



**HAL**  
open science

## A user-centric vision of service-oriented Pervasive Information Systems

Salma Najjar, Manuele Kirsch Pinheiro, Bénédicte Le Grand, Carine Souveyet

► **To cite this version:**

Salma Najjar, Manuele Kirsch Pinheiro, Bénédicte Le Grand, Carine Souveyet. A user-centric vision of service-oriented Pervasive Information Systems. IEEE RCIS 2014, May 2014, Marrakesh, Morocco. pp.359-370. hal-01020987

**HAL Id: hal-01020987**

**<https://paris1.hal.science/hal-01020987v1>**

Submitted on 15 Jul 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A user-centric vision of service-oriented Pervasive Information Systems

Salma Najjar, Manuele Kirsch Pinheiro, Bénédicte Le Grand, Carine Souveyet  
Centre de Recherche en Informatique, Université Paris 1 – Panthéon Sorbonne  
Paris, France  
Salma.Najjar@malix.univ-paris1.fr, {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 Hagrais [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.

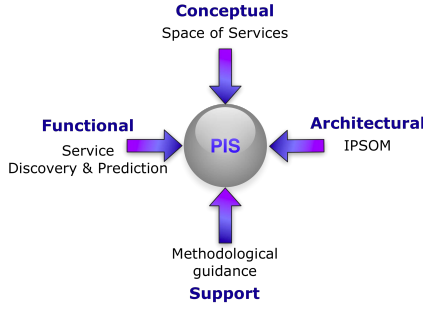


Fig. 1. Four points of views of a Pervasive Information System

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 Services*. 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.

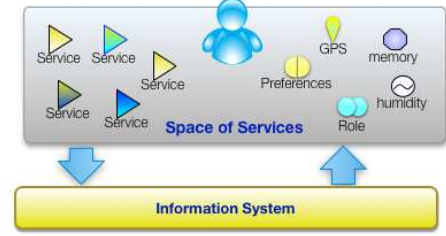


Fig. 2. Space of Services representation

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  $I$ . Each intention  $I_i \in I$  is defined by a verb  $v$  characterizing its action, a target  $tg$  over which action takes place and a set of optional parameters  $par$ .

$$I = \{\langle v, tg, par \rangle\} \quad (2)$$

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.

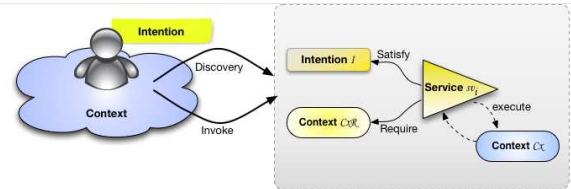


Fig. 3. Relationship among services, intentions, and context information

As illustrated by Fig. 3, a service  $sv_i$  belongs to a given context  $C_x$  (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  $\xi$ . 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  $\mathcal{F}$  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.

$$sv_i = \langle I, \mathcal{F}, Cx, CxR \rangle \quad (3)$$

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  $\mathcal{M}$ .

$$O_{cp_i} = \{ \langle ob_j, t_j \rangle \}, \text{ where } ob_j = \langle s_j, eo, v, \mathcal{M} \rangle, \text{ in which} \quad (4)$$

- $s_j$  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;
- $\mathcal{M}$  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  $O_{cp_i}$ , 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  $O_{cp_i}$  and by the context  $Cx$  also described by a set of observations.

$$cp_i = \{ O_{cp_i}, Cx \} \quad (5)$$

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  $\xi$  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  $\xi$ .

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  $\xi$ , 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  $\mathcal{M}$  that characterize this observation.

$$Cx = \{ \langle ob_j, t_j \rangle \}, ob_j = \langle e_i, eo, v, \mathcal{M} \rangle \quad (7)$$

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.

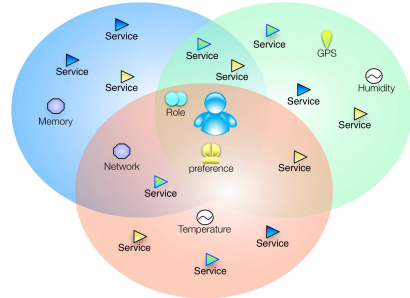


Fig. 4. Multiple spaces of services: Permeability

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  $\xi$  at instant  $t$ , noted  $\xi^t$ , thus corresponds to active and passive entities actually available in the space  $\xi$  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  $\xi$  at instant  $t$ , as defined in (8).

The *state* of a space of services  $\xi$  at instant  $t$ , noted  $\xi^t$ , 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 ( $C_{\mathcal{R}}$ ). 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].

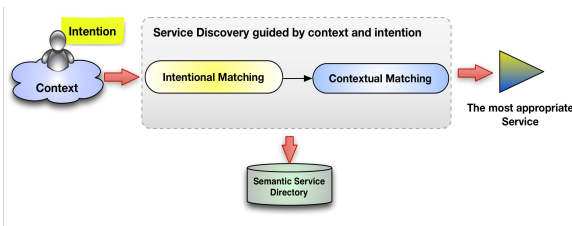


Fig. 5. Schematic view of the Services Discovery mechanism

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  $I_U = \langle \mathcal{V}_U, \mathcal{T}_U \rangle$  and the service's intention  $I_{svi} = \langle \mathcal{V}_{svi}, \mathcal{T}_{svi} \rangle$ .

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  $\mathcal{T}_U$  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 ( $C_{\mathcal{U}}$ ) and service context descriptions ( $C_{\mathcal{R}_{svi}}$ ). The context description for a user ( $C_{\mathcal{U}}$ ) represents a set of context observations  $C_{\mathcal{U}} = \{cx_j | j > 0$  and the context description for a service ( $C_{\mathcal{R}_{svi}}$ ) represents a set of context conditions  $C_{\mathcal{R}_{svi}} = \{cx_i | i > 0$ , 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)

$$Cx_{score} = \sum_{i=1}^n (w * \text{ContextConditionMatching}(cx_i, cx_j)) \quad (9)$$

Thus, in order to define the relation *ContextMatching*, we consider the relation *ContextConditionMatching* that matches individually the different context observations ( $C_{\mathcal{U}}$ ) and context conditions ( $C_{\mathcal{R}_{svi}}$ ). The context matching proceeds as follows: for each observation  $cx_i$  and  $cx_j$ , 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  $cx_i.cd$  with respect to the user's context observations value, one by one. More details about the discovery mechanism are presented in Najjar *et al.* [28].

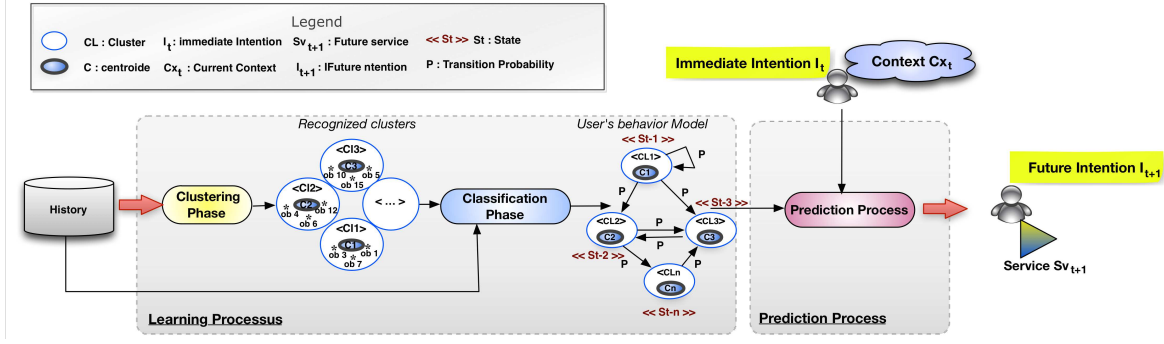


Fig. 6. Schematic view of the services prediction mechanism

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 ( $S_i$ ) as the user's intention ( $I_v$ ), in a given context ( $Cx_v$ ), satisfied by a specific service ( $sv_i$ ) resulting from a previous discovery process, as presented in (10).

$$S_i = \langle I_v, Cx_v, sv_i \rangle \quad (10)$$

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 ( $I_v$ ) and on the current user's context ( $Cx_v$ ). 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)  $St_t$  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  $St_t$  to the situation  $St_j$ . The situation  $St_j$  is the successor state of  $St_t$  with a certain probability  $p$ . This transition probability represents the ratio of the transition from  $St_t$  to  $St_j$  divided by the number of all the possible transitions from  $St_t$ . It is represented in (11).

$$p_{St_t St_j} = P(X_{t+1} = St_j | X_t = St_t) = \frac{N_{St_t St_j}}{N_{St_t St_k}} \quad (11)$$

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.

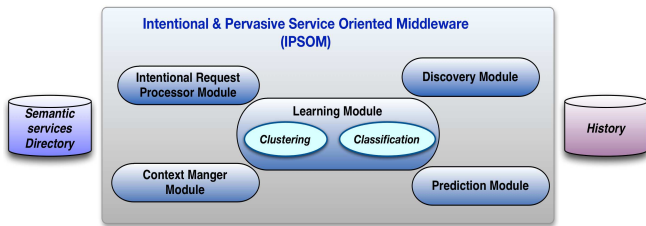


Fig. 7. Intentional & Pervasive Service Oriented Middleware Architecture

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 sent 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.

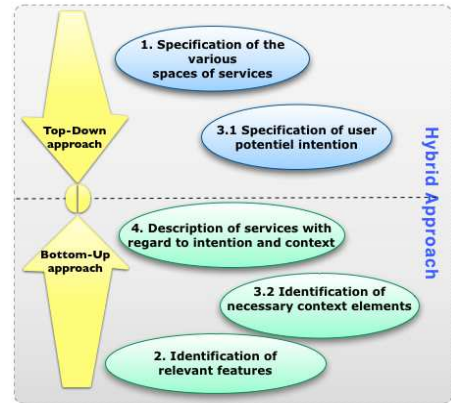


Fig. 8. Methodological guidance for PIS design

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	- <b>AccessClientViewVPN</b> (TunnelVPN, SSLAuthentication, DataEncryption, ClientListPage, ViewClientWS)
Service Sv2	I1 Consult client view	- Device.Network.type ≠ Ethernet - User.Profile.Expertise = High - User.Role = Commercial	- <b>AccessClientView</b> (SSLAuthenticationWS, FindClientByAddressWS, DetailedClientView)
Service Sv4	I2 Edit propoal	- Device.Network.type = Ethernet - User.location.name = company - Device.memory > 512	- <b>ProposalEditionFax</b> (ProposalEditionWS, FaxService)
Service Sv12	I4 Organize conference	- User.Role = Commercial - User.location.name ≠ Company - Device.Memory > 512 - Device.Network.type ≠ Ethernet	- <b>RequestVideoConference</b> (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

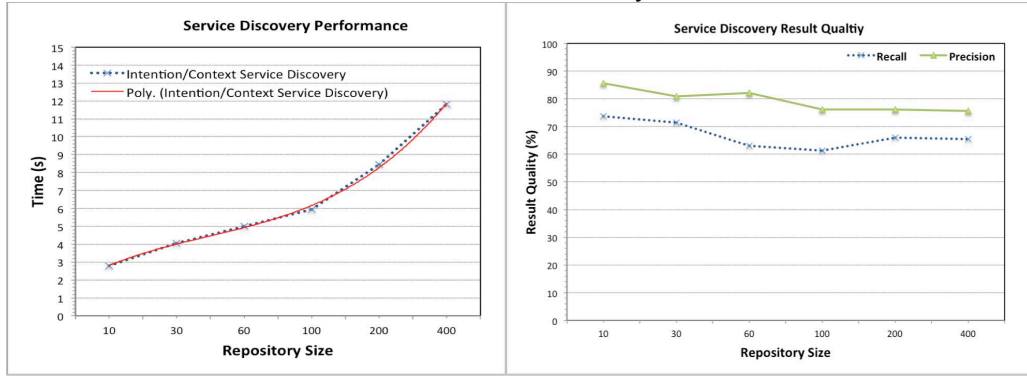


Fig. 9. a) the Service Discovery performance, b) The service discovery result quality

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. 9a, the execution time follows a polynomial trend of degree three. However, even if the execution time is 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. 9b 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

purpose of these experiments is to evaluate the validity of our algorithms and their feasibility. Thus, we formulate a set of user's requests relative to the travel domain. These requests 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. 9b. 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. 10b, 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.

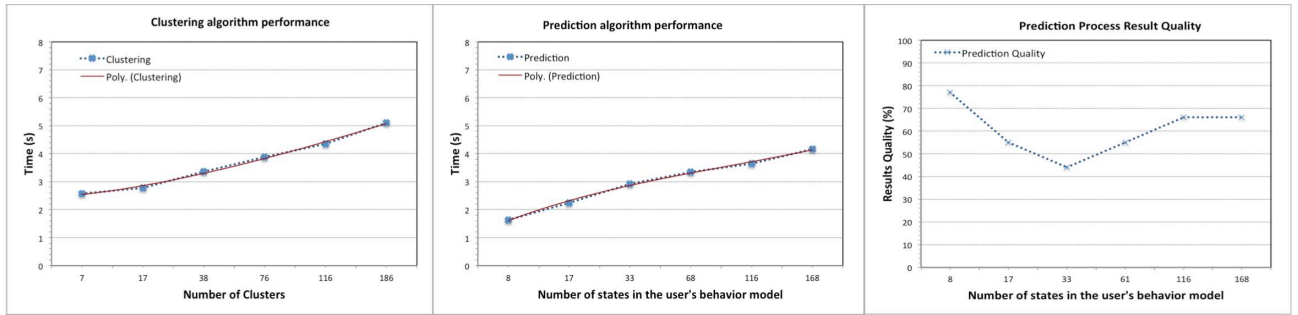


Fig. 10. a) The clustering performance, b) The prediction performance, c) The prediction process results quality

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 10c 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 10c 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 establish 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 their 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. Henriksen, 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](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. Hagrais, "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 Engineering 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 Steffemel, "Service Discovery Mechanism for an Intentional Pervasive Information System," 19<sup>th</sup> 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," 3<sup>rd</sup> 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," 16<sup>th</sup> 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.