



**HAL**  
open science

# Service discovery and prediction on Pervasive Information System

Salma Najar, Manuele Kirsch Pinheiro, Carine Souveyet

► **To cite this version:**

Salma Najar, Manuele Kirsch Pinheiro, Carine Souveyet. Service discovery and prediction on Pervasive Information System. *Journal of ambient intelligence and humanized computing*, 2015, 6 (4), pp.407-423. <10.1007/s12652-015-0288-5>. <hal-01196942>

**HAL Id: hal-01196942**

**<https://paris1.hal.science/hal-01196942v1>**

Submitted on 10 Sep 2015

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.



HAL Authorization

# Service discovery and prediction on Pervasive Information System

Salma Najar, Manuele Kirsch Pinheiro, Carine Souveyet

Centre de Recherche en Informatique / Université Paris 1 – Panthéon Sorbonne

90 Rue Tolbiac, Paris 75013, France

{ FirstName.Last-Name }@univ-paris1.fr

## Abstract

Recent evolution of technology and its usages, such as BYOD (Bring Your Own Device) and IoT (Internet of Things), transformed the way we interact with Information Systems (IS), leading to a new generation of IS, called the Pervasive Information Systems (PIS). These systems have to face heterogeneous pervasive environments and hide the complexity of such environment end-user. In order to reach transparency and proactivity necessary for successful PIS, new discovery and prediction mechanisms are necessary. In this paper, we present a new user-centric approach for PIS and propose new service discovery and prediction based on both user's context and intentions. Intentions allow focusing on goals user wants to satisfy when requesting a service. Those intentions rise in a given context, which influence the service implementation. We propose a service discovery mechanism that observes user's context and intention in order to offer him/her the most appropriate service satisfying her/his intention on the current context. We also propose a prediction mechanism that tries to anticipate user's intentions considering the user's history and the observed context. We evaluate both mechanisms and discuss advanced features future PIS will have to deal with.

**Keywords.** Pervasive Information System, context-awareness, service discovery, service prediction, clustering, classification.

## 1. Introduction

New technologies transform the way we interact with Information Systems (IS) and the services they offer, expanding the frontiers of IS outside the companies' environment. The BYOD (Bring Your Own Device) illustrates 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. *BYOD lets employees use their personal device to work seamlessly across their personal user space and enterprise workspace instead of using multiple devices depending on business need, location, and circumstances* (Chang et al. 2014). This tendency does not concern only laptops. Employees are increasingly using tablets, smartphones and other "thin clients" to access network (LAN, WAN and VPN), and hosted applications, as well as applications that run locally on (some of) the devices. There is also a growing opportunity for businesses to exploit the capabilities of wearable devices, such as Microsoft HoloLens, Apple iWatch or Bluetooth headsets, among countless other devices that are changing the way we search, navigate, transact, and live (Andriole and Bojanova 2014).

These new usages are changing the panorama of IS (Information Systems) and IT departments. Several challenges raise from these, such as the management of heterogeneous hardware and software, or the enforcement of security policies on this heterogeneous environment (Chang et al., 2014; Early et al. 2014). Nevertheless, this new panorama also offers advantages and opportunities for business companies, such as reducing technology costs and increasing employees' productivity (Andriole and Bojanova 2014).

According to Earley et al. (2014), this blending of personal and business technology is having a significant impact on corporate IT departments, which traditionally have governed, deployed, and controlled the technology that employees use to do their jobs. Nowadays, IT is about the integration of emerging business models and technologies; it is now a permanent partnership with the business (Andriole and Bojanova 2014). The consequence of this new panorama is a necessary evolution of IS that have to cope with a pervasive environment and may integrate services from very different natures. A new generation of IS is rising, the Pervasive Information Systems (PIS). PIS should be adapted to this new extended view of the enterprise and of its IS. Indeed, according to Castro-Leon (2014), the notions of what is inside or outside the enterprise become blurry, with corporate processes using resources beyond the traditional enterprise perimeter. In this environment, the user community is not concerned about distinguishing between internal and external resources; users expect to use the devices, data, and tools in the same way they always have.

Pervasive Information Systems (PIS) intend to increase user's productivity by making IS services available anytime and anywhere. These systems changed the interaction paradigm from desktop computing to new technologies. They evolved from a fully controlled environment (the office) to a dynamic pervasive one (Kourouthanassis and Giaglis 2006). Contrary to traditional IS, PIS have to support a multitude of heterogeneous devices and service types, challenging its design. We argue that PIS should be designed for helping user to better satisfy her/his needs according to her/his environment. PIS must not only consider the goals they must reach as an IS, but they must also handle heterogeneity of pervasive environment. They should hide this heterogeneity from the user, allowing her/him to concentrate on her/his needs, instead of on the technology itself.

By extending its influence outside traditional enterprise perimeter, PIS are able to integrate services that are made available by pervasive environments frequented by users, wherever they are. Available services, as well as their nature, may vary according to the environment they integrate. This complexity is not necessarily understood by users, who are not interested in knowing service alternatives, but only on having an appropriate service. Discovering best-suited service becomes then a challenging task since it is necessary to observe not only user's goals, but also the circumstances under which such goals appear. Besides, a more proactive behavior should also be considered. Indeed, user expects a PIS that is able to anticipate her/his needs, in a transparent way, for more productivity. Thus, transparency and proactivity become key aspects on PIS, which require offering user appropriate services considering her/his goals and the context in which such goals appear, as well as the capability of anticipating future goals in this context. New service discovery and prediction mechanisms that cope with such requirements are then necessary.

We propose a new user-centric approach for service discovery and prediction considering PIS. This approach is based on both user's intentions, representing the goals she/he wants to achieve without saying how to perform it (Kaabi and Souveyet 2007), and on the context in which these intentions have been formulated. The notion of context can be seen as any information that can be used to characterize the situation of an entity (Dey 2001). We consider that context information can influence service execution and, consequently, what service can be chosen to satisfy a given intention. In our opinion, both concepts should be considered during service discovery, since the main purpose is providing user with a service that can fulfill her/his goals in a fairly understandable and non-intrusive way. Thus, we propose a new service discovery mechanism that intends discovering the available service that can satisfy the immediate user's intention in the current context. Based on the discovery results, we propose a new prediction mechanism that identifies common situations representing usage patterns, i.e., recurrent context and intentions observed during the PIS usage. By analyzing these patterns, the prediction mechanism learns user's behavior when using a PIS, and therefore anticipates future intentions and the most appropriate services that may satisfy it.

This paper is organized as follows: Section 2 presents related works on service discovery and prediction. Section 3 introduces the user-centric approach we propose for PIS. Based on this approach, Section 4 presents the proposed service discovery mechanism, while Section 5 introduces the service prediction mechanism. Section 6 presents an evaluation of both mechanisms. Finally, Section 7 discusses future directions, before concluding in Section 8.

## 2. Related works

Applications and services are no longer limited to the strict boundaries set by the desktop computer. Users are interested in having access to those services independently of whether these services are provided via a desktop computer, a mobile phone, or an information kiosk located in a public place (Kourouthanassis and Giaglis 2006).

In order to make this ubiquitous access possible, Pervasive Information Systems (PIS) have to conciliate two different worlds: the dynamic and heterogeneous pervasive environments with the predictable and expected behavior necessary to an Information System. From this analysis, a set of requirements applying to PIS can be delineated, including heterogeneity (being able to handle heterogeneity of devices and services), predictability (being able to satisfy user's goals in a predictable and expected manner) and context awareness (Najar et al. 2014). Context-awareness can be seen as the capability pervasive artifacts have to collect, to process, and to manage environmental or user-related information on a real-time basis (Kourouthanassis and Giaglis 2006). It has been defined by Baldauf et al. (2007) as the ability of a system to adapt its operations to the current context, aiming at increasing its usability and its effectiveness by taking environmental context into account. In contrast to desktop computing, where user's action precedes system response, PIS promote system pro-action based on environmental stimuli (Kourouthanassis and Giaglis 2006). Thus, PIS should be able to observe changes in the execution environment and to adapt consequently their behavior (Najar et al. 2014).

During the last decade, a lot of research has been conducted concerning pervasive systems, and notably several works concerning context-aware services (Chaari et al. 2007; Toninelli et al. 2008; Vanrompay et al. 2011). On the one hand, context-awareness becomes a necessary feature for providing 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 (Eikerling et al. 2007). On the other hand, loose coupling offered by services fits the requirements of high dynamic pervasive environments, in which entities are often mobile, entering and leaving the environment at any moment (Kirsch-Pinheiro et al. 2008).

Different service discovery mechanisms have been proposed in the literature (Toninelli et al. 2008; Vanrompay et al. 2011; Mokhtar et al. 2006; Mokhtar et al. 2008). Main goal of such works is to propose services that are better adapted to the user's requirements by observing real conditions under which such services are invoked. According to Suraci et al. (2007), context-aware service discovery can be defined as the ability to make use of context information in order to discover the most relevant service for the user. Most of the proposed context-aware service discovery mechanisms consider context information as a non-functional aspect of a service (Toninelli et al. 2008; Mokhtar et al. 2006), or as a condition for service selection and execution (Vanrompay et al. 2011). On both cases, a matchmaking, using semantic matching (Mokhtar et al. 2006) or similarity measures (Vanrompay et al. 2011), is performed between context information related to the service and the one related to the user or to the execution environment.

Often on those works (Toninelli et al. 2008; Mokhtar et al. 2006; Mokhtar et al. 2008; Cubo and Pimentel 2012), context information is considered as required capabilities imposed by the user's request. Context information and the service capabilities are semantically described thanks to multiple ontologies. Such ontologies allow describing service functional properties (mainly input and output information) and describing context elements and values. Based on these ontologies, different semantic matching mechanisms have been proposed (Toninelli et al. 2008; Mokhtar et al. 2006; Cubo and Pimentel 2012). Most of them are based on simple relations of *subsumes* and *plugins* between concepts on an ontology. Toninelli et al. (2006), for example, consider *exact*, *subsumes* and *plugin* relations between capabilities offered by the services and those required by the user, both represented as ontology concepts. Similarly, Mokhtar et al. (2008) adopts the same relations in order to discover services that propose exactly the required capabilities (*ExactCapabilityMatch*), or services that offer more general capabilities than those required by the user (*InclusiveCapabilityMatch*) or more specific

ones (*WeakCapabilityMach*). Additionally, Mokhtar et al. (2006) also consider to calculating a semantic distance, still based on ontologies, between concepts used to describe offered and required capabilities.

In addition to user's request, other authors, such as Suraci et al. (2007) and Vanrompay et al. (2011), also consider that services have their own requirements on the context information they need from user or from the environment in order to work properly. For Suraci et al. (2007), a service can require the user to provide specific context information, like her/his location or some terminal capabilities. These authors propose to enrich OWL-S service description with a "context" element representing such requirements. A similar approach is adopted by Vanrompay et al. (2011), which consider that a service should expose both its current execution context (representing context under which it executes) and its context requirements (context under which the service is expected to be invoked by the user). Both are represented as graphs and compared with user's context and requirements using different kinds of similarity measures: local measures comparing context elements (nodes), and global measures comparing graphs overall structure.

Nevertheless, only a few research works (Fensel et al. 2011; Olsson et al. 2011; Santos et al. 2009) consider the notion of intention on service description. Intentions represent a high level description of user's goals, delineating her/his requirements about the service (why she/he needs the service), without specifying how these requirements could be achieved. Intentions can be associated with service descriptions as a set of capabilities, with their pre- and post-conditions (Fensel et al. 2011). By considering the notion of intention as part of service definition, works such as (Fensel et al. 2011; Kaabi and Souveyet 2007; Santos et al. 2009; Olsson et al. 2011) allow users to focus on their own goals instead of considering the technology itself, which represent an important issue for achieving transparency necessary to PIS. Those works consider often intentions as a guide for service discovery. For instance Olsson et al. (2011) proposes a service discovery mechanism based on a refining process in which intentions are decomposed on low-level intentions. Similarly, Santos et al. (2009) propose to associate user's intentions with possible tasks that a service is able to perform for the users, allowing then a service discovery based on such intentions.

Unfortunately, the influence of context on the intention satisfaction is merely considered on the literature, context being often seen as a simple input on intention-based mechanisms (Santos et al. 2009).

A similar situation can be observed when considering service prediction. Anticipating user's needs represents an important step towards transparent PIS. A proactive behavior can offer new possibilities to enhance available informational services or construct new currently unavailable ones (Mayrhofer 2005). Indeed, when interacting with Information Systems, users develop often work habitudes that can be understood as work patterns. Discovering and anticipating user's actions based on such patterns may lead to a more proactive behavior, necessary to obtain transparent PIS.

Until now, several works have considered context prediction (Mayrhofer 2005; Mayrhofer 2004; Ali et al. 2008; Sigg et al. 2010, König et al. 2011) or service recommendation based on context information (Adomavicius and Tuzhilin 2011; Xiao et al. 2010).

On the one hand, several works propose anticipating context information based on historical data (Mayrhofer 2004; Vanrompay and Berbers 2012) or on pattern matching (Sigg et al. 2010). For instance, Mayrhofer (2004) uses recommendation techniques for anticipating context information and for predicting the next likely situation of the user. This author proposes a non-supervised classification method for clustering observed context information. Identified clusters are considered as states, allowing interpreting new context information as a possible state changes for prediction purposes. Vanrompay and Berbers (2012) also propose to anticipate context information, but contrarily to previous work, these authors consider analyzing the quality of context information during the prediction process. According to Vanrompay and Berbers (2012), low quality context information can lead to a wrong prediction, which can be costly and reduce user satisfaction. They propose multiple learning operators that are able to consider uncertainty of context information when classifying it, identifying representative patterns on it. Different from previous authors, Sigg et al. (2010) suggest to recommend context information in order to fulfill context description with missing elements based on

similar previously observed contexts. These authors use temporal series in order to identify context patterns. Such patterns are used during an alignment method, which looks for most probable series that matches current observed sequence. Similarly, König et al. (2011) also proposes an align method, but, different from Sigg et al (2010), these authors propose a multi-dimensional algorithm, allowing authors to better consider correctness and accuracy. Finally, works such as Ramakrishnan et al. (2013; 2014) consider combining different recommendation approaches, such as Bayesian networks and A-Priori algorithm, in order to recognize possible correlation among different context elements. Similarly, Nazerfard and Cook (2015) also propose a prediction mechanism based on Bayesian networks. These authors consider Bayesian networks on a model, named CRAFT, used for activity prediction, and more precisely, for predicting the start time of the next activity, in the context of smart homes.

On the other hand, service prediction works proactively propose services based on user's historical context information. Similar to traditional recommender systems, such as Javari et al. (2014), that consider the correlation among user and items, service prediction works, such as Adomavicius and Tuzhilin (2011), Xiao et al. (2010) and Baltrunas and Ricci (2013), consider the correlation between context information and an item (e.g. a service) using different filtering techniques (Baltrunas and Ricci 2013), which can be correlated with ontology-based matching (Xiao et al. 2010). According Adomavicius and Tuzhilin (2011), it is possible to organize such recommender systems on to three categories: 1) pre-filtering, in which context information is used to filter out irrelevant ratings before they are used for computing recommendations; (2) post-filtering, in which context information is used after non-contextual recommendation methods; and (3) contextual modeling, in which context information is used inside the recommendation algorithms with the user and item data. Cremonesi et al. (2011) propose a post-filtering recommendation approach that uses association rules to identify the most significant correlations between a given context information and an item. These rules are then used to filter the predictions performed by traditional recommender systems. Similarly, Baltrunas and Ricci (2013) propose an approach named "Item Splitting", in which items experienced in two alternative contextual conditions are "split" into two items that are then used in a rating algorithm. Finally, Xiao et al. (2010) propose using ontology for improving semantic associated to context values on recommendation approaches. Instead of using "if-then" rules, these authors propose discovering emerging relationships between observed context values using knowledge available on the ontologies.

Besides context prediction and service recommendation, other works (Foresti et al. 2015; Paridel et al. 2014; Basiri and Malek 2014) have considered classification techniques for anticipating user's situation. For instance, Foresti et al. (2015) have proposed an architecture for situation awareness on emergency scenarios that aggregates data acquired by a smart environment and user generated. Paridel et al. (2014) propose a context grouping mechanism that allows clustering vehicles on VANETs according their location and direction. Finally, Basiri and Malek (2014) suggest a fuzzy analysis of the location information, proposing a better interpretation of the proximity (the "here") concept.

Unfortunately, at the best of our knowledge, none of these works has proposed combining intention aspect with context information in a proactive or anticipatory behavior. In general, the notion of intention, representing concrete user's goals, remains totally unexplored by these works. This may represent an important drawback, since prediction is performed based only on technical solutions, ignoring user's intentions behind (and guiding) user's actions. Indeed, by ignoring user's goals, these approaches may improve the risk of recommending irrelevant services for users.

### **3. Meeting Service Context and Service Intention**

Pervasive Information Systems (PIS) have to face complex pervasive environments, in which heterogeneous technologies cohabit and interact with each other and with the system itself. Besides, a PIS is supposed to behave as an Information System (IS), allowing its users to accomplish their business goals through the services it proposes. A PIS can thus be characterized by this dual nature: it must be considered in terms of IS and as a pervasive system.

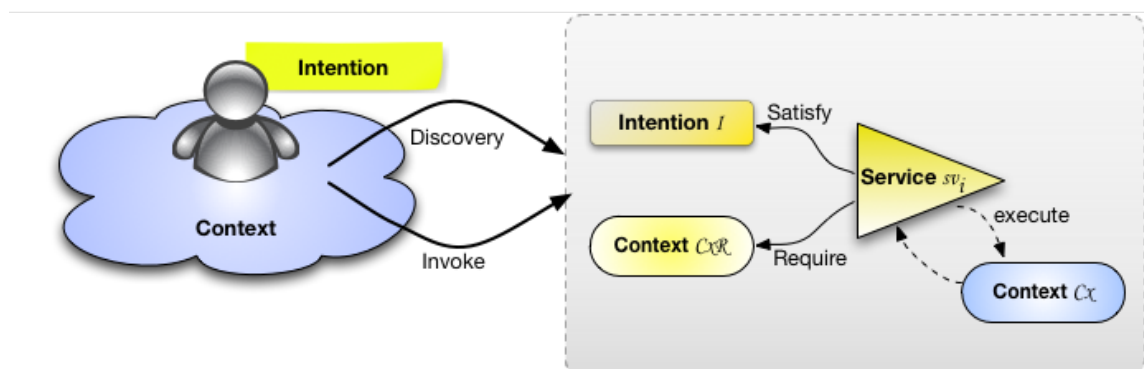
As a pervasive system, a PIS must handle dynamicity and heterogeneity of pervasive environments. According to Capilla et al. (2014), context information determines the environment in which a system operates. According to these authors, because different contextual conditions can provoke different system responses, suitable runtime adaptation mechanisms and intelligent decision making may enable systems to successfully handle the demands and the complexity of dynamic variability and adaptation. The context-aware nature of PIS is then ineluctable.

As an Information System, a PIS must consider goals users want to achieve by executing services it offers. Indeed, an IS is designed to support precise business goals, proposed as part of the overall organization strategy. These goals are independent from the technology used to implement them. Users should focus on these goals and not on the technology itself.

We believe that in order to handle such dual nature, a PIS must reduce the user's understanding effort, augmenting its transparency, allowing users to totally focus on the goals they want to achieve. A PIS must hide the complexity of the multiple available services it offers. This transparency will be only possible thanks to a user-centered vision. In this paper, we adopt a user-centric vision of PIS that is based on a close relationship between the notions of *Intention*, *Context* and *Service*. This vision allows, on the one hand, to consider 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.

We consider that a service offered by a PIS is proposed in order to satisfy a given user's goal, corresponding to the user's needs. In other words, we consider that *services* should be associated with the *intentions* they allow users to satisfy. Such intentions emerge in a given *context*, which should be observed in order to fully satisfy such an intention. Indeed, we consider that a user does not require a service just because she/he is located in a given place or under a given context. For us, the user does not require a service because she/he has an intention that a service can satisfy in this context. The context in which an intention emerges has an important impact on its satisfaction. A given intention may be satisfied by different means according to the context in which it emerges, since different services can be proposed according to the available environment, representing an important source of variability for a PIS.

This vision, illustrated by Fig. 1 and detailed on Najar et al. (2014), consider that a service is proposed in order to satisfy a given intention  $I$  under a given context  $C_{\mathcal{R}}$ . This later represents the context in which the service is supposed to be invoked in order to work properly. In other terms, the required context  $C_{\mathcal{R}}$  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's context and the required context  $C_{\mathcal{R}}$  is, the higher the chances of adapting it to the current situation and of satisfying the user. Moreover, we also consider that a service is performed under a context  $C_{\mathcal{X}}$  which indicates the execution conditions of the service in the provider side.



**Fig. 1 Relationship among context information, service and user's intention (Najar et al. 2014)**

We present, in this paper, two mechanisms that explore this user-centric vision. First, we propose a discovery mechanism (cf. Section 4) that is able to discover most appropriate service capable of satisfying current user’s intention under a given context. Secondly, we propose a prediction mechanism (Section 5) that will try to anticipate user’s future intention, based on current user’s intention and context, in order to proactively offer her/him the next service according her/his habits in the current context.

#### 4. Service Discovery Mechanism

Based on our user-centric approach (cf. Section 3), we propose a service discovery mechanism guided by user’s intention and context. Its objective is to hide implementation complexity, and consequently to achieve the transparency promised by PIS. This service discovery mechanism selects the most appropriate service for a given user, i.e., the service that satisfies her/his immediate intention in a given context. It is based on a semantic service description and on a semantic matching algorithm. This service description, which we detail in Najar et al. (2012a), extends service description on OWL-S in order to include the intention the service is able to satisfy, called  $I_{svi}$ , and the description of the required context associated with the service, called  $C_{\mathcal{R}_{sv}}$ . 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 the user’s current context with the service required context. The service having the highest matching score is selected. It represents the most appropriate service that satisfies the user’s immediate intention in her/his current context.

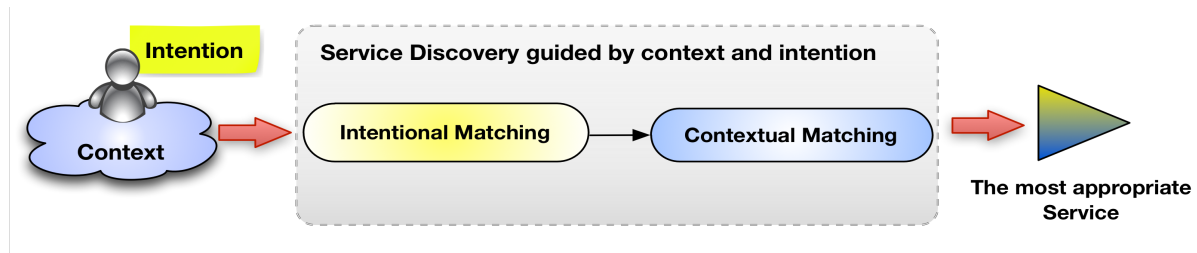


Fig. 2 Schematic view of the Services Discovery mechanism

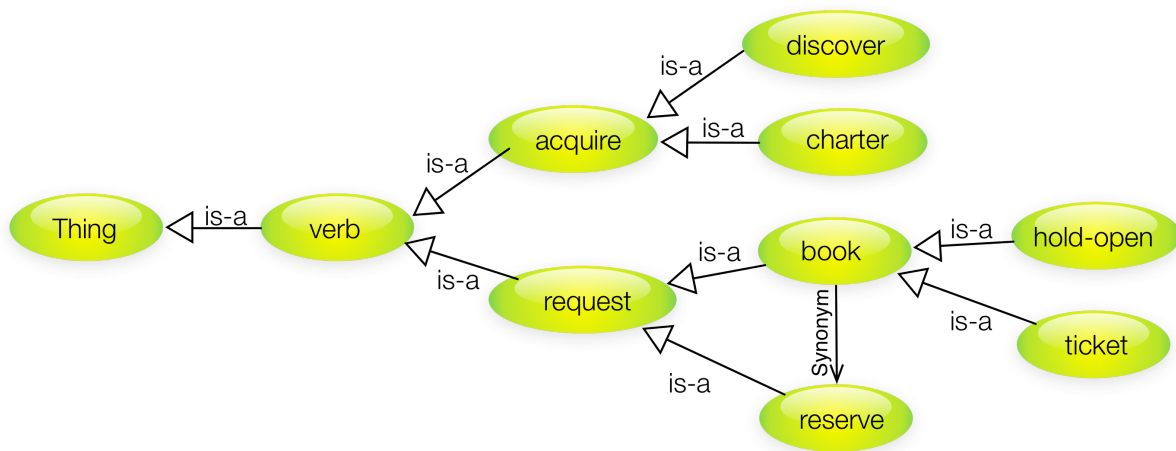
The semantic matching algorithm, as illustrated in Fig. 2, is a two-step process: *intention matching* and *context matching*. The first step, *intention matching*, compares the user’s intention with the service one. It is based on a set of ontologies and on a semantic matching. An intention is represented as a n-uplet  $I = \langle \mathcal{V}, \mathcal{T} \rangle$  composed of a verb ( $\mathcal{V}$ ) and a target ( $\mathcal{T}$ ) representing user’s goals (Kaabi and Souveyet, 2007; Najar et al. 2012b). In this step, we propose to compare the user’s required intention  $I_u = \langle \mathcal{V}_u, \mathcal{T}_u \rangle$  and the intention  $I_{svi} = \langle \mathcal{V}_{svi}, \mathcal{T}_{svi} \rangle$ <sup>1</sup> associated with a service  $svi$  in two separated matching process. For the *verb matching*, we use a verb ontology (named  $Onto\mathcal{V}$ ), which contains a domain-specific set of verbs, representing significant actions authorized by the PIS, with their different meanings and relations. A degree of similarity between the verb  $\mathcal{V}_u$ , present in the user’s intention  $I_u = \langle \mathcal{V}_u, \mathcal{T}_u \rangle$ , and the verb  $\mathcal{V}_{svi}$ , offered by a given service intention  $I_{svi} = \langle \mathcal{V}_{svi}, \mathcal{T}_{svi} \rangle$  is calculated based on the distance between these verbs in the verb ontology:  $score = (1/L + 1)$ , where L represents the number of links between two concepts in the ontology. We define five levels of similarity, inspired from Paolucci et al. (2002), as explained in Table 1. For instant, Fig. 3 illustrates a fragment of a verb ontology related to tourism application domain. In this ontology, the verb “book” is related to the verb “reserve” by a “synonym” relation and related to the verb “ticket” by a “hypernym” relation (indicated by the “is-a” link). This means that if the user demands an intention containing the verb “book”,

<sup>1</sup> In this paper, we use the indices  $u$  and  $svi$  in order to distinguish information related to the user (index  $u$ ) from the information obtained from the description of a given service (index  $svi$ ).

services containing the verb “reserve” will obtain a score 0.9 on the verb matching, while those containing the verb “ticket” will obtain a score of 0.5 (1/2).

**Table 1. 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       |



**Fig. 3 Example of a verb ontology (fragment)**

Similarly, during the *target matching*, we use a domain-specific ontology, namely *OntoT* for target ontology. This ontology represents the possible targets that are made available through the PIS. We compare the required target  $T_u$ , proposed in the user’s intention, with the target  $T_{svi}$  provided by the intention  $I_{svi}$  offered by a service *svi*. For this comparing both targets, we use a degree of similarity also based on the distance between these concepts in the ontology. This semantic similarity, based on Paolucci et al. (2002), proposes four levels of similarity, represented in Table 2: *exact*, *plugin*, *subsume* and *fails*. As explained in Table 2, the *plugin* relation is similar to the *hyponym* relation in the verb matching, while the *subsume* relation is similar to the *hypernym* relation. More details about the intention matching can be found in Najar et al. 2012b.

**Table 2. Target matching relations**

| Matching Relation | Explanation  | Link | Score   |
|-------------------|--|------|---------|
| Exact             | Required target is equivalent to the provided target   | 0    | 1       |
| Plugin            | Required target is more specific than the provided one | L    | 1/(L+1) |
| Subsume           | Required target is more generic than the provided one  | L    | 1/(L+1) |
| Fail              | No relation between the two verbs                      | -1   | 0       |

The second step, *context matching*, is based on the context ontology (named *OntoCX*) and on a set of similarity measures. These measures match individually the different context elements constituting the user's context description, identified as  $Cx_u$ , and the service required context description, identified as  $Cx_{\mathcal{R}_{svi}}$  for a given service *svi*. Indeed, context information is often semantically represented using ontologies in which context information is structured (e.g. Toninelli et al. 2008; Mokhtar et al. 2006; Vanrompay and Berbers 2012). In our case, context information is represented as context elements (location, available memory, user's expertise, etc.) that are observed from a given subject (a user, a device, etc.). Both context elements and subjects are semantically described using ontologies. The context description for a given user ( $Cx_u$ ) represents a set of context observations ( $cx_j$ ) associated with this user  $Cx_u = \{cx_j \mid j > 0\}$ , in which each  $cx_j$  indicates the value observed for a given context element. The required context for a service ( $Cx_{\mathcal{R}_{svi}}$ ) describes a set of context conditions  $Cx_{\mathcal{R}_{svi}} = \{cx_i \mid i > 0\}$  expressed over context elements. The former corresponds to the current observed user's context, while the latter represents contextual conditions for which service was designed (i.e. context under which service is able to better satisfy its intention).

Each context condition  $cx_i$  is compared to the context observations  $cx_j$ , resulting in a *context matching* score  $Cx_{score}$ . This score is calculated as the sum of the scores of each context condition, as follows:

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

As indicated above, we consider the relation *ContextConditionMatching* that individually matches the observations  $cx_j$ , obtained from the user's context description  $Cx_u$ , and the context conditions  $cx_i$ , indicated in the service required context  $Cx_{\mathcal{R}_{svi}}$ . The observable context elements can be divided into several types. Their values can be distinguished between numerical or non-numerical types. In order to take into account this diversity, the relation *ContextConditionMatching* identifies the nature of the context element and accordingly triggers the suitable measure to compare them. These measures evaluate if the user's context element satisfies the service context condition, based on a specific operator (equal, not-equal, between, higher-than, etc.). For example, a service context condition can be having the device bandwidth higher than 12500. From the user's current context, if the observed value of the user's device bandwidth is really higher than 12500, then evaluating measure will return an exact match.

The overall context matching process proceeds as follows: for each condition  $cx_i$  and an observation  $cx_j$ , we first match the corresponding subjects, using the context ontology. This match is calculated, like the verb and target matching, based on the distance between both concepts in the context ontology. If this matching score is higher than a given threshold, then we match the context elements associated with this subject. This last matching takes also into account a weight assigned to the context elements. If the matching score between them is higher than a given threshold, only at this moment we evaluate the satisfaction of the context condition regarding to the user's context observations value one by one.

Besides, the weight ( $w$ ) corresponds to a weight assigned by the user to each context element 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. More details about the context matching are presented in Najjar et al. (2012b).

The proposed service discovery mechanism allows determining the most appropriate service considering immediate user's intention and her/his current context. Nevertheless, this mechanism corresponds to a reactive behavior, since discovery process is launch by an active service request made by the user. User actively demands to the service discovery mechanism to satisfy a given intention. This reactive behavior has to be completed by a more proactive one, in which the system anticipates

the user's next intention. Such a proactive behavior is offered by the prediction mechanism, that we explain in the next section.

## 5. Service Prediction Mechanism

In addition to the service discovery mechanism presented in the previous section, we also propose in our user-centric 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 can be detected, even in a dynamic PIS. We define a situation  $S_i$  as the n-uplet  $S_i = \langle I_u, C\chi_u, S v_i \rangle$  composed by a given intention  $I_u$  expressed by the user in a given context  $C\chi_u$  and satisfied by a specific service ( $S v_i$ ) resulting from a previous discovery process. These situations are time-stamped and stored in a database after each service discovery process, forming a history. The user's history  $\mathcal{H}$  is defined as a set of all the observed situations  $S_i$  ordered according to their time of occurrence. Thus, by analyzing the history  $\mathcal{H}$ , the prediction mechanism can learn the user's behavior model, called  $\mathcal{M}_c$ , in a dynamic environment, and thus deduce its coming immediately intention.

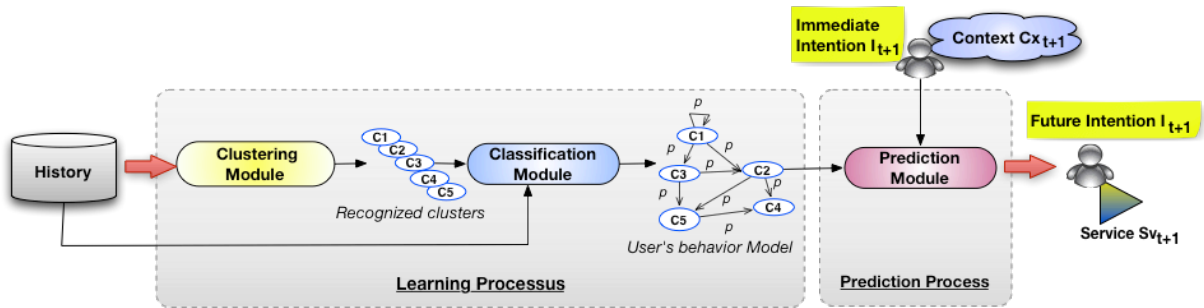


Fig. 4 Schematic view of the Services Prediction mechanism

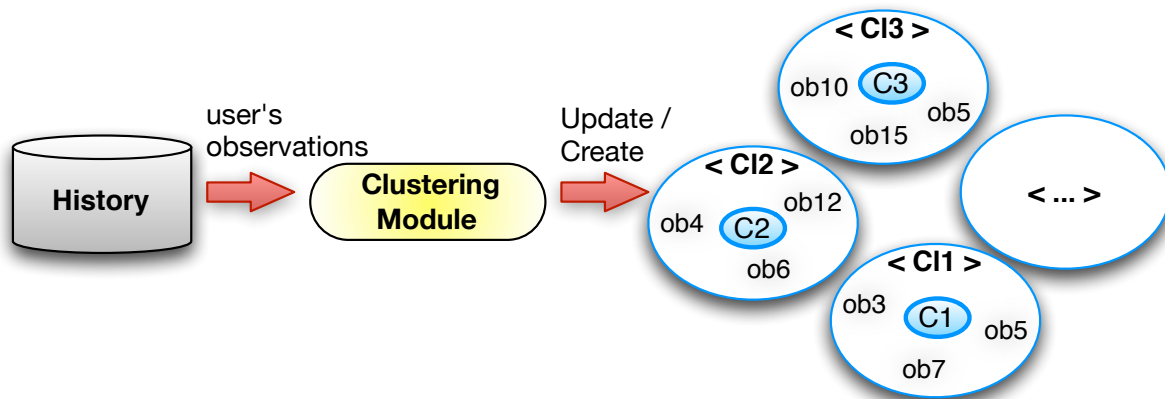
Two main processes compose the service prediction mechanism, as illustrated in Fig. 4: the *learning process* and the *prediction process*. Indeed, to realize anticipatory and proactive behavior of PIS, we need first to dynamically learn about the user and her/his behavior in a frequently changing environment. This represents the *learning process*, in which similar situations are grouped into clusters, during the phase of *clustering*. In the next step, these clusters are interpreted as states of a state machine. This phase is called *classification*. It aims at representing, from the recognized clusters, the user's behavior model ( $\mathcal{M}_c$ ) based on her/his situations ( $S_i$ ). By interpreting situation changes as a trajectory of states, we can anticipate her/his future needs. Therefore, the *prediction process* will try to anticipate the user's next situation based on the user's behavior model ( $\mathcal{M}_c$ ), resulting from the learning process, on the current user's intention ( $I_u$ ), as well as on the current user's context ( $C\chi_u$ ).

It is worth noting that, in order to represent a situation, we attach a particular *service* to the couple  $\langle Intention, Context \rangle$ , forming the tuple  $S_i = \langle I_u, C\chi_u, S v_i \rangle$ . We are aware that this represents a strong constraint, since the concept of 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. Indeed, the prediction mechanism will determine the next probable situation of the user, including her/his coming intention and context, as well as the service that has been used to satisfy this intention in this precise context. Thanks to the presence of the service in the concept of situation, a new discovery process is not necessary, since we have already an appropriate service for this couple  $\langle Intention, Context \rangle$ . As a consequence of such a definition, it is important to regularly update the clusters in order to have the service that best matches the couple intention and

context associated with the situation. Learning process is thus an off-line process that should be regularly executed, independently of the prediction process.

The main role of clustering is to detect recurrent situations among those previously observed and grouped in a *cluster*. A cluster consists of a *centroid* and a set of situations, both defined by a n-uplet  $\langle I_u, C_{X_u}, S_{V_i} \rangle$ . The centroid represents the identifier of the cluster, which symbolizes the situation the most similar to all the situations grouped in this cluster. The main task of the *clustering phase*, as illustrated in Fig. 5, is then to detect, for a given situation, the closest set of situations corresponding to highly similar intentions in a 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 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. The same applies to context information, since an intention may rise on similar contexts.

More precisely, the input of the *clustering phase* consists of a set of vectors representing user's situations stored in the history. For each new observed situation in the user's history, the clustering phase determines closest cluster by comparing intention and context from this situation with those coming from the centroid of the identified clusters. In order to compare them, we use the same intention matching and context matching used for the discovery phase. If this matching is below a given threshold, the situation is included in the closest identified cluster. If no existent cluster is enough similar to the situation, a new cluster is created. Thus, after a clustering phase, the corresponding cluster identifier is attached to each new situations stored in the history.



**Fig. 5 Schematic view of the clustering phase**

From these recognized clusters and the user's history, the *classification phase* determines and maintains a user's behavior model, as illustrated in Fig. 6. This model represents the user's behavior as a set of states with a transition probability. Each state is represented by the centroid of the recognized cluster. Each probability is calculated based on the history and determines the probability of moving from one state to another.

Several classification techniques exist. Among these techniques, the Markov chains (Feller 1968) represents one of the well-known classification algorithms that can be used in a PIS. It represents a method for representing a stochastic process in discrete time with discrete state space. We represent the user's behavior model by a Markov chains model ( $\mathcal{M}_c$ ), which is defined 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.

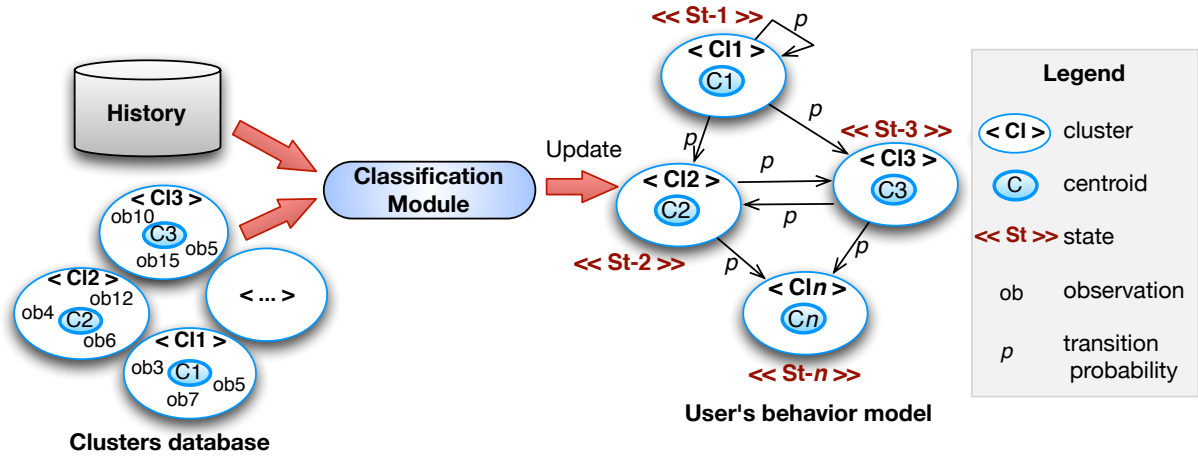


Fig. 6 Schematic view of the classification phase

At a given moment  $t$ , the user is in a state  $St_i$ . In the PIS, the user's intention and/or her/his context may change. Therefore, the user moves from the state  $St_i$  to a state  $St_j$ . The state  $St_j$  represents the successor of  $St_i$  with a certain probability  $p$ . This transition probability represents the ratio of the transitions from  $St_i$  to  $St_j$  divided by the number of all the possible transitions from  $St_i$ . This probability is represented as follow:

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

For each new situations stored in the history, we proceed by selecting for each situation  $St_i$  (identified by its cluster), its successor  $St_j$ . The successor  $St_j$  represents the situation that is directly stored after the situation  $St_i$ . Indeed, by observing timestamp associated with each observed situation individually, it is possible to determine the situation that follows this one, indicating a possible transition between the clusters containing these situations. Then, we calculate the number of the existing transitions from  $St_i$  to  $St_j$  ( $N_{St_i St_j}$ ). Next, for each situation  $St_i$ , we determine the entire possible next situation  $St_k$  ( $N_{St_i St_k}$ ). We note that, in the history database, the former number of transitions from  $St_i$  to  $St_j$  ( $N_{St_i St_j}$ ) and the number of all the possible transitions from  $St_i$  ( $N_{St_i St_k}$ ) are already stored. This information is updated, and all the states and transition probability are represented and calculated accordingly. The Markov chains model resulting from this process represents the user's behavior model ( $\mathcal{M}_c$ ) that is used during the prediction process.

The *prediction process* is thus mainly based on the results of the *classification phase* in order to predict the next user's intention and service. This prediction process, represented by the algorithm in Table 3, is based on the semantic matching between the user's immediate intention and context and those of the states represented in the user's behavior model. Similar to the discovery mechanism, the semantic matching is based on ontologies in order to calculate the intentional and contextual matching scores. 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 going through all the states of the model, we can determine the state 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 can anticipate the user's future needs by offering her/him the most appropriate service that can interest her/him.

**Table 3. Algorithm representing the prediction process**

| Algorithm 1 : Context-aware Intentional Service Prediction |  |
|--|--|
| 1  | <b>Procedure ServicePrediction</b>   |
|  | <b>Input</b>   |
|  | $I_u$ : current user's intention   |
|  | $CX_u$ : current user's context  |
| 5  | $M_c$ : user's behavior model  |
|  | <b>Output</b>  |
|  | Result : tuple $\langle I_{next}, CX_{next}, Sv_{next} \rangle$ corresponding to the user's next situation |
|  | Ranked = $\emptyset$   |
|  | Result = $\emptyset$   |
| 10   | States = <i>getStates</i> ( $M_c$ )  |
|  | FOR EACH id $\in$ States   |
|  | DO   |
|  | $I_{id}$ = <i>getIntention</i> (id)  |
|  | $CX_{id}$ = <i>getContext</i> (id)   |
| 15   | Score = ( <i>IntentionalMatch</i> ( $I_{id}$ ) + <i>ContextMatch</i> ( $CX_{id}$ ) ) / 2                   |
|  | Ranked = Ranked $\cup$ { $\langle id, Score \rangle$ }   |
|  | END FOR  |
|  | IF Score NOT EMPTY   |
|  | THEN   |
| 20   | Observed = <i>MaxScore</i> (Ranked)  |
|  | Score <sub>Max</sub> = <i>getScore</i> (Observed)  |
|  | IF Score <sub>Max</sub> $>$ $\alpha$ THEN  |
|  | Success = <i>FindNextState</i> ( <i>getState</i> (Observed), $M_c$ )                                       |
|  | $I_{next}$ = <i>getIntention</i> (Success)   |
| 25   | $CX_{next}$ = <i>getContext</i> (Success)  |
|  | $Sv_{next}$ = <i>getService</i> (Success)  |
|  | Result = { $\langle I_{next}, CX_{next}, Sv_{next} \rangle$ }  |
|  | END IF   |
|  | END IF   |
| 30   | RETURN Result  |

## 6. Evaluation

The service discovery and prediction mechanisms presented in this paper were implemented using Java language, Jena2<sup>2</sup> and the reasoner Pellet<sup>3</sup>. In order to evaluate the proposed mechanisms, we use the test collection OWLS-TC<sup>4</sup>, a very popular set of OWL-S service descriptions available in different domains. We focus, for evaluation purposes, on the traditional travel domain, selecting the services descriptions concerning this domain. We extend these service descriptions proposed by OWLS-TC, using our API OWL-SIC (OWL-S Intentional & Contextual) (Najar et al. 2012a), which includes, on each service description, intentional and contextual information. Besides, we create a database containing user's traces, recognized clusters and the user's behavior model. We mixed a set of arbitrary traces with others following a scenario representing a well-defined user behavior. Thereafter, we launch the clustering algorithm on the set of traces in order to determine all the recognized clusters. Then, we execute the classification algorithm on all traces and clusters in order to update the user's behavior model stored in the database.

<sup>2</sup> <http://jena.sourceforge.net/>

<sup>3</sup> <http://www.mindswap.org/2003/pellet/index.shtml>

<sup>4</sup> <http://www.semwebcentral.org/projects/owl-s-tc/>

As part of our experiments, we deployed the proposed mechanism 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. Two criteria retained our attention: the algorithms scalability and the quality of their results. By the scalability, we intend to evaluate if the execution time is reasonable and if it scales up in a reasonable way. By evaluating the quality of the proposed, we intend to evaluate whether the proposed algorithms can effectively reach their goals. In order to perform such an evaluation, we formulate a set of user's requests relatives to the travel domain. These requests represent the user's intention and her/his current context. These requests are formalized according to three different distributions. The *first distribution* considers requests that are very similar to one or more available services (for the service discovery mechanism) or to a clusters centroid (for the service prediction mechanism). Then, the *second distribution* illustrates situations in which the elements describing the intention and/or the context are not described in the ontologies, while there are services or clusters that could be considered as 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.

Our first experiment concerns the evaluation of the proposed service discovery mechanism. We measure the performance of the discovery algorithm by varying the number of services between 100 and 400. As illustrated in Fig. 7, the execution time follows a polynomial trend of degree three varying from 2.79 s for 10 services to 11.82 s for 400 services. Indeed, this trend correspond to a complexity  $O(N^3)$ , where N represents the set of available services. We consider that this set will be greater than the set of contextual conditions that is supposed to be match by a service, and than the set of context observations available on the user's context. This means that, in its worst case, discovery algorithm executes respecting such a polynomial complexity. Such polynomial complexity explains the observed execution time. Even if this time seems still high, 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.

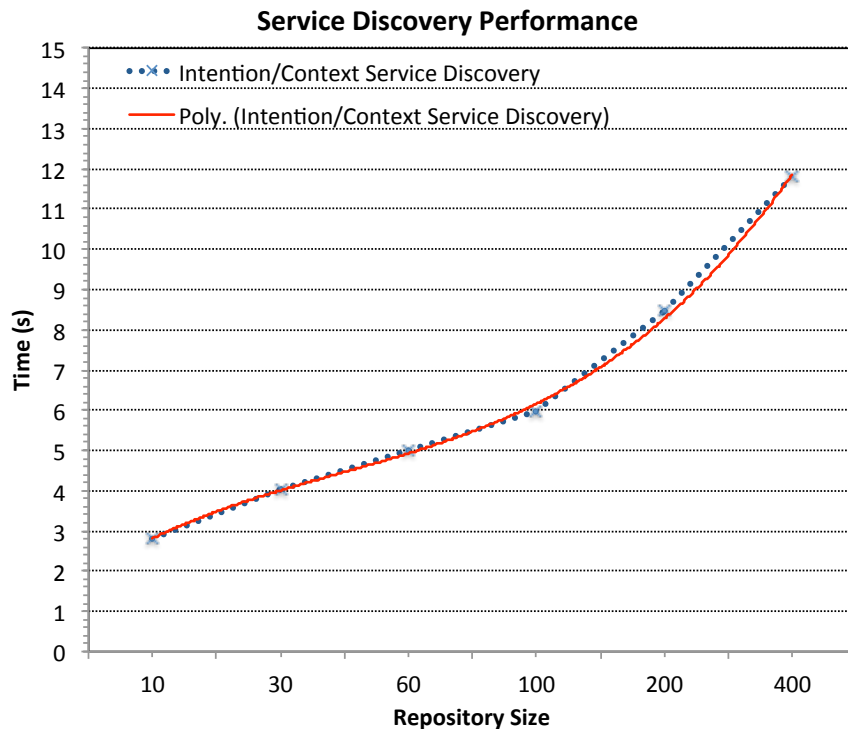


Fig. 7. Performance evaluation of the Service Discovery algorithm

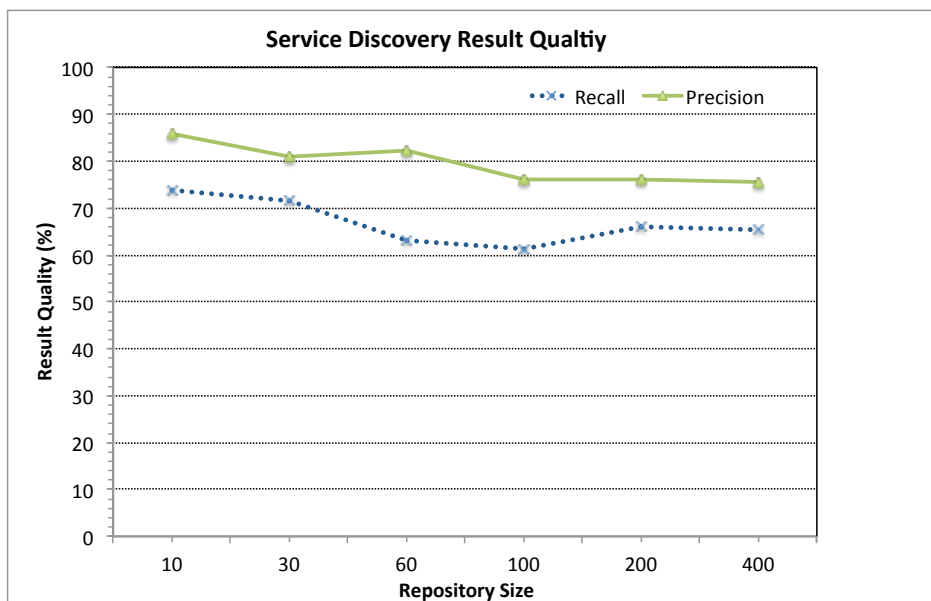
Besides, in order to measure the quality of the result, we cover the two most useful quality metrics: precision and recall (Xiao et al. 2010). As indicated in Table 4, precision represents the capacity of a system to return only relevant items, while recall indicates the capacity of a system to discover all

relevant items. Through the experiments, we observe that the precision and recall are interesting factors when considering the intention and the context in service discovery. The result presented in Fig. 8 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. 8. We can observe that the average recall approximates the 67%. These results can be explained by the different request distribution we prepared for these experiments. Indeed, in our second distribution, we have described some user's requests in which the elements of the intention are not described in ontologies, while it exists a set of services able to satisfy this intention in the current user's context. The evaluation of the requests belonging to this second distribution leads to the evaluation of situations that can potentially harm the results quality.

**Table 4. Definition of precision and recall metrics**

$$Precision = \frac{|\{relevant\ items\} \cap \{retrieved\ items\}|}{|\{retrieved\ items\}|}$$

$$Recall = \frac{|\{relevant\ items\} \cap \{retrieved\ items\}|}{|\{relevant\ items\}|}$$



**Fig. 8. Quality evaluation of the service discovery algorithm**

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, situations and states, by measuring the average processing time. Concerning the clustering algorithm, we measure the time this algorithm takes to determine the belonging cluster of a new observation. Results, presented in Fig. 9, demonstrate that the execution time observes a polynomial trend, varying from 2.56s for 7 clusters up to 5.1s for 186 clusters. In other words, by increasing about 6x the number of clusters, execution time has augmented only about 2x.

A similar trend could be observed for the classification algorithm (see Fig. 10), in which we evaluate the time the algorithm takes to update the user's behavior model. We vary the number of available observations from 10 to 200 observations, which represents a variation of 20x in the number of

observations, and we obtained an execution time varying from 39ms up to 398ms, representing a variation of only 10x in the execution time.

Finally, 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. As illustrated in Fig. 11, the execution time also follows a polynomial trend of degree three, like the service discovery algorithm, varying from 1.63s for 8 states up to 4.16s for 168 states. In other words, we increased the number of states over 25x, while the execution time has only increased by 2.5x. All these results allow us to validate the feasibility of our proposed algorithms.

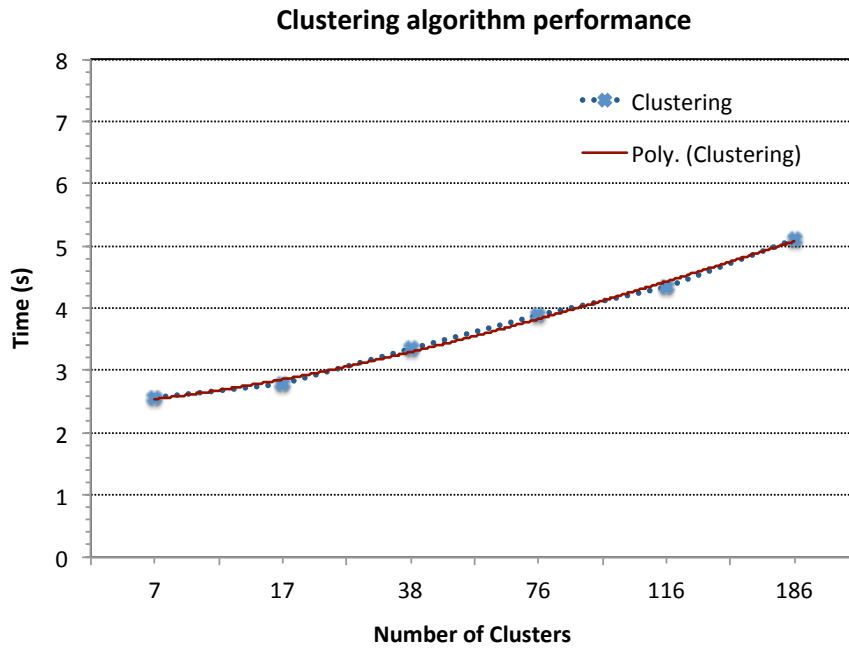


Fig. 9. Performance evaluation of the clustering algorithm

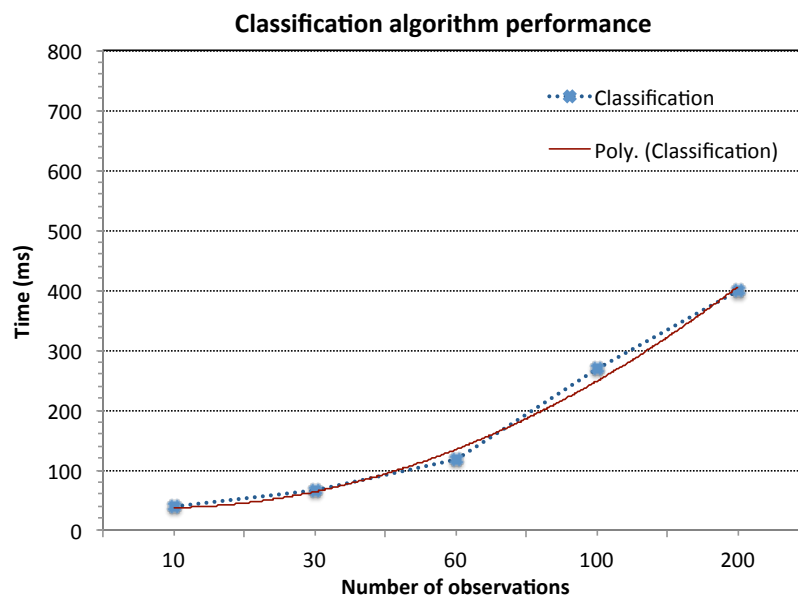
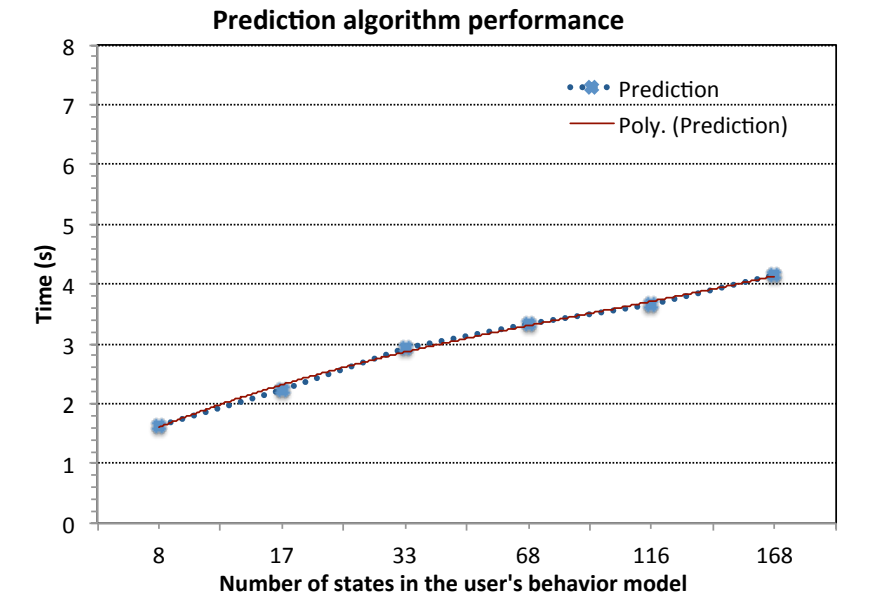


Fig. 10. Performance evaluation of the classification algorithm



**Fig. 11. Performance evaluation of the prediction algorithm**

In order to measure the result quality of the prediction process, we use a quality metric inspired from the precision used to evaluate the service discovery. This metric is used to check whether the predicted service is the one that is expected or not. We determined previously the service that the prediction algorithm should predict for a given user's request, based on the user's behavior model. Then, we compared this service with the service returned by the algorithm.

We illustrate in Fig. 12 the quality percentage achieved by the algorithm by varying the number of states. This percentage represents the average quality obtained for all the user's requests. The results presented in Fig. 12 indicate that the prediction algorithm has a good quality around 60%. Similar to the discovery mechanism, these results can be explained by the evaluation of certain situations described by our requests, which can significantly degrade the results quality obtained. For example, in some requests using intentions in which the verb and/or the target are fairly generic or specific, we obtain a poor quality that is, in some cases, below 45%. Besides, these results also demonstrate the impact of the threshold on the prediction mechanism. Indeed, when considering high threshold settings in the prediction algorithm, some clusters or states that could meet the immediate user's intention in her/his current context will not be selected, and this contributes to degradation of the results quality.

The analysis of these results shows the importance of the service discovery and prediction mechanisms. We believe that the proposed mechanisms allow the selection of the service that fulfills the user's immediate needs and the anticipation of her/his future need. This is thanks to both our intentional approach, which is more transparent to the user, and our contextual approach that restricts services to those that are valid. However, it is important to note that such good results cannot be obtained if the system designer does not establish, from the beginning, a rich description of the available services and of the different ontologies, as well as an appropriate threshold setting.

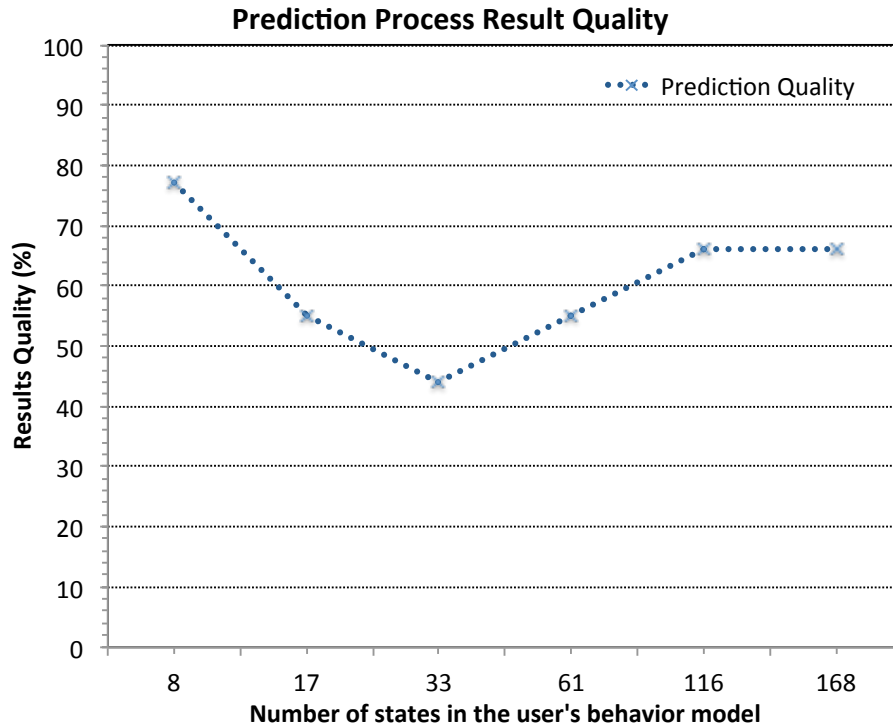


Fig. 12. Quality evaluation of the prediction process

## 7. Discussion and Future Directions

Evaluation results presented earlier in this paper (cf. section 6) demonstrate that, by observing the user's intention and combining it with the notion of context, it is possible to propose users better suited services, which are able to satisfy this intention considering current user's context conditions. Results considering the proposed prediction mechanism demonstrate that by analyzing user's traces, it is possible to achieve a proactive behavior, anticipating future user's needs. Besides these quality considerations, evaluation results also demonstrate the feasibility of the proposed mechanisms, which show good performance behaviors.

However, several other questions raise from these mechanisms and from future directions on PIS. The first of these questions consider the complexity of intentions related to the PIS services. The prediction mechanism we propose allows identifying only simple sequences of 1+1 services, representing the current intention and the following one, without considering whenever such sequences of intentions correspond to a bigger strategy. In fact, according to Kaabi and Souveyet (2007), high-level intentions can be decomposed on lower level ones. This means that, in order to satisfy a complex high-level intention, user may have to first satisfy several low level intentions, simplest and consequently easiest to operationalize. Currently, our prediction mechanism allows the dynamic composition of sequences of two intentions, but different other forms of composition are possible. Indeed, Kaabi and Souveyet (2007) have identified several different operators, including parallel intentions, i.e. intentions related by a AND construct, in which the satisfaction of a bigger intention demands the satisfaction of other smaller intentions simultaneously, and variability, in which multiple alternative intentions can be used in order to satisfy a high-level intention (like an OR construct). Today, such composition of intentions is only available at design time: services encapsulating such compositions should be predefined and described as such (see Najar et al. 2012a for details); no other dynamic composition of intentions is possible.

We advocate that, by analyzing user's traces, it will be possible to dynamically identify such compositions. We believe that these compositions of intentions can be seen as a user's behavioral

patterns describing complex strategies users apply in order to reach bigger goals. However, identifying such complex behavioral patterns is a harder task than identifying only simple sequences, which may demand more complex learning techniques, such as Bayesian networks for instance. Besides, dynamically identifying such behavior patterns may contribute to improve transparency of PIS, since these patterns allow composing fine-grained intentions (and services) on coarse-grained ones, easiest to understand by the users.

Dynamically composing coarse-grained services will become a priority for PIS in a near future thanks to the integration of IoT technologies on future PIS. With the development of new IoT technologies, PIS will have at their disposal in the future new multiple devices (sensors, actuators, etc.), offering multiple services and data. Indeed, the cost of bandwidth and processing power have declined as well as the cost of sensors, that become so cheap they can be used in many devices (Jones, 2014). According Broy and Schmidt (2014), *“within the next 25 years the world around us will consist mainly of input and output components to networked computers; most of the things and devices we interact with will be linked to a global computing infrastructure”*. The IoT vision considers a full new world of physical and virtual ‘things’ that are seamlessly integrated into the network and fully communicating. These ‘things’ are supposed to interact and to communicate among themselves and with the environment by exchanging data and information ‘sensed’ about the environment while reacting to physical world events (Sundmaeker et al. 2010). A full new panel of collected data and services will be then possible thanks to the integration of such ‘things’ into PIS, opening on those systems new perspectives of advanced features and services.

According to (Sundmaeker et al. 2010), things on the IoT are expected to become active, participating in business, information and social process. PIS will then be at the center of adoption of IoT in business companies. PIS will have to integrate these technologies, which will offer large volumes of data and an increasing number of services integrated to the environment. Nevertheless, the nature of the services offered by PIS will evolve and we may reasonably expect that their granularity will also change, varying from fine grained services offered by small actuators integrated into the physical environment, till coarse grained services, involving complex business process. More then never, a user-centric approach based on the user’s intention will be necessary, in order to make this complex environment understandable for the user. Discovery and prediction mechanisms we propose represent a first step in this direction. The evaluation results discussed earlier in this paper (cf. section 6) demonstrated not only the interest of this approach, but also interesting performance results, with a good scalability, making then possible to consider the application of the proposed mechanisms on such dynamic and dense environment.

Besides, the possibility of dynamically composing services thanks to observed user’s behavior patterns represents thus a necessary step towards transparent PIS on a IoT environment. Different from current composition mechanisms, which consider often only service input and output information in addition to context information, such new composing mechanisms should consider user’s intention associated with these services in addition to the context itself. It is then necessary to go beyond our current prediction mechanism, towards advanced learning algorithms, combining both intention and context information on more complex patterns.

Moreover, we believe that the development of IoT will bring an important swift on the way context awareness is conceived. Until now, context information is established on design time: system designers have to anticipate what context information they need (or want) to observe at design time and integrate on their systems design the corresponding sensing mechanisms. However, thanks to IoT technologies, PIS will have available in their environment several new context information that was not necessarily predefined or known at design time. Sensing context information is not the challenge anymore. The challenge will be how to opportunistically explore context information collected and discovered from the environment at execution time, instead of using only predefined context elements. The question of relevance of context information becomes central for service discovery and prediction, since the volume and the nature of available context information will exponentially grow. We strongly believe that it is necessary to go further on today’s clustering and classification techniques in order to efficiently identify context elements that have a real impact on service discovery and prediction, and more generally, on the overall use of PIS.

We advocate that new successful PIS will have to integrate competitive BI (Business Intelligence) techniques, adapted to the Big Data reality in order to deal with the explosion of context data brought by IoT. Indeed, according to Andriole and Bojanova (2014), to remain competitive, all companies will invest heavily in analytics and business intelligence (BI). However, volume of data affected by such analytics techniques will exponentially grow with IoT, particularly with we consider all context information that can be observed, requiring the use of Big Data techniques in order to store and handle huge volume of data. For Jones (2014), Big Data analytics makes IoT possible. This means that it is necessary to go beyond today data warehouses for data analytics. Since the nature of the context information granted by IoT is quite different from what we are currently analyzing with BI techniques today, other challenges are rising. Dynamic and heterogeneous nature of context information collected by smartphones, sensors and other IoT devices, represents an interesting challenge for data analytics techniques. These techniques are not necessarily prepared for such heterogeneity and volume of data, as demonstrated by Jaffal et al. (2014) and Ramakrishnan et al. (2014). New improvement on these techniques will then be necessary for fully explore context information as Big Data on such IoT environment.

To sum up, we advocate that future PIS will have to integrate techniques from multiple domains, such as Big Data and Data analytics, in order to be successful. And inversely, new PIS offer interesting challenges for these domains, by confronting Information System needs to a new environment offered by IoT and pervasive technologies.

## **8. Conclusions**

In this paper, we have proposed a user-centric approach for service discovery and prediction considering PIS. This approach is needed in order to hide the complexity of these systems and to achieve the transparency required by their users. Indeed, new technologies are transforming the way users interact with Information Systems (IS). New trends, such as BYOD and IoT transform IS environment on a pervasive environment, full of complex and heterogeneous technologies. New Pervasive Information Systems have to integrate this environment, without penalizing the user, who should focus on her/his own activities and not on the environment. A new approach for conceiving PIS is then necessary in order to reach context-awareness and transparency needed for these systems. We have introduced in this paper a user-centric approach that focus on both user's intention and execution context. We advocate that PIS transparency and proactivity can be enhanced through the proposed service discovery and prediction mechanisms, defined considering the user's point of view. These allow us not only offering user the most suitable services given her/his current intention and context, but also to anticipate the future user's needs in a fairly understandable way. By this approach, we believe contributing to the improvement of PIS transparency and proactivity through a user-centric perspective focusing on the intentions that services satisfy in a given context.

Moreover, evaluation results we have presented in this paper are promising and encourage us to go further on our approach. Proposed algorithms have demonstrated to scale up and have presented interesting quality results. These encourage us to improve current implementation of these algorithms, by considering different optimization techniques, such as parallelizing segments of code. Besides, an improved evaluation of the algorithms is also needed. Indeed, evaluating the user acceptance 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, mainly 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.

Besides, other improvements on the proposed algorithms might be considered. For instance, the influence of people on their peers, as demonstrated by Baras et al. (2014), could be considered on the discovery mechanism. Similarly, other algorithms, such as those proposed by Kas et al. (2014), could

be considered for calculating the centroid of a cluster on the prediction algorithms. Such improvements should be considered and evaluated in the future.

Finally, as discussed previously in this paper, PIS open new research perspectives for IS. The development of IoT and the potential integration of a large volume of data, and notably, context data, and services will durably impact PIS. Successful PIS in the future will be those capable of not only handle big volumes of data, but also to explore these data, with appropriate mining and data analytics techniques. Service discovery and prediction mechanism are only a few examples of the numerous mechanisms and services offered by PIS that can benefit from these new developments.

## References

- Adomavicius G, Tuzhilin A (2011). Context-aware recommender systems. In: Ricci F, Rokach L, Shapira B, Kantor PB (*Eds.*), *Recommender Systems Handbook*, Springer, 2011, pp. 217-253
- Ali F, Lee SW, Bien Z, Mokhtari M (2008) Combined Fuzzy State Q-learning Algorithm to Predict Context Aware User Activity under Uncertainty in Assistive Environment. Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD '08). Doi: 10.1109/SNPD.2008.13 . pp. 57-62
- Andriole SJ, Bojanova I (2014) Optimizing Operational and Strategic IT. *IEEE IT Professional* 16(5):12-15. September/October 2014.
- Baldauf M, Dustdar S, Rosenberg F (2007) A survey on context-aware systems. *Int. J. Ad Hoc Ubiquitous Comput.* 2(4): 263–277
- Baltrunas L, Ricci F (2013) Experimental evaluation of context-dependent collaborative filtering using item splitting. *User Modeling and User-Adapted Interaction: Special issue on Context-Aware Recommender Systems*, 24: 7-34
- Baras D, Ronen A, Yom-Tov E (2014) The effect of social affinity and predictive horizon on churn prediction using diffusion modeling. *Social Network Analysis and Mining*, Springer, 4(1):232. doi 10.1007/s13278-014-0232-2.
- Basiri A, Malek MR (2014) Providing relevant information in an ambient services using service requester's "logical area". *Journal of Ambient Intelligence and Humanized Computing*, Springer, 5(4) : 539-549.
- Broy M, Schmidt A (2014) Challenges in Engineering Cyber-Physical Systems. *Computer*, IEEE Computer Society, 47(2): 70-72
- Capilla R, Ortiz O, Hinchey M (2014) Context Variability for Context-Aware Systems. *Computer*, IEEE, 47(2): 85-87. doi: 10.1109/MC.2014.33
- Castro-Leon E (2014) Consumerization in the IT Service Ecosystem, *IEEE IT Professional* 16(5): 20-27. September/October 2014.
- Chaari T, Laforest F, Celentano A (2007) Adaptation in context-aware pervasive information systems: the SECAS project. *Journal of Pervasive Computing and Communications*. 3(4): 400-425.
- Chang JM, Ho P-C, Chang T-C (2014) Securing BYOD. *IEEE IT Professional* 16(5): 9-11. September/October 2014.
- Cremonesi P, Garza P, Quintarelli E, Turrin R (2011) Top-N recommendations on Unpopular Items with Contextual Knowledge. In: Adomavicius G, Baltrunas L, Hussein T, Ricci F, Tuzhilin A (*Eds.*), 3<sup>rd</sup> Workshop on Context-Aware Recommender Systems (CARS) in conjunction with the 5th ACM Conference on Recommender Systems (RecSys 2011), Chicago, USA. *CEUR Workshop Proceedings* vol 791. <http://ceur-ws.org/Vol-791/>. Accessed 3 November 2014.

- Cubo J, Pimentel E (2012) On the Service Discovery using Context-Awareness, Semantic Matching and Behavioural Compatibility. IEEE 15<sup>th</sup> International Conference on Computational Science and Engineering. doi 10.1109/ICCSE.2012.43. pp. 259-266.
- Dey A (2001) Understanding and using context. *Personal and Ubiquitous Computing* 5(1): 4-7.
- Earley S, Harmon R, Lee MR, Mithas S (2014) From BYOD to BYOA, Phishing, and Botnets. *IEEE IT Professional* 16(5): 16-18. September/October 2014.
- Eikerling H-J, Mazzoleni P, Plaza P, Yankelevich D, Wallet T. (2007) Services and mobility: the PLASTIC answer to the Beyond 3G challenge. White paper, Dec. 2007, [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). Accessed February 2014.
- Feller W (1968) *An Introduction to Probability Theory and its Applications*. New Jersey, Wiley. ISBN: 0-471-25708-7
- Fensel D, Facca FM, Simperl E, Toma I (2011) *Semantic Web Services*. Springer Berlin Heidelberg. ISBN: 978-3-642-19192-3.
- Foresti GL, Farinosi M, Vernier M (2015) Situational awareness in smart environments: socio-mobile and sensor data fusion for emergency response to disasters. *Journal of Ambient Intelligence and Humanized Computing*, Springer, 6(2): 239-257.
- Jaffal A, Kirsch-Pinheiro M, Le Grand B (2014) Unified and Conceptual Context Analysis in Ubiquitous Environments, In: Mauri JL, Steup C, Knoch S (Eds.), 8th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2014), August 24 - 28, 2014 – Rome, Italy. IARIA, ISBN: 978-1-61208-353-7, pp. 48-55.
- Javari A, Gharibshah J, Jalili M (2014) Recommender systems based on collaborative filtering and resource allocation. *Social Network Analysis and Mining*, Springer, 4(1): 234. doi 10.1007/s13278-014-0234-0.
- Jones M (2014) Internet Of Things: Shifting From Proprietary To Standard, ValueWalk, July 2014, <http://www.valuwalk.com/2014/07/internet-of-things-iot/>. Accessed October 2014.
- Kaabi RS, Souveyet C (2007) Capturing intentional services with business process maps. In: Rolland C, Pastor O, Cavarero J (Eds.) 1st IEEE International Conference on Research Challenges in Information Science (RCIS 2007), pp 309-318.
- Kas M, Carley KM, Carley LR (2014) An incremental algorithm for updating betweenness centrality and k-betweenness centrality and its performance on realistic dynamic social network data. *Social Network Analysis and Mining*, Springer, 4(1): 235. doi 10.1007/s13278-014-0235-z
- Kirsch-Pinheiro M, Vanrompay Y, Berbers Y (2008) Context-aware service selection using graph matching. In: Paoli FD, Toma I, Maurino A, Tilly M, Dobson G (Eds.), *Proceedings of the 2nd Workshop on Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC'08) at ECOWS 2008*. CEUR Workshop Proceedings vol 411. <http://ceur-ws.org/Vol-411/>. Accessed October 2014.
- König I, Voigtmann C, Klein B, David K (2011). 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 (Context 2011)*, Lecture Notes Computer Science, Springer Berlin Heidelberg. 6967: 159-172
- Kourouthanassis PE, Giaglis GM (2006) A Design Theory for Pervasive Information Systems. In: Mostéfaoui SK, Maamar Z, Giaglis GM (Eds.) 3rd International Workshop on Ubiquitous Computing (IWUC 2006), In conjunction with ICEIS 2006, INSTICC Press, pp. 62-70.
- Mayrhofer R (2004). *An Architecture for Context Prediction*. PhD thesis, Johannes Kepler University of Linz. <http://www.mayrhofer.eu.org/downloads/publications/PhD-ContextPrediction-2004.pdf>. Accessed 3 November 2014.

- Mayrhofer R (2005). Context prediction based on context histories: Expected benefits, issues and current state-of-the-art. In Prante T, Meyers B, Fitzpatrick G, and Harvel LD (*Eds.*), *Proceeding of the 1st International Workshop on Exploiting Context Histories in Smart Environments (ECHISE 2005)*, 3<sup>rd</sup> International Conference on Pervasive Computing (PERVASIVE 2005). <http://www.pervasive.ifi.lmu.de/workshops/w8/papers/echise2005-s17-ContextPredictionBasedOnContextHistories-Mayrhofer.pdf>. Accessed 3 November 2014.
- Mokhtar SB, Kaul A, Georgantas N, Issarny V (2006). Efficient Semantic Service Discovery in Pervasive Computing Environments. In: Steen V, Henning M (*Eds.*), 7<sup>th</sup> Int. Middleware Conference (Middleware'06), *Lecture Notes Computer Science*, Springer Berlin Heidelberg. 4290: 240-259.
- Mokhtar SB, Preuveneers D, Georgantas N, Issarny V, Berbers Y (2008). EASY: Efficient semAntic Service discoverY in pervasive computing environments with QoS and context support. *Journal of System and Software*, 81(5): 785–808.
- Najar S, Kirsch-Pinheiro M, Souveyet C (2012a). Enriched Semantic Service Description for Service Discovery: Bringing Context to Intentional Services. *International Journal on Advances in Intelligent Systems*, 5(1&2): 159-174
- Najar S, Kirsch Pinheiro M, Souveyet C, Steffanel A (2012b). Service Discovery Mechanism for an Intentional Pervasive Information System. In: Goble CA, Chen PP, Zhang J (*Eds.*), *IEEE 19<sup>th</sup> International Conference on Web Services ICWS*, Honolulu, United States, pp. 520-527
- Najar S, Kirsch-Pinheiro M, Souveyet C (2014). A user-centric vision of service-oriented Pervasive Information Systems. In: Bajec M, Collard M, Deneckère R (*Eds.*), 8th International Conference on Research Challenges in Information Science, *IEEE*, pp. 359-370
- Nazerfard E, Cook DJ (2015) CRAFFT: an activity prediction model based on Bayesian networks. *Journal of Ambient Intelligence and Humanized Computing*, Springer, 6(2): 193-205.
- Olsson T, Bjurling B, Chong M, Ohlman B (2011). Goal Refinement for Automated Service Discovery. 3<sup>rd</sup> International Conferences on Advanced Service Computing, pp. 46–51. [http://www.thinkmind.org/index.php?view=article&articleid=service\\_computation\\_2011\\_3\\_10\\_1013](http://www.thinkmind.org/index.php?view=article&articleid=service_computation_2011_3_10_1013). Accessed 3 November 2014.
- Paridel K, Mantadelis T, Yasar A, Preuveneers D, Janssens G, Vanrompay Y, Berbers Y (2014) Analyzing the efficiency of context-based grouping on collaboration in VANETs with large-scale simulation. *Journal of Ambient Intelligence and Humanized Computing*, Springer, 5(4) : 475-490.
- Paolucci M, Kawamura T, Payne T, Sycara K (2002). Semantic Matching of Web Services Capabilities. In Horrocks I, Hendler J (*Eds.*), *The Semantic Web (ISWC 2002) - Lecture Notes in Computer Science*, Springer Berlin Heidelberg. 2342: 333-347
- Ramakrishnan A, Preuveneers D, Berbers Y (2014). Enabling self-learning in dynamic and open IoT environments. In: Shakshuki, E. & Yasar, A. (*Eds.*), *The International Conference on Ambient Systems, Networks and Technologies (ANT-2014)*, the 4th International Conference on Sustainable Energy Information Technology (SEIT-2014), Hasselt, Belgium, June 2 - 5, 2014, *Procedia Computer Science*, Elsevier, 32: 207-214
- Ramakrishnan A, Preuveneers D, Berbers Y (2013). A Modular and Distributed Bayesian Framework for Activity Recognition in Dynamic Smart Environments. In: Augusto JC, Wichert R, Collier R, Keyson D, Salah AA, Tan AH (*Eds.*), *Ambient Intelligence*, 4th International Joint Conference, *Aml 2013*, *Lecture Notes in Computer Science*, Springer Berlin Heidelberg. 8309: 293-298
- Santos LOB da Silva, da Silva EG, Pires LF, van Sinderen M (2009). Towards a Goal-Based Service Framework for Dynamic Service Discovery and Composition. 3rd. Int. Conf. on Information Technology: New Generations, *IEEE Computer Society*, doi: 10.1109/ITNG.2009.27. pp. 302-307
- Sigg S, Haseloff S, David K (2010). An Alignment Approach for Context Prediction Tasks in UbiComp Environments, *IEEE Pervasive Computing*, 9(4): 90-97

Sundmaeker H, Guillemin P, Friess P, Woelfflé S (2010). Vision and Challenges for Realising the Internet of Things. Cluster of European Research projects on the Internet of Things (CERP-IoT). doi 10.2759/26127.

[http://www.theinternetofthings.eu/sites/default/files/Rob%20van%20Kranenburg/Clusterbook%202009\\_0.pdf](http://www.theinternetofthings.eu/sites/default/files/Rob%20van%20Kranenburg/Clusterbook%202009_0.pdf). Accessed 3 November 2014

Suraci V, Mignanti S, Aiuto A (2007). Context-aware Semantic Service Discovery. 16<sup>th</sup> IST Mobile and Wireless Communications Summit, pp. 1-5

Toninelli A, Corradi A, Montanari R (2008) Semantic-based discovery to support mobile context-aware service access. *Computer Communications*. 31(5): 935-949.

Vanrompay Y, Kirsch-Pinheiro M, Berbers Y (2011). Service Selection with Uncertain Context Information, In: Reiff-Marganiec S, Tilly M (Eds.), *Handbook of Research on Service-Oriented Systems and Non-Functional Properties: Future Directions*, IGI Global, pp. 192-215.

Vanrompay Y, Berbers Y (2012). A Methodological Approach to Quality of Future Context for Proactive Smart Systems. In: Andreev, S.; Balandin, S. & Koucheryavy, Y. (Eds.), *Internet of Things, Smart Spaces and Next Generation Networking*, Lecture Notes Computer Science, Springer Berlin Heidelberg 7469: 152-163

Xiao H, Zou Y, Ng J, Nigul L (2010). An Approach for Context-Aware Service Discovery and Recommendation, *IEEE International Conference on Web Services (ICWS 2010)*, doi: 10.1109/ICWS.2010.95, pp. 163-170.