA : B

Devlic, A., Reichle, R., Wagner, M., Kirsch-Pinheiro, M., Vanrompay, Y., Berbers, Y., Valla, M. "Context Inference of Users' Social Relationships and Distributed Policy Management". **6th IEEE Workshop on Context Modeling and Reasoning (CoMoRea), 7th IEEE International Conference on Pervasive Computing and Communication (PerCom'09)**, Galveston, Texas, 13 March 2009, pp. 1-8

Vanrompay, Y., Kirsch-Pinheiro, M., Berbers, Y., “Context-Aware Service Selection with Uncertain Context Information”, **Context-Aware Adaptation Mechanism for Pervasive and Ubiquitous Services 2009 (CAMPUS 2009)**, Electronic Communications of the EASST, Volume 19.

Najar, S., Saidani, O., Kirsch-Pinheiro, M., Souveyet, C., and Nurcan, S. “Semantic representation of context models: a framework for analyzing and understanding”. In: J. M. Gomez-Perez, P. Haase, M. Tilly, and P. Warren (Eds), **Proceedings of the 1st Workshop on Context, information and ontologies (CIAO 09), European Semantic Web Conference (ESWC'2009)**, (Heraklion, Greece, June 01 - 01, 2009). ACM, New York, NY, 1-10. DOI=http://doi.acm.org/10.1145/1552262.1552268

2008

Kirsch-Pinheiro, M. Vanrompay, Y,. Berbers, Y. “Context-aware service selection using graph matching”, **2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC'08)**, at ECOWS 2008, Dublin, Ireland, November 12, 2008

Steffenel, L.A.; Kirsch-Pinheiro, M.; Bebers, Y. “Total Order Broadcast on Pervasive Systems”, **23rd Annual ACM Symposium on Applied Computing (SAC 2008)**, Fortaleza, Brazil, March 2008.
Classement ERA : B

Kirsch-Pinheiro, M., Vanrompay, Y. Victor, K., Berbers, Y., Valla, M.; Frà, C., Mamelli, A., Barone, P., Hu, X,. Devlic, A., Panagiotou, G. “Context Grouping Mechanism for Context

Distribution in Ubiquitous Environments", In: Robert Meersman, Zahir Tari et al.(eds.), **10th International Symposium on Distributed Objects, Middleware, and Applications (DOA'08), OTM 2008 Conferences**, Nov 10 - 12, 2008, Monterrey, Mexico.

Kirsch Pinheiro, M., Villanova-Oliver, M., Gensel, J., Berbers, Y., Martin, H. "Personalizing Web-Based Information Systems through Context-Aware User Profiles", **2nd International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2008)**, September 29 - October 4, Valencia, Spain.
Classement ERA : C

Hu, X., Ding, Y., Paspallis, N., Bratskas, P., Papadopoulos, G.A., Vanrompay, Y., Kirsch Pinheiro, M., Berbers, Y., "A Hybrid Peer-to-Peer Solution for Context Distribution in Mobile and Ubiquitous Environments", **17th International Conference on Information Systems Development (ISD2008)**, Paphos, Cyprus, August 25-27, 2008, Spring, pp. 501-510.
Classement ERA : A

2006 Kirsch-Pinheiro, M.; Villanova-Oliver, M.; Gensel, J.; Martin, H. "A Personalized and Context-Aware Adaptation Process for Web-Based Groupware Systems", **4th Int. Workshop on Ubiquitous Mobile Information and Collaboration Systems (UMICS'06), CAiSE'06 Workshop**, Luxembourg, June 5-6, pp. 884-898.

2005 Kirsch-Pinheiro, M., Villanova-Oliver, M., Gensel, J., Martin, H. "Context-Aware Filtering for Collaborative Web Systems: Adapting the Awareness Information to the User's Context", **20th ACM Symposium on Applied Computing (SAC 2005) - Web Technologies and Applications (WTA)**, Santa Fe, New Mexico, Mars 2005, ACM Press, 2005, pp. 1668-1673.
Classement ERA : B

Kirsch-Pinheiro, M., Villanova-Oliver, M., Gensel, J., Martin, H. "BW-M: A framework for awareness support in web-based groupware systems". **9th International Conference On Computer Supported Cooperative Work In Design (CSCWD)**, 2005, Coventry, UK, pp. 240-246.
Classement ERA: B

2004 Kirsch-Pinheiro, M., Villanova-Oliver, M., Gensel, J., Martin, H., "A Context-Based Awareness Mechanism for Mobile Cooperative Users", In: R. Meersman, Z. Tari, A. Corsaro (Eds.), LNCS 3292 - **On the Move to Meaningful Internet Systems 2004: OTM 2004 Workshops: OTM Confederated International Workshops and Posters, GADA, JTRES, MIOS, WORM, WOSE, PhDS, and INTEROP 2004**, octobre 2004. Springer-Verlag, 2004, pp. 9-10.

Kirsch-Pinheiro, M., Gensel, J., Martin, H., "Representing Context for an Adaptative Awareness Mechanism", In: G.-J. de Vreede; L.A. Guerrero, G.M.Raventos (Eds.), **LNCS 3198 - X International Workshop on Groupware (CRIWG 2004)**, San Carlos, Costa Rica, septembre 2004, Springer-Verlag, 2004, pp. 339-348.

Kirsch-Pinheiro, M., Gensel, J., Martin, H., "Awareness on Mobile Groupware Systems", In: A. Karmouch, L. Korba, E.R.M. Madeira (Eds.), **LNCS 3284 - 1st International Workshop on Mobility Aware Technologies and Applications (MATA 2004)**, Florianópolis, Brésil, octobre 2004. Springer-Verlag, 2004, pp. 78-87.

2003 Kirsch-Pinheiro, M., Villanova-Oliver, M., Gensel, J., Lima, J.V., Martin, H. “Providing a Progressive Access to Awareness Information”. In: R. Meersman et al. (Eds.), **LNCS 2888 - CoopIS/DOA/ODBASE 2003**, Springer-Verlag, 2003, pp. 336-353.
Classement ERA : A

2002 Kirsch-Pinheiro, M., Lima, J.V., Borges, M.R.S., “A Framework for Awareness Support in Groupware Systems”, **7th International Conference on Computer Supported Cooperative Work in Design**, Rio de Janeiro, Brazil, 2002. pp. 13-18.
Classement ERA : B

2001 Kirsch-Pinheiro, M., Lima, J.V., Borges, M.R.S. “Awareness em Sistemas de Groupware” (Awareness on Groupware Systems). In**: Proceedings of IDEAS'01**, Centre de Información Tecnológica (CIT), San Diego, Costa Rica, 2001. pp 323-335

National conferences & workshop papers

2019 Kirsch-Pinheiro, M., Souveyet, C. « Le Rôle des Ressources dans l'Évolution des Systèmes d'Information », **Congrès INFORSID 2019 : Informatique des Organisations et Systèmes d'Information et de Décision** 2019: 85-97

Souveyet, C., Villari, M., Steffenel, L.A., Kirsch Pinheiro, M. « Une approche basée sur les MicroÉléments pour l'Évolution des Systèmes d'Information ». **Atelier "Evolution des SI : vers des SI Pervasifs ?", INFORSID 2019**, Paris, Jun 2019, Paris. Disponible sur : https://evolution-si.sciencesconf.org/274949

Steffenel, L.A., Kirsch-Pinheiro, M. « Accès aux Données dans le Fog Computing : le cas des dispositifs de proximité ». **Conférence d'informatique en Parallélisme, Architecture et Système (Compas 2019)**, Jun 2019, Anglet, France

2016 Jaffal, A., Le Grand, B., Kirsch-Pinheiro, M., « Extraction de connaissances dans les Systèmes d'Information Pervasifs par l'Analyse Formelle de Concepts », **Extraction et Gestion des Connaissances (EGC 2016),** Revue des Nouvelles Technologies de l'Information, RNTI-E-30, 2016, pp 291-296.

Steffenel, L.A., Kirsch-Pinheiro, M., « Stratégies Multi-Échelle pour les Environnements Pervasifs et l'Internet des Objets », **11èmes Journées Francophones Mobilité et Ubiquité (UBIMOB 2016).** July 5, 2016. Lorient, France. Disponible sur https://ubimob2016.telecom-sudparis.eu/files/2016/07/Ubimob_2016_paper_6.pdf

2013 Kirsch Pinheiro, M., Le Grand, B., Souveyet, C., Najar, S., « Espace de Services : Vers une formalisation des Systèmes d'Information Pervasifs », **XXXIème Congrès INFORSID 2013 : Informatique des Organisations et Systèmes d'Information et de Décision**, 29-31 Mai 2013, Paris : France, pp. 215-223.

Rychkova, I., Kirsch-Pinheiro, M., Le Grand, B. « CAPE: Context-Aware Agile Business Process Engine », **Workshop GT EASY-DIM 2013 - Vers l’Ingénierie d’Entreprise de demain : les enjeux d’une maquette numérique de l’entreprise**, abstract paper, Mercredi 22 mai 2013, Paris, pp. 8-11

2012

Najar, S. Kirsch-Pinheiro, M., Steffenel, L. A., Souveyet, C., « Analyse des mécanismes de découverte de services avec prise en charge du contexte et de l'intention ». In : Philippe Roose & Nadine Rouillon-Couture (dir.), **8èmes Journées Francophones Mobilité et Ubiquité (Ubimob 2012)**, June 4-6, 2012, Anglet, France. Cépaduès Editions, pp. 210-221. ISBN: 978.2.36493.018.6

Najar, S. Kirsch-Pinheiro, M., Souveyet, C., « Mécanisme de prédiction dans un système d'information pervasif et intentionnel », In : Philippe Roose & Nadine Rouillon-Couture (dir.), **8èmes Journées Francophones Mobilité et Ubiquité (Ubimob 2012)**, June 4-6, 2012, Anglet, France. Cépaduès Editions, pp. 146-157. ISBN: 978.2.36493.018.6

2009

Najar, S., Kirsch Pinheiro, M., Le Grand, B., Souveyet, C., « Systèmes d'Information Pervasifs et Espaces de Services : Définition d'un cadre conceptuel », **UbiMob 2013 : 9èmes journées francophones Mobilité et Ubiquité**, 5-6 juin 2013 Nancy (France), sciencesconf.org:ubimob2013:19119

2008

Gensel, J., Villanova-Oliver, M., Kirsch-Pinheiro, M., « Modèles de contexte pour l'adaptation à l'utilisateur dans des Systèmes d'Information Web collaboratifs », **8èmes Journées Francophones d'Extraction et Gestion des Connaissances (EGC'08), Atelier sur la Modélisation Utilisateur et Personalisation d'Interfaces Web**, Article invité, Sophia-Antipolis, France, janvier 2008.

2005

Kirsch-Pinheiro, M. « Adaptation contextuelle de l'information de conscience de groupe dans les SIW collaboratifs ». **Actes du XXIIème Congrès Informatique des organisations et systèmes d'information et de décision – INFORSID 2005**, Grenoble, France, 2005. pp. 285-300.

Kirsch-Pinheiro, M.; Villanova-Oliver, M.; Gensel, J.; Martin, H. « Une formalisation du cotexte dans les environnements collaboratifs nomades ». **Actes des Deuxièmes Journées Francophones : Mobilité et Ubiquité 2005 (UbiMob'05)**, Grenoble, France, 2005. pp. 1-8.

Reports & dissertations

2013

Kirsch Pinheiro, M., Requirements for Context-aware MapReduce on Pervasive Grids, PER-MARE Deliverable D3.1, 09/07/2013

2006

Kirsch Pinheiro, M. Adaptation contextuelle et personnalisée de l'information de conscience de groupe au sein des Systèmes d'Information Coopératifs. Thèse en informatique, Université Joseph-Fourier, Grenoble, 2006.

2001

Kirsch Pinheiro, M. Mecanismo de Suporte a Percepcao em Ambientes Cooperativos. Dissertation PPGC/UFRGS, Porto Alegre, Brasil, 2001.

1999

Kirsch Pinheiro, M.; Lima, J.V. Edição Cooperativa de Hiperdocumentos na WWW. Rapport technique , PPGC/UFRGS, Porto Alegre, Brasil, 1999.

1997

Kirsch Pinheiro, M.; Augustin, I. Compilação Simulada na Web. Rapport technique, Curso de Informática da UFSM, Santa Maria, Brasil, 1997.