



HAL
open science

Intention Prediction Mechanism In An Intentional Pervasive Information System

Salma Najjar, Manuele Kirsch Pinheiro, Yves Vanrompay, Luiz Angelo Steffenel, Carine Souveyet

► **To cite this version:**

Salma Najjar, Manuele Kirsch Pinheiro, Yves Vanrompay, Luiz Angelo Steffenel, Carine Souveyet. Intention Prediction Mechanism In An Intentional Pervasive Information System. Intelligent Technologies and Techniques for Pervasive Computing, IGI Global, pp.251-275, 2013. hal-00831733

HAL Id: hal-00831733

<https://hal-centralesupelec.archives-ouvertes.fr/hal-00831733>

Submitted on 7 Jun 2013

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.

Intention Prediction Mechanism In An Intentional Pervasive Information System

Salma Najar

Centre de Recherche en Informatique / Université Paris1 – Panthéon Sorbonne - France

Manuele Kirsch Pinheiro

Centre de Recherche en Informatique / Université Paris1 – Panthéon Sorbonne - France

Yves Vanrompay

MAS Laboratory / Ecole Centrale Paris - France

Luiz Angelo Steffens

Centre de Recherche en STIC (CReSTIC) / Université de Reims Champagne-Ardenne – France

Carine Souveyet

Centre de Recherche en Informatique / Université Paris1 – Panthéon Sorbonne - France

ABSTRACT

Nowadays, the development of pervasive technologies has allowed the improvement of services availability. These services, offered by information systems (IS), are becoming more pervasive, i.e., accessed anytime, anywhere. However, those pervasive information systems (PIS) remain too complex for the user, who just wants a service satisfying his needs. This complexity requires considerable efforts from the user in order to select the most appropriate service. Thus, an important challenge in PIS is to reduce user's understanding effort. In this chapter, we propose to enhance PIS transparency and productivity through a user-centred vision based on an intentional approach. We propose an intention prediction approach. This approach allows anticipating user's future requirements, offering the most suitable service in a transparent and discrete way. This intention prediction approach is guided by the user's context. It is based on the analysis of the user's previous situations in order to learn user's behaviour in a dynamic environment.

INTRODUCTION

Nowadays, the development of mobile and pervasive technologies has allowed a significant increase of services offered to users by information system (IS). Instead of having Information Technology (IT) in the foreground, triggered and manipulated by users, IT is gradually residing in the background, monitoring user's activities, processing this information and intervening when required (Kourouthanassis & Giaglis, 2006). In other terms, we are observing the emergence of a Pervasive

Information System (PIS) that intends to increase user's productivity by making IS available anytime and anywhere. Indeed, PIS arise from the ambition to provide pervasive access to IS, while adapting itself to user's context. The notion of context is employed in order to make these systems more intelligent and adaptive. It corresponds to any entity considered as relevant to the interaction between the user and the application (Dey, 2001).

Contrarily to traditional IS, whose interaction paradigm is the desktop, PIS deals with a multitude of heterogeneous devices, providing the interaction between the user and the physical environment (Kouruthanassis&Giaglis, 2006). As pointed out by Kouruthanassis&Giaglis (2006), the main characteristics of PIS are not only the heterogeneity of devices, but also the property of context-awareness. Therefore, the evolution of IS into PIS leads us to consider PIS as more than a simple set of logical services.

Weiser (1991) suggests that a pervasive environment will be characterized by its transparency and homogeneity. Twenty years later, we can notice that this pervasive environment, which was meant to be an invisible or unobtrusive one, represents a technology-saturated environment. This environment combines several devices highly present and visible. PIS has to deal with such an environment, in which a rapidly evolving and increasing number of services is available, with multiple implementations. In spite of this rapid evolution, PIS remains too complex for the user, who just wants a service that satisfies his needs. This complexity requires considerable effort from the user in order to understand what is happening around him and in order to select the service that best fulfils his needs.

Nowadays, pervasive environments represent reactive systems based only on current user's context. The proactive and anticipatory behaviour of PIS, notably by predicting the user's future situation, is hardly developed. Thus, most research on this topic remains on a technical level, discovering next context information or suitable service implementations. They do not consider the intentional requirements behind the user's experience. As a consequence, the user is often provided with several possibilities, even if he is not always able to understand what is proposed to him. We believe that, in order to achieve transparency advocated by Weiser (1991), PIS must reduce the user's understanding effort. PIS must hide the complexity of such multiple implementations and context situations. This will only be possible through a user-centred vision. This vision is based on the prediction of user's future requirements in a given context. It ensures a transparent access to a "space of services". This space of services hides technical details concerning how to perform these services.

In this chapter, we propose a new vision of PIS based on a space of services and on an intentional prediction approach. Our purpose is to predict the user's future intention based on his context, in order to offer the most suitable service that can interest him in a transparent and discrete way. This approach considers PIS through the notion of intention. The notion of intention can be seen as the goal that a user wants to achieve without saying how to perform it (Kaabi&Souveyet, 2007). It is described also as a goal to be achieved by performing a process presented as a sequence of goals and strategies to the target goal (Bonino et al., 2009). In other words, an intention is a requirement that a user wants to be satisfied without really caring about how to perform it or what service allows him to do so. This intentional vision allows us to focus on the *why* of the service instead of the *how*. By adopting this vision, we propose to improve the transparency by considering, on the one hand, the intention a service allows a user to satisfy, and on the other hand, the context on which this intention emerges. Based on this information, we propose an intention prediction approach that tries to anticipate user's future intention on a given context. The main purpose of such approach is to provide to the user a service that can fulfil his needs in a fairly understandable and non-intrusive way, reducing user's understanding effort.

To better illustrate this approach, we present in this chapter our middleware, called IPSOM (Intentional Pervasive Service Oriented Middleware). The purpose of IPSOM is to satisfy the user's intention by discovering, predicting and selecting for him the most suitable service in a given context. IPSOM integrates an intentional prediction mechanism guided by the context. This prediction mechanism is based on the assumption that, even in a dynamic and frequently changing pervasive information system, *common situations can be found*. Based on this assumption, this prediction mechanism considers a set of time series representing observed user's situations. A situation represents a user's *intention* in a given

context satisfied by a specific *service*. Thus, we are able to track and store these situations in a database, after each successful discovery process (history). By analysing the user's history, represented by these triplets $\langle \textit{Intention}, \textit{Context}, \textit{Service} \rangle$, a prediction mechanism can learn user's behaviour in a dynamic environment, and therefore deduce his immediate future intention.

This chapter is organized as follow: The next section presents an overview of related work. Then, we introduce our IPSOM, after which we detail after our proposed intention prediction mechanism. Next we present a discussion and future work. Finally, we conclude this chapter in the last section.

RELATED WORK

Nowadays, pervasive environments are merely reactive. Decisions are taken solely based on the current context. Indeed, research in the anticipatory and proactive behaviour of PIS, notably by predicting the user's future situation, is hardly done. By avoiding focusing on the prediction of future user's situation, current systems lack an important element in the search for transparency and homogeneity.

In order to help end users obtain their desired services, some research on Ubiquitous Computing (Abbar et al., 2009; Adomavicius et al., 2005; Boytsov&Zaslavsky, 2011; Sigg et al., 2010; Vanrompay, 2011) proposes mechanisms to automatically predict or recommend services using user's context. These researches focus especially on context prediction and on context based recommendation systems.

Concerning the first aspect, we have major contributions towards generic context prediction, such as Mayrhofer, Harald et al. (2003), Sigg et al. (2010) and Petzold et al. (2005). According to Vanrompay (2011), Mayrhofer et al. (2003, 2004) propose an architecture and a framework for high-level context prediction. It is based on an unsupervised classification, which tries to find previously unknown classes from input data. This architecture is based on five steps: sensor data acquisition, feature extraction, classification, labelling and prediction.

Similar to Mayrhofer et al. (2003, 2004), Sigget al. (2008, 2010) provide a formal definition of the context prediction task. They propose a context prediction architecture based on an alignment method, from which missing low-level context information is deduced. This alignment method is based on typical pattern and on alignment technique. It allows predicting the continuation of the typical sub-sequence the most similar to the suffix of the observed sequence.

Petzold et al. (2003, 2005) present an approach restricted to the prediction of primary context information (time, location, activity), which is less generic. In 2005, Petzold et al. propose to predict the user's next location in a 'smart office' based on previously visited locations.

Moreover, particular applications of context prediction have been developed by Hong et al. (2009), Lee & Cho (2010) and Meiners et al. (2010). First, Hong et al. (2009) propose a framework to automatically personalize services. They extract the relationships between user profiles and services under the same contextual situation. For this they analyse the user's context history. Meiners et al. (2010) present a generic and structured approach to context prediction based on two key principles. Firstly, developers can incorporate the knowledge of the application domain at design time. Secondly, multiple exchangeable prediction techniques, which are appropriate for the domain, can be selected and combined by the application developers. However, this work does not allow for the selection of appropriate prediction algorithms at runtime.

More recently, in 2011, challenges related to context prediction and applications of the prediction of context information have been identified by Boytsov&Zaslavsky. These authors propose an architecture based on reinforcement learning. They mention that an automated decision-making is a major challenge concerning context prediction. This should be based on the quality of the predicted context (Boytsov&Zaslavsky, 2011). In 2010, Boytsov&Zaslavsky extend the context spaces theory to enable context prediction and proactive adaptation (Boytsov et al., 2009). In context spaces theory, any kind of data that is used to reason about context is called a context attribute. A context attribute corresponds to a domain of values of interest. Context state refers to the set of all relevant context attributes at a certain time. A set of all possible context states constitutes application space. Therefore, application space can be

viewed as a multi-dimensional space where the number of dimensions is equal to the number of context attributes in the context state. The state of the system is represented by a point in the application space and the behaviour of the system is represented by a trajectory moving through the application space over time. Situation space represents a real life situation and it is defined as a subspace of the application space. Prediction of context information in this work occurs only on a high-level, i.e., situation space, using various machine-learning algorithms like Markov chains and Bayesian networks. A number of well-known machine learning approaches are evaluated on their appropriateness for context spaces. To decide on the execution of a specific adaptation given a prediction, authors propose a reinforcement learning approach.

We must also cite the contribution of authors on recommendation systems, which aim to propose services based on the user's context. For example, Adomavicius et al. (2005) propose the integration of contextual information into recommendation processes in order to improve recommender capabilities. In 2005, Adomavicius & Tuzhila propose a multidimensional approach. This approach is based on ratings that are sensitive to contextual information such as time, place and accompanying people.

In 2008, Yang et al. design an event-driven rule based system. This system recommends services according to user's context changes. Abbar et al. (2009) provide a similar approach, in which services are recommended based on user's log files and current context. Nevertheless, in order to select and recommend services, those approaches require historical data, which are not always available.

Without relying on log files, Xiao et al. (2010) propose an approach to dynamically derive a context model from ontologies and recommend services using context. They automatically extend the semantics of the context value using public available ontologies. Then, they use this semantics to recommend services.

All these works try to anticipate user's needs in order to offer him more transparency. Context prediction approaches (Boytsov & Zaslavsky, 2011; Meiners et al. 2010; Sigg, 2010; Vanrompay, 2011) try to predict user's next context based on the user's current context and history. However, none of these works consider the services a user invokes on a given context. Hence, most recommendation systems (Abbar et al., 2009; Adomavicius et al., 2005) propose a next service to users based solely on their context information, without considering the user's requirements behind a service, i.e., its goals. They propose an implementation to the user, ignoring why this service is needed.

Today, an important challenge in the field of PIS is to position itself at the user level. Current research remains on the technical level, discovering next context information or suitable service implementations, without considering the intentional requirements behind the user's experience. As a consequence, several possibilities are offered to the user, who is not always able to understand what is proposed to him.

INTENTIONAL PERVASIVE SERVICE ORIENTED MIDDLEWARE: IPSOM

Overview

In this chapter, we present a new vision of Pervasive Information System (PIS). It is based on a 'space of services', representing a user-centred approach. This approach emerges from our ambition to achieve more transparency for PIS, while addressing the limitations of existing approaches. Indeed, those existing approaches focus primarily on context adaptation, including the location and devices, neglecting therefore the intentional needs of the user.

Today, it is clear that the evolution of IS into PIS brings much more than a simple set of logical services into the IS. With the development of pervasive technologies, IT becomes embedded in the physical environment. It offers innovative services to the users evolving in this environment. However, contrarily to traditional IS, PIS may offer both logical and physical services. For this reason we introduce the notion of *space of services*. This notion is represented as a way to develop such user-centred view through a space not only including logical services, representing traditional information systems themselves, but also physical services embedded on the physical world. We consider that, in a PIS, a user evolves in a space of services. This space offers him a set of heterogeneous services whose focus is to accommodate user's needs. It allows representing knowledge about users and their environment, in order to discover and predict the most appropriate service that satisfies a user's intention in a given context.

This new approach aims to deal with the user's overload. This overload comes from the many possible implementations for each service, on one side, and from the effort required to understand them, on the other side. This new approach is able to hide the complexity of the pervasive environment through an intentional approach guided by the user's context of use. We advocate that understanding user's intention can lead to a better understanding of the real use of a service. Consequently, it leads to the selection of the most appropriate service that satisfies user's needs. To meet these requirements, contextual information plays a central role since it influences the selection of the best strategies of the intention satisfaction. The key elements of our approach are highlighted in Figure 1. The different modules constitute our *intentional and pervasive service oriented middleware* (IPSOM). IPSOM is a platform for service discovery and prediction based on the intention and context of the user.

Figure 1: Intentional Pervasive Information System Approach

The first element of IPSOM is the *context manager* (CM). The purpose of CM is to hide the context management complexity by providing a uniform way to access context information. First, the CM receives raw context data from different physical and logical sensors (GPS, RFID...). Then, it interprets such data in order to derive context knowledge represented on a higher level. Finally, this knowledge is stored in a knowledge database using a context model.

Next, the *intentional query processor* (IQP) is in charge of processing user's request. Such request represents user's intention, expressed by a verb, a target and a set of optional parameters. This intention is represented according to a specific template (Kaabi&Souveyet, 2007) (Rolland et al., 2010). The IQP enriches this request with context obtained from CM. This enriched request, represented in XML format, is then transferred to the discovery module.

Then, the service discovery module (DM) allows the satisfaction of the immediate user's intention. It is in charge of discovering and selecting the most appropriate service that fulfils his immediate intention in a given context. The DM mechanism is based on a semantic service description and a matching algorithm, which is detailed in next sections.

Next, the *learning module* (LM) is responsible for dynamically determining the user's behaviour model (classification). This module is based on the recognized clusters representing similar user's situations (clustering). The user's behaviour model, learned and maintained by the LM, will then be used by the prediction module.

Finally, the *prediction module* (PM), guided by the user's intention and context, is based on the results from the discovery process previously stored in a history database. From this data, the PM is able to anticipate the user's future needs. Then, it is able to propose him, in a proactive manner, a service that may be of interest. Thus, when the DM selects a service, the triplet $\langle \textit{intention}, \textit{context}, \textit{service} \rangle$ is sent to the prediction module. From the user's behaviour model proposed by the LM, the PM will determine the future user's intention and select the service that can meet its future needs. Thus, the PM is responsible for selecting, from the user's behaviour model, the situation that best represents the current user's situation. As a result, a service selected by the DM or by the PM will be presented in the form of an URI and sent to the *service invoker module*, which is in charge of invoking and executing it.

In next sections, we present our proposed extension of OWL-S service description in details. This extension takes into account the notion of intention and context. We also present an overview of our service discovery process associated to IPSOM.

Semantic Service Description

When a user requests a service, he chooses the intention that the service is supposed to satisfy. To be more exact, this intention emerges in a given context, which can also be used to characterize the service. Thus, from this assumption, we propose to enrich the OWL-S service description in order to include information about the context and the intention that characterizes a service (Najar et al., 2011). The information related to the intention is described through the addition of a sub-ontology. This sub-ontology represents the intention that a service is supposed to satisfy. Expert communities sharing a common

vision of their respective fields establish the ontologies defining the intention, like community-supported ontologies proposed by Mirbel&Crescenzo (2010). This vision fits perfectly with the PIS, since the services offered in these systems are tailored to a specific user community. Moreover, information related to context is described by a URL referring to an external resource. It allows the service provider to easily update the context information related to the service description. With this extension of OWL-S, we can describe the intention that a service is deemed to satisfy and the context conditions under which this service is valid and can be executed. This semantic service description is briefly described in next section. More detailed explanation of this extension can be found in Najjar et al. (2011).

Intentional description

According to an intentional perspective, a user requires a service because he has an intention that the service (Sv) is supposed to satisfy. Hence, the importance of considering user's intentions emerges on service orientation. This new dimension is central to the definition of a service.

The term *intention* has several different meanings. According to Jackson (1995), an intention is an "optative" statement expressing a state that is expected to be reached or maintained. The notion of intention can be seen as *the goal that we want to achieve without saying how to perform it* (Kaabi&Souveyet, 2007). Bonino et al. (2009) define an intention as *the goal to be achieved by performing a process presented as a sequence of intentions and strategies to the target intention*. Moreover, Ramadour&Fakhri (2011) characterize an intention as *the formulation of needs as a service in order to satisfy a composition*.

Even if they differ, all these definitions let us consider an *intention* as *a user's requirement representing the goal that a user wants to be satisfied by a service without saying how to perform it*. It represents a requirement formulated by the user, who knows exactly what he expects from the service, but who has no ability to indicate how to perform it.

The intention can be formulated according to a specific template. This template is based on a linguistic approach (Prat, 1997), representing user and service's requirements. This approach is inspired by the Fillmore's case grammar (Fillmore, 1968) and its extensions by Dick (Dick, 1989). According to this, an intention is formalized as follows:

Intention: [verb] [target] ([parameter])*

In this template, an intention (I) is composed by two mandatory elements: verb (V) and target (T). The *verb* exposes the action allowing the realization of the intention. Possible verbs can be organized in a verb ontology that recognizes significant verbs for a given community. Then, the *target* represents either the *object* existing before the satisfaction of the intention or the *result* created by the action allowing the realization of the verb. Finally, a parameter represents additional information needed by the verb.

We propose to enrich OWL-S service description with the intention associated to it. This is done, as illustrated in Figure 2, by adding a new sub-ontology that describes the intentional information of the service.

Figure 2: Service Intention in OWL-S (based on Martin et al. (2004))

This sub-ontology first adds a property to the Service that we called 'satisfies'. The range of this property is the added class 'Service Intention'. Thus, each instance of Service will satisfy a Service Intention description. The Service Intention provides the information needed to discover the appropriate service that satisfies a specific intention. Besides, the service intention presents "what the service satisfies", in a way that is suitable to determine whether the service fulfils the user's intention.

Figure 3: Example of Enriched Service Description in OWL-S

This part of the service description, as illustrated in Figure 3, presents the main intention of the service. This intention is formulated, as we described above, according to a specific template (Rolland et al., 2010), in which an intention is represented by a verb, a target and a set of parameters, as described above.

Contextual description

The notion of context (**C**) represents a key characteristic of any pervasive information system. It corresponds to a very wide notion. Most of the definitions agree that context has something to do with interactions between the user and the information system. The widely acknowledged definition describes the context as any information that can be used to characterize the situation of an entity (a person, place, or object considered as relevant to the interaction between a user and an application) (Dey, 2001). Also, Truong & Dustdar (2009) consider context elements as any additional information used in order to improve the service's behaviour in a specific situation. This contextual information allows service to operate better or more appropriately (Truong & Dustdar, 2010).

Furthermore, the notion of context is central to context-aware services that use it for adaptation purposes. Context information can stand for a plethora of information, from user's location, device resources (Reichle et al., 2008), up to user's agenda and other high level information (Kirsch-Pinheiro et al., 2004). Nevertheless, in order to perform such adaptation processes, context should be modelled appropriately. The way context information is used depends on what it is observed and how it is represented. Besides, the context adaptation capabilities depend on the context model (Najar et al., 2009).

Figure 4: Context Model

Different kinds of formalism for context representation have been proposed. Nevertheless, an important tendency can be observed in most recent works: the use of ontologies for context modelling (Najar et al., 2009). According to Najar et al. (2009), different reasons motivate the use of ontologies, such as their capability of enabling knowledge sharing in a non-ambiguous manner and their reasoning possibilities.

A formal context representation was proposed by Reichle et al. (2008). They represent context information based on three main concepts: 1) the *entity* specifying the element to which the context information refers; 2) the *scope* identifying the exact attribute of the selected entity that it characterizes; and 3) the *representation* used to specify the internal representation used to encode context information in data-structures. According to this context model, we directly associate the scope that we observe with the entity that the context element refers to. This let us consider that, in order to have the value for a given scope, we have to observe its corresponding entity. However, this raises an ambiguity since some scopes are not directly related to a precise entity. Therefore, in order to make this context model more meaningful, we believe that we must clearly separate the notion of entity that we want to represent from the property that we want to observe.

Based on Reichle et al. (2008), we define our context model illustrated in Figure 4. In this context model, context information is identified by two important concepts, the *entity* and the *attribute*. The distinction between these two concepts is adopted in order not to mix up the entity to which the context information refers to (e.g. user, device, etc.) with the attribute that characterize the property that we want to observe. The attribute represents a piece of context information about the environment (location, time...), a user (profile, role...) or a computational entity (resource, network...).

Our context model is based on a multi-level ontology, illustrated in Figure 5, representing knowledge and describing context information. It consists of an upper level, defining general context information (e.g. profile, activity, location, network, etc.), and a lower level, with more specific context information (temperature, latency, etc.). Besides, it provides flexible extensibility to add specific concepts in different domains. All these domains share common concepts that can be represented using a general context model, but they differ in some specific details.

Figure 5: Multi-Level Context Ontology

The importance of context information can differ from a user to another according to their preferences. Consequently, we propose a *profile context model*, as illustrated in Figure 6. According to this model, we assign to each context *entity* a *profile*. The *profile* allocates a *weight* to each context *attribute*. The weight reflects the importance of the context attribute. It is represented using the scale indicated below:

- Nil: {0.0}
- Poor: {0.1, 0.2, 0.3}
- Medium: {0.4, 0.5, 0.6}
- Good: {0.7, 0.8, 0.9}
- Excellent: {1.0}

Accordingly, the importance of the context attribute is proportional to its weight. When the weight decreases, the importance of the attribute decreases. When the weight increases, the importance of the attribute grows.

Figure 6: Profile Context Model

Then, we assume that a service is valid in a given context and needs to satisfy a set of context conditions in order to be executed. According to this, we propose to extend the service profile. This extension allows the service provider to define context information that characterizes an intentional service. Contextual information can then be considered as part of the service description, since it indicates context conditions to which the service is better suited. However, according to Kirsch-Pinheiro et al. (2008), context information cannot be statically stored in the service profile due to its dynamic nature. Context properties related to service execution can evolve (e.g., server load may affect properties of services running on it), whereas a service profile is supposed to be a static description of the service.

Thus, in order to handle dynamic context information on a static service description, we enrich the OWL-S service profile with a context attribute (Kirsch-Pinheiro et al., 2008). This context attribute represents a URL pointing to a context description file (see Figure 3). Since context information is dynamic and cannot be statically stored on the service profile description, we opt to describe context elements in an external file to allow the service provider to easily update such context information related to the service description itself. The context description of a service describes, on the one side, the situation status of the requested service (environment in which the service is executed), and on the other side, the contextual conditions (requirements) to execute the service. Both information elements can be used for service discovery purpose that is described in the next section.

Service Discovery

The Intentional Pervasive Service Oriented Middleware (IPSOM), which has been presented above, integrates a Service Discovery module. This module is based on a service discovery mechanism guided by user's intention and context. This intentional and contextual mechanism is proposed in order to hide implementation complexity, and consequently to achieve the transparency promised by pervasive environments. The intention concept is used to expose services and to implement a user-centred vision of PIS in a given context. Besides, contextual information plays a central role since it influences the selection of the best strategies of the intention satisfaction.

The service discovery, based on these two concepts (context and intention), will help users by discovering the most appropriate service for them, i.e., the service that satisfies the immediate user's intention in a given context. This service discovery is based on a semantic service description, as presented above, and on a semantic service discovery algorithm. This algorithm performs a semantic matching process in order to select the most appropriate service to the user. The goal of this matching algorithm is to rank the available services based on their contextual and intentional information. Then, it selects the most suitable one for the user. This algorithm semantically compares the user's intention with the intention that the service satisfies and user's current context with the service's context conditions. Then the service

having the highest matching score is selected. It represents the most appropriate service that satisfies user's immediate intention in his current context.

Figure 7: Service Discovery Process

More specifically, the semantic matching algorithm, as illustrated in Figure 7, is a two-step process (Najar et al. 2012): *intention matching* and *context matching*. In the first step, the intention matching is based on the use of ontologies, semantic matching and degree of similarity. Concerning the intention formulation, the intention matching is especially based on a verb and target matching. For the verb matching, we use an ontology of verbs. This ontology of verbs contains a domain-specific set of verbs, their different meanings and relations. The degree of similarity is then based on a semantic matching performed using this ontology. It reflects the existence of a semantic link between two verbs in the verb ontology, i.e., the relation between them. We define 5 levels of similarity:

- **Exact** to which we attribute the score 1
- **Synonym** to which we attribute the score 0,9
- **Hyponym**, i.e., the required verb is more specific than the provided verb, to which we attribute the score 0,7
- **Hypernym**, i.e., the required verb is more general than the provided verb, to which we attribute the score 0,5
- **Fail** to which we attribute the score 0

Similarly, for the target matching, we use a domain-specific ontology. This ontology represents the possible targets in a specific domain, from required targets T_U and provided target T_S . The degree of similarity is based on a semantic similarity calculated using the target ontology. This similarity represents a distance calculated based on the semantic link between two targets in the ontology. This semantic similarity is based on the algorithm proposed by Paoluci et al. (2002), using the following 4 levels:

- **Exact**: the required target is equivalent to the provided target
- **Plug-In**: the provided target subsumes the required target
- **Subsume**: the required target subsumes the provided target
- **Fail**: there is no subsumption between the two targets.

Thus, the intention matching between user's intention $I_U = \langle V_U, T_U \rangle$ and service's intention $I_S = \langle V_S, T_S \rangle$ is calculated based on the *target matching* and on the *verb matching* (Najar et al., 2012).

The second step, i.e. the context matching, is based on a context ontology, semantic similarity and a set of similarity measures. It matches individually the different context elements constituting the user (C_U) and service context descriptions (C_S). The context description for a user (C_U) or a service (C_S) represents a set of observable context elements, in which $C_U = \{c_j\}_{j>0}$ and $C_S = \{c_i\}_{i>0}$. Each context element is described by an entity (to which the context element refers), an attribute (that characterizes the property that we observe) and a set of observed values. Thus, the context matching is based on *entity matching*, *attribute matching* and *value matching*. In order to optimize our matching algorithm, we set a threshold below which the context matching process is stopped. This threshold can be customised according to the IS. It is based on the existence of a semantic link between the concepts in the ontology. By default, to proceed with the matching process, it is required that the distance between the concepts c_i and c_j in the ontology does not exceed two links. Accordingly, the threshold is calculated as a matching score: $1/(L+1) = 1/(2+1) = 0,33$, where L represents the number of links between two concepts in the ontology.

Thus, the context element match proceeds as follows: for each c_i and c_j , we (i) match semantically the entity of c_i with the entity of c_j ; if the matching score between them is higher than 0,33 then we (ii) match the attribute of c_i with the attribute of c_j ; if the matching score between them is higher than 0,33 then we (iii) match the different values one by one. The matching of the context attribute takes into account the weight assigned to it as explained in the section contextual description. The final score of the attribute

matching is equal to the weight assigned to it multiplied by the score of matching between them. More details about the service discovery mechanism are presented in Najjar et al. (2012).

INTENTION PREDICTION MECHANISM

In this chapter, we detail an approach predicting the future user's intention (\mathbf{I}). This approach provides proactively a service (S_v) that can fulfil the user's future needs. Indeed, this approach is based on the assumption that common situations (\mathbf{S}) can be detected, even in a dynamic and frequently changing Pervasive Information System. Based on this assumption, this prediction mechanism considers a set of time series representing the user's observed situation. We define the notion of situation (\mathbf{S}) as the user's intention (\mathbf{I}), in a given context (\mathbf{C}), satisfied by a specific service. These observations are time stamped and stored in a database after each service discovery process (history). Thus, by analysing the history (\mathbf{H}) represented by the triplet $\langle \text{intention, context, service} \rangle$, the prediction mechanism can learn the user's behaviour model (\mathbf{M}) in a dynamic environment, and thus deduce its future immediately intention.

Figure 8: Service Prediction Mechanism

Two main processes compose this intention prediction mechanism: the *learning process* and the *prediction process*, as illustrated in Figure 8. In the learning process, similar situations (\mathbf{S}) are grouped into clusters, during the *clustering step*. It is a way to reduce the size of the history log by looking for recurring situations. 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 step*, aims to represent, from the recognized clusters, the user's behaviour model (\mathbf{M}) based on his situations (\mathbf{S}). By interpreting situation changes as a trajectory of states, we can anticipate his future needs. In our approach, this process consists of estimating the probabilities of moving from one situation to other possible future situations.

The intention prediction process is based on the user's behaviour model (\mathbf{M}), on the current user's intention (\mathbf{I}) and the current user's context (\mathbf{C}). Based on this information, the prediction process allows predicting the user's future needs. Thus, it provides him a service that can meet his needs in a fairly understandable way.

Before detailing these processes, we should describe the structure of the history used by these processes. This represents the trace management, described in the following section.

Trace Management

The service discovery process is based on the current user's intention (\mathbf{I}) and context (\mathbf{C}) in order to find the most appropriate service. The service, which best meets the immediate user's intention in his current context, is selected. We define the notion of situation (\mathbf{S}) as follows:

Situation = $\langle \text{intention, context, service} \rangle$

Indeed, the intention prediction mechanism is based not only on the current situation of the user, but also on its previously observed situations. As a consequence, these observations may be saved for future needs, such as the intention prediction. Therefore, we refer to time series of observed situations as the user's history (\mathbf{H}). Each time series represents a time stamped observed situation, as illustrated in Table 1.

Time/Date	Intention	Context	Service
t_1	intention ₁	context ₁	service ₁
t_2	intention ₂	context ₂	service ₂
t_3	intention ₃	context ₃	service ₃
...

t_i	intention_i	context_i	service_i
...
t_n	intention_n	context_n	service_n

Table 1: The Structure of the User's History

Whenever a service is selected, the situation of the user is registered at the end of the history base in order to keep a trace of the user's past situations. The **intention** is represented as an XML schema containing two mandatory elements, namely the *verb* and the *target*. Then, the **context** is also represented as an XML schema containing the context description. Finally, the **service** represents the name of the service selected to satisfy this intention in this context.

Let the user's history \mathbf{H} be defined as a set of all the observed situations ξ_i ordered according to their time of occurrence.

$$\mathbf{H} = \{\xi_i \mid i \in [1, n] \text{ with } n \text{ the history size}\}$$

Each observation ξ_i represents a user's situation \mathbf{S}_i , observed at the time t_i :

$$\xi_i = \{ \langle \mathbf{S}_i, t_i \rangle \mid \forall i \in [1, n], \mathbf{S}_i \in \mathbf{H} \wedge \text{TimeStamp}(\mathbf{S}_i) = t_i \}$$

The user's observed situation \mathbf{S}_i is composed of the user's intention \mathbf{I}_i , his context \mathbf{C}_i and the selected service \mathbf{S}_{vi} at the time t_i .

$$\mathbf{S}_i = \{ \langle \mathbf{I}_i, \mathbf{C}_i, \mathbf{S}_{vi} \rangle \mid \forall i \in [1, n], \mathbf{I}_i, \mathbf{C}_i, \mathbf{S}_{vi} \in \mathbf{H} \wedge \text{TimeStamp}(\mathbf{I}_i, \mathbf{C}_i, \mathbf{S}_{vi}) = t_i \}$$

Thus, maintaining the log of the user's observed situations helps the learning process in order to deduce the user's behaviour. This learning process will be explained in the following section.

Learning Process

To realize anticipatory and proactive behaviour of PIS, we first need first to dynamically learn about the user and his behaviour in a frequently changing environment. This represents an important step for the prediction mechanism.

The learning process is based on the analysis of this history in order to reduce the size of existing data. We proceed by grouping the different observed situations into clusters of similar situations and learn the user's behaviour model. It is triggered independently of the prediction step, and may be characterized as a background task that runs periodically. As a result, this process is responsible for dynamically determining the user's behaviour model (*classification*) from the recognized clusters representing similar situations (*clustering*).

Clustering

The first step of our intentional prediction mechanism is the clustering of user's logs. As the history log contains several user's observed situations, it is likely that some of them are similar. Since the size of this history in a dynamic environment can be quite large, clustering similar situations for a user represents an appropriate solution to reduce the data size. Also, the analysis of the clusters allows a better definition on user's habits, which can improve the accuracy of our prediction mechanism. The input of this step represents vectors representing user's situation stored in the history (Table 1).

	KM	FKM	SOM	NG	GNG
Online	✓	✓	✓	✓	✓
Adaptability	✓		Variants		✓
Soft Classification		✓	✓		✓

Limited Resource	✓	✓	✓	✓	Depends on growing
Privacy	✓	✓	✓	✓	✓
Real-Time Execution	---	---	--	-	-
Unbounded Clusters					✓
Variable Topology					✓

Table 2: Clustering algorithms comparison (based on Mayrhofer et al.2003; 2004)

The main task of the clustering is to detect recurrent situations (S) from all the situations observed before. In fact, the clustering is responsible for determining the situation that is the closest to a set of situations corresponding to highly similar intentions in quite similar context. This provides us with a powerful mechanism to evaluate the user's intention. A user can express the same intention in a slightly different way by using verbs and targets that are semantically similar enough. Based on verb and target ontologies, we perform a semantic matching between two intentions in order to determine their degree of similarity. On the other hand, the user's context represents highly heterogeneous data: numerical, nominal, qualitative, etc. In addition, the same context element class may have different representations (e.g., the location can be expressed as GPS coordinates, postal address, predefined location, etc.). Thus, to compare two context descriptions, we also use a semantic matching between the context elements. This is based on using similarity measures between the values of context element. Therefore, the clustering will help to find these situations and represent them by one common situation that is closest to all the members of the same cluster.

However, to better adapt to the PIS, the clustering algorithm must meet certain requirements. These are shown in Table 2, based on Mayrhofer et al. (2003, 2004). It represents some essential criteria for pervasive information systems:

- **Unsupervised:** Clusters must be trained in an automatic manner without prior knowledge and without the help of the user;
- **Adaptable:** The clustering process needs to update the clusters already recognized as the user's behaviour can change;
- **Offline:** Clusters must be updated regularly without hindrance to the normal functioning of the system, suggesting a strategy 'offline'. This can be based on a clustering parameter that defines after how long time this process will be triggered. This parameter can be defined according to the dynamics of systems that employ them;
- **Privacy:** We must take into account that the user prefers that some context information will not be used in the clustering process;
- **Limited Resources:** We must consider the capacity constraints of the application in which the algorithm may be deployed. Algorithms consuming fewer resources are recommended.

Given the dynamic of PIS and in order to cope with changes in the dimensionality of the inputs, the clustering algorithm must be *unsupervised* with a *variable topology*. Since the main objective is to minimize user intervention, clustering must be *unsupervised*. It should not ask for a priori knowledge about the clusters to be recognized and should be able to adapt itself dynamically when a change occurs. Moreover, in order to reduce costs, while keeping up-to-date the clusters, the clustering algorithm should be *offline* using a clustering parameter. This parameter defines after which time we can handle the clustering process. It must be set according to the dynamics of the system in question.

Table 2 illustrates a comparison between different clustering algorithms. Through this table, we can observe that the algorithms K-Means (KM) (Daszykowski et al., 2002) and Fuzzy K-Means (FKM) (Nelles, 2001) can not be applied in our case. Indeed, they require a priori knowledge about the clusters to learn and have a relatively high real-time execution. In addition, Fuzzy K-Means algorithm does not adapt itself dynamically to change. The NG (Martinetz et al., 1993), which also presents an extension of K-

means taking into account the property of neighbor's classification, requires a priori specification of the number of cluster to use. This constraint has led to the elimination of NG since it is difficult to determine the number of clusters a priori in a dynamic pervasive environment. Moreover, SOM (Self-Organizing Map) (Kohonen, 1995) can also be eliminated for the same reasons that K-Means. Furthermore, according to Mayrhfer et al. (2003), SOM tend to forget quickly the clusters previously recognized due to its learning strategy and not the variability of the topology.

Therefore the algorithm GNG (Growing Neural Gas) (Daszykowski et al., 2002) seems to be the most appropriate candidate, since it is closest to our criteria. It adapts itself to the dynamics of the environment, does not require knowledge a priori and has a reasonable real-time execution. The GNG (Daszykowski et al., 2002), compared to other algorithms, offers more flexibility, allowing it to cope with frequent changes in PIS.

The Growing Neural Gas (GNG) shares the same structure with many neural networks. The role of GNG is to recognize and update a set of clusters according to the input vector. It connects the input to a set of outputs nodes that we called 'clusters'. The GNG apply the neighbour property by connecting some neighbour nodes together. Applied to our clustering step, the input represents user's situation composed by an intention, a context and a service. The output represents the recognised clusters representing similar situations.

Once the clustering process is completed, recognized clusters are then interpreted as states of the user's behaviour model. This is the classification process, presented in the next section.

Classification

In a pervasive environment, users follow a set of behaviour schemas that change over time and depend on the user's situations (S). The user cannot be described accurately in advance. Therefore, a dynamic user's behaviour model is necessary. It must be able to adapt to user's change and take into account the probabilistic nature of his behaviour.

From the recognized clusters and the user's history, the classification module determines and maintains a user's behaviour model. This model represents the user's behaviour as a set of states with a transition probability. This probability determines the probability of moving from one state to another.

Similar to the clustering algorithms, classification algorithms pose some requirements. These algorithms must follow the change and the dynamic of pervasive environments, and therefore adapts themselves accordingly. Moreover, in such environment, it is difficult to establish a priori knowledge about the user's behaviour. Thus, among the requirement necessary for a better classification in a pervasive environment, we can list:

- **Unsupervised:** the model must be estimated in an automatic manner without a prior knowledge and without the help of the user;
- **Online:** the model must continuously adapt itself to user's change;
- **Incremental:** When a new cluster is recognized, the model must increase its internal structure incrementally, without requiring a full learning;
- **Heterogeneous and multidimensional data:** the user's situations are represented by heterogeneous data that can be of nominal, ordinary, numeric, etc. These different types of data must be taken into account;
- **Memory and load processes:** in a PIS, a classification algorithm may be deployed on different mobile devices with limited memory capacity often.

Several classification techniques exist. Among these techniques, we note the Bayesian network (BN) (Friedman et al., 1997), Markov Chain (Feller, 1968), Hidden Markov Model (HMM) (Rabiner, 1989), ARMA (Hsu et al., 1998), Support Vector Machines (SVM) (Burges, 1998), Active Lempel Ziv (ALZ) (Gopalratnam & Cook, 2003). Firstly, the BN (Friedman, et al., 1997) works with discrete variables. It requires a priori knowledge and must specify, from the beginning, the different states and hidden variables, which does not meet the above prerequisites. Then, the SVM (Burges, 1998) is a classification method treating only numerical data. In addition, it requests a fixed size of the space of input data.

ARMA (Hsu et al., 1998), meanwhile, represents one of the most efficient and most appropriate classification techniques in our field. Nevertheless, the major drawback is its limitation in numeric data processing, making difficult its application to intention and some contextual data having a symbolic nature. The HMM (Rabiner, 1989) represents one of the well-known classification technique. However, this technique can not be applied in a pervasive environment, which requires a dynamic and automatic adaptation to changes, mainly due to the its supervised method.

Thus, Markov chains (Feller, 1968) are more suitable than the HMM for its unsupervised and online characteristic. Moreover, Markov chains are able to classify multidimensional and heterogeneous data in a pervasive environment. Therefore, Markov chains are the most suitable candidates for Pervasive Information Systems, which best meet the criteria outlined above.

The Markov chain (Feller, 1968) is a well-known method for representing a stochastic process in discrete time with discrete state space. We represent the Markov chains model (M) as the doublet $M = (\mathbf{S}, p)$, with \mathbf{S} 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) $s \in \mathbf{S}$ representing its intention in a given context. In a pervasive information system, the intention of the user and his context may change. Therefore, the user moves from the situation s to the situation $s' \in \mathbf{S}$. The situation s' is the successor state of s with a certain probability p . This transition probability represents the ratio of the transition from s to s' divided by the number of all the possible transitions from s . This probability is represented as follows:

$$p(s, s') = P(X_{t+1} = s' | X_t = s) = \frac{N_{ss'}}{\sum_{s'' \in \mathbf{S}} N_{ss''}}$$

The prediction process, described in the next section, is mainly based on the results of the classification to predict the next user's intention.

Prediction Process

A more proactive behaviour can be obtained with the prediction of future user's needs. The purpose of this prediction process is to predict the future user's intention in order to propose him the next service that can meet his future intention. This process is triggered when the user sends his intention to the IPSOM middleware. Based on the user's intention (\mathcal{I}) and his current context (\mathcal{C}), IPSOM is able not only to select the service that best fulfils his immediate needs (service discovery), but also to propose him the next step (service prediction). Based on the user's behaviour model (M), his current intention and context, the prediction module is responsible for finding the state that is the most similar to the current situation of the user.

Figure 9: Service Prediction Algorithm

The Figure 9 illustrates our proposed algorithm for predicting the future user's intention and consequently the most appropriate next service. The line 9 of Figure 9 shows the first step of the prediction process. It illustrates the semantic matching between the intention and context of each state of the model with the user's immediate intention and context. First, this step is based on a semantic matching between the user's intentions and the intention of the state. As mentioned above, an intention consists of a verb and a target. The semantic matching of intentions is therefore based on ontologies describing these elements in order to calculate the matching score between them. Then, the algorithm performs a semantic matching between the user's context description and the context descriptions of the different states of the model. This matching is based on a domain-specific ontology and on similarity measures between the values of context (see Najar et al. (2011) 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. Going through all the states of the model, we can determine the state the most similar to the current user's situation (line 13).

Subsequently, if a state is identified, IPSOM is responsible for selecting the next state based on the transition probabilities (line 14). This transition probability must exceed a certain threshold. If several successor states are retrieved, then the one having the highest transition probability is chosen. By this choice, we derive the successor state, which represents the future user's intention in a given context. We anticipate the user's future needs by offering him the most appropriate service that can interest him.

When a new service is added to the semantic service directory, IPSOM checks whether this new service can best respond to the situations represented as a state in the user's behaviour model. In this case, the state is updated with the new service. Therefore, the service to be offered to the user during the prediction process remains the most appropriate service according to the user's intention in its context of use.

DISCUSSION AND FUTURE RESEARCH DIRECTIONS

According to Weiser (Weiser, 1991), *pervasive systems* are characterised by their *transparency*. In 1991, Mark Weiser described “*the most profound technologies as those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it*”. Twenty years later, it is clear that we have not reached the homogeneity and invisibility described by Weiser. However, this does not mean that pervasive computing is not already a reality. According to Bell & Dourish (2007), pervasive computing has simply taken a different form than expected by Weiser, in which mobile devices represent the central element of our everyday life. In fact, we interact with a variety of devices and services offered by Information Systems (IS) surrounding us. While great efforts have been focused on context adaptation, especially on location and on used devices, today we can observe the limitations of these approaches.

More specifically, in 2007, Kourouthanassis & Giaglis define pervasive information systems as: “*interconnected technological artefacts diffused in their surrounding environment, which work together to sense, process, store, and communicate information to ubiquitously and unobtrusively support their user's objectives and tasks in a context-aware manner*”. However, due to the complexity, heterogeneity and lack of transparency of our environment, the transition from a *controlled information system* to an *information system available anytime and anywhere*, is still in progress. Moreover, current PIS propose to user different implementations for the same service. Thus, the user, requesting services offered by IS, focuses no longer on his real requirement and on the tasks that really interest him. He finds himself spending time to understand the offered choices in order to select the best implementation, which affects the transparency of such pervasive information system.

An important challenge in Pervasive Computing and especially for Pervasive Information System is to achieve more transparency and invisibility. This is indeed order to make these systems more productive and efficient in a dynamic environment. The real challenge is to hide the complexity of existing systems in order to reduce the intervention and the effort of users by making PIS non-intrusive. By satisfying the user's needs in an invisible manner, PIS can let the user focus on more important tasks that really interest him, instead of spending his time to understand what is happening around him in order to choose the best service implementation.

Thus, the user-centred vision of PIS that we propose represents an interesting step towards the expected transparency. We are strongly convinced that this new vision of PIS represents a more intelligent and personalized systems that concentrate especially on user's requirements in order to satisfy his needs with the most effective way. In fact, what we present here is a proactive PIS based on an intentional prediction approach representing user's requirement as an intention. This intention is emerged, and is more significant, in a specific context. Thus, by considering the *intention* a user wants to satisfy and the *context* on which this intention emerges, we contribute to the improvement of the transparency of PIS, by hiding its complexity and letting the user concentrate on his real tasks.

The user-centred PIS proposed in this chapter presents new opportunities to be explored in different area. Then, and in order to make these systems more personalized and intelligent, a robust mechanism is

needed in order to discover user's requirements, and design new services accordingly. This comes from the analysis of user's needs based on requirement elicitation methods. Besides, we consider that a service composition process has to be explored in such systems in order to cope with more complex user's requirements and services. Thus, the specification of a service composition process becomes essential for the improvement of this user-centred PIS.

Finally, an evaluation of our proposed intentional prediction mechanism with a real case study will be conducted in order to demonstrate its validity, efficiency and precision.

CONCLUSION

Nowadays, our environment is characterized by the evolution of pervasive technologies and the growth of services offered to the user. However, pervasive information systems that derived from using current IS on pervasive environments are quite complex requiring an important user's understanding effort. Indeed, user still has to understand by his own the different service implementations offered by the system and choose the one that is most appropriate to his needs.

Therefore, we propose in this chapter a user-centred vision of PIS based on an intentional prediction approach in a space of services, in order to hide PIS complexity. This approach allows us to anticipate the future user's needs, in order to propose a service that can interest him in a fairly understandable and less intrusive way. By this approach, we believe contributing to the improvement of PIS transparency and productivity through a user-centred view. This view perceives the PIS by the intentions it allows the user to satisfy in a given context.

Thus, we propose an intentional prediction mechanism guided by the context, being integrated in our proposed IPSOM middleware. This prediction mechanism allows: (i) clustering similar user's situations in a set of clusters, (ii) learning the user's behaviour model according to recognized clusters and user's history (iii) deducing the user's future intention based on his behaviour's model and on his current context and intention.

This intention prediction mechanism highlights the anticipatory and proactive behaviour of our proposed vision of PIS. We strongly believe that an intentional prediction approach can answer to transparency and homogeneity requirements, necessary for fully acceptance of Pervasive Information System.

Currently, we are finishing the first implementation of the prediction and learning module and its integration in our proposed Intentional & Pervasive Service Oriented Middleware (IPSOM). Based on this implementation, we plan to evaluate our intentional prediction mechanism under a real usage scenario in order to: (i) verify the behaviour of our system with different services under different context configurations; and (ii) demonstrate its validity, efficiency and precision. Besides, we are currently working on a methodology for setting the clustering parameter. This parameter should be customized according to the system and its use.

REFERENCES

Abbar, S., Bouzeghoub, M., & Lopez, S. (2009). Context-aware recommendation systems: a service-oriented approach. *35th International Conference on Very Large Data Bases (VLDB)*, France

Adomavicius, G., Sankaranarayanan, R., Sen, S., & Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transaction on Information Systems (TOIS)*, 23(1), 103-145

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 17(6), 734-749

- Bell, G., & Dourish, P. (2007). Yesterday's tomorrows: notes on ubiquitous computing's dominant vision. *Personal and Ubiquitous Computing*, 11(2), Springer London, 133- 143
- Bonino da Silva Santos, L.O., Guizzardi, G., Pires, L.F., & Van Sinderen, M.(2009).From User Goals to Service Discovery and Composition.*ER Workshops*, 265-274
- Boytsov, A., Zaslavsky, A., & Synnes, K. (2009). Extending context spaces theory by predicting run-time context. *In Proceedings of the 9th International Conference on Smart Spaces and Next Generation Wired/Wireless Networking and Second Conference on Smart Spaces, NEW2AN '09 and ruSMART '09*, 8–21, Berlin, Heidelberg, Springer-Verlag
- Boytsov, A., & Zaslavsky, A. (2010) Extending context spaces theory by proactive adaptation. *In Proceedings of the Third conference on Smart Spaces and next generation wired, and 10th international conference on Wireless networking, ruSMART/NEW2AN'10*, 1–12, Berlin, Heidelberg, Springer-Verlag
- Boytsov, A., & Zaslavsky, A. (2011). Context prediction in pervasive computing systems: Achievements and challenges. *In Frada Burstein, Patrick Brezillon, and Arkady Zaslavsky, editors, Supporting Real Time Decision- Making*, volume 13 of Annals of Information Systems, 35–63, Springer US
- Burges, C. J. C. (1998).A tutorial on support vector machines for pattern recognition.*Data Mining and Knowledge Discovery*, 2(2), 121–167
- Chen, I. Y. L., Yang, S. J. H., & Jiang, J. (2006) Ubiquitous provision of context aware web services.*IEEE International Conference on Services Computing (SCC)*, USA, 60-68
- Daszykowski, M., Walczak, B., & Massart, D.L. (2002).On the optimal partitioning of data with k-means, growing k-means, neural gas, and growing neural gas. *Journal of Chemical Information and Computer Sciences*, 42 (1), 1378–1389
- Dey, A. (2001).Understanding and using context.*Personal and Ubiquitous Computing*, 5(1), 4-7
- Dik, S.C. (1989). *The theory of functional grammar*.Dordrecht/Nederthlands:*Foris publications*
- Feller, W. (1968).*An Introduction to Probability Theory and its Applications*.New Jersey: Wiley
- Fillemore, C.J. (Ed.). (1968).*The case for case, in Universals in linguistic theory*. New York: Holt, Rinehat and Winston, E.Bach/R.T.Harms
- Friedman, N., Geiger, D.,& Goldszmidt, M. (1997). Bayesian Network Classifiers.*Machine Learning*, 29(2-3), 131- 163
- Gopalratnam, K., & Cook, D. J. (2003). Active Lezi: An incremental parsing algorithm for sequential prediction.*Proceedings of the Florida Artificial Intelligence Research Symposium (FLAIRS)*, 38-42
- Hong, H., Suh, E.H, Kim, J., & Kim, S. (2009). Context-aware system for proactive personalized service based on context history. *Expert Syst. Appl.*, 36(4), 7448–7457
- Hsu, W.H., Gettings, N.D., Lease, V.E., Pan, Y., & Wilkins, D.C. (1998). Heterogeneous time series learning for crisis monitoring. In Predicting the future: ai approaches to time-series problems,*Workshop held in conjunction with the fifteenth national conference on artificial intelligence*, 98, 34–41

- Jackson, M. (1995). *Software Requirements and Specifications: A lexicon of practice, principles and prejudices*. Addison Wesley Press, 256
- Jiménez Molina, A., Koo, H.M., & Ko, I.Y. (2007). A Template-Based Mechanism for Dynamic Service Composition Based on Context Prediction in Ubicomp Applications. *International Workshop on Intelligent Based Tools IWBT, IEEE ICTAI*
- Kaabi, R.S., & Souveyet, C. (2007). Capturing intentional services with business process maps. *1st IEEE International Conference on Research Challenges in Information Science (RCIS)*, 309-318
- Kirsch-Pinheiro, M., Gensel, J., & Martin, H. (2004). Representing Context for an Adaptive Awareness Mechanism. G.J. de Vreede; L.A. Guerrero, G.M. Raventos (Ed), *LNCS 3198 - X Workshop on Groupware (CRIWG)*, 339-348
- Kirsch-Pinheiro, M., Vanrompay, Y., & Berbers, Y. (2008). Context-aware service selection using graph matching. *2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC'08), ECOWS. CEUR Workshop proceedings*, 411
- Kohonen, T. (1995). *Self-Organising Maps*. 30, Springer
- Kouruthanassis, P.E., & Giaglis, G.M. (2006). A design theory for pervasive information systems. *3rd International Workshop on Ubiquitous Computing (IWUC)*, 62-70
- Lee, K.C., & Cho, H. (2010). A general bayesian network-assisted ensemble system for context prediction: An emphasis on location prediction. In Tai-Hoon Kim, Young-Hoon Lee, ByeongHo Kang, and Dominik Slezak (Ed.), *FGIT: Vol 6485 of Lecture Notes in Computer Science* (pp. 294–303), Springer
- Martin, D., Paolucci, M., McIlraith, S., Burstein, M., Mcdermott, D., Mcguinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Srinivasan, N., & Sycara, K. (2004). Bringing Semantics to Web Services: The OWL-S Approach. Cardoso, J. & Sheth, A. (Ed.), *SWSWPC 2004, LNCS 3387*, Springer, 26-42
- Martinetz, T.M., Berkovich, S.G., & Schulten, K.J. (1993). 'Neural-Gas' network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4 (4), 558–569
- Mayrhofer, R., Harald, R., & Alois, F. (2003). Recognizing and Predicting Context by Learning from User Behaviour. *International Conference On Advances in Mobile Multimedia (MoMM2003)*, In W. Schreiner, G. Kotsis, A. Ferscha, and K. Ibrahim (Ed.), pp. 25–35
- Mayrhofer, R. (2004). *An Architecture for Context Prediction*. PhD thesis, Johannes Kepler University of Linz
- Meiners, M., Zaplata, S., & Lamersdorf, W. (2010). Structured context prediction: A generic approach. In R. Kapitza F. Eliassen (Ed.), *Proceedings of the 10th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS 2010)* (pp. 84–97). IFIP, 6: Springer
- Mirbel, I., & Crescenzo, P. (2010). From end-user's requirements to Web services retrieval: a semantic and intention-driven approach. J.-H. Morin, J. Ralyte, M. Snene, "Exploring service science," *First International Conference on Exploring Services Sciences (IESS)*, 30-44, Springer,
- Najar, S., Saidani, O., Kirsch-Pinheiro, M., Souveyet, C. & Nurcan, S. (2009) Semantic representation of context models: a framework for analyzing and understanding. J. M. Gomez-Perez, P. Haase, M. Tilly,

and P. Warren (Ed.), *1st Workshop on Context, information and ontologies CIAO, European Semantic Web Conference ESWC'2009* (pp. 1-10). ACM

Najar, S., Kirsch-Pinheiro, M., & Souveyet, C. (2011). The influence of context on intentional service. *5th International IEEE Workshop on Requirements Engineering for Services (REFS)- IEEE Conference on Computers, Software and Applications (COMPSAC)*, Germany, 470–475

Najar, S., Kirsch-Pinheiro, M., & Souveyet, C. (2011). Towards semantic modeling of intentional pervasive information systems. *6th International Workshop on Enhanced Web Service Technologies WEWST*, 30-34

Najar, S., Kirsch-Pinheiro, M., Souveyet, C., & Steffemel, L.A (2012). Service Discovery Mechanism for an Intentional Pervasive Information System. *Proceeding of 19th IEEE International Conference on Web Services (ICWS)*, Honolulu, Hawaii, 520-527

Nelles, O. (2001). *Nonlinear system identification*. Berlin/Germany: Springer

Nurmi, P., Martin, M., & Flanagan, J. A. (2005). Enabling proactiveness through Context Prediction. *In Proceedings of the Workshop on Context Awareness for Proactive Systems*, Finland, 159-168

Paolucci, M., Kawamura, T., Payne, T., & Sycara, K. (2002). Semantic matching of web services capabilities. *In Proceedings of the First International Semantic Web Conference*, Springer LNCS 2342, Sardinia, Italy.

Petzold, J. (2005). *State Predictors for context prediction in Ubiquitous Systems*. PhD thesis, University of Augsburg

Petzold, J., Bagci, F., Trumler, W., & Ungerer, T. (2005). Next location prediction within a smart office building. *1st Int. Workshop on Exploiting Context Histories in Smart Environments (ECHISE05)*, *3rd International Conference on Pervasive Computing*

Prat, N. (1997). Goal formalisation and classification for requirements engineering. In Proc. of the *3rd International Workshop on Requirements Engineering: Foundations of Software Quality (REFSQ'97)*. E. Dubois, A.L. Opdahl, K. Pohl. (Ed.), Presses Universitaires de Namur

Rabiner, L.R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77, 257–286

Ramadour, P., & Fakhri, M. (2011). Modèle et langage de composition de services. *INFORSID*, 59-76

Reichle, R., Wagner, M., Khan, M., Geihs, K., Lorenzo, L., Valla, M., Fra, C., Paspallis, N., & Papadopoulos, G.A. (2008). A Comprehensive Context Modeling Framework for Pervasive Computing Systems. *In 8th IFIP Conference on Distributed Applications and Interoperable Systems (DAIS)*, Springer

Rolland, C., Kirsch-Pinheiro, M., & Souveyet, C. (2010). An Intentional Approach to Service Engineering. *IEEE Transactions on Service Computing*, 3(4), 292-305

Sigg, S. (2008). *Development of a novel context prediction algorithm and analysis of context prediction schemes*. PhD thesis, Kassel University

Sigg, S., Haseloff, S., & David, K. (2010). An Alignment Approach for Context Prediction Tasks in UbiComp Environments. *IEEE Pervasive Computing*, 9(4), 90-97

Truong, H. L., & Dustdar, S. (2009). A survey on context-aware web service systems. *International Journal of Web Information Systems*, 5(1), 5-31

Truong, H. L., & Dustdar, S. (2010). Context coupling techniques for context-aware web service systems: an overview. In *Enabling Context-Aware Web Services: Methods, Architectures, and Technologies, 1th ed.*: Chapman and Hall/CRC, 337-364.

Vanrompay, Y. (2011). *Efficient Prediction of Future Context for Proactive Smart Systems*. PhD Dissertation, Katholieke Universiteit Leuven

Weiser, M. (1991). The computer of the 21st Century. *Scientific American*, 265(3), 94-104

Xiao, H., Zou, Y., Ng, J., & Nigul, L. (2010). An Approach for Context-Aware Service Discovery and Recommendation. *17th IEEE International Conference on Web Services (ICWS)*, 163-170

Yang, S. J. H., Zhang, J., & Chen, I. Y. L. (2008). A JESS-enabled context elicitation system for providing context-aware Web services. *Expert Systems with Applications*, 34 (4), 2254-2266

KEY TERMS & DEFINITIONS

Pervasive Information System (PIS) is a new vision of Information Systems available anytime and anywhere, while adapting itself to user's context

Context can be defined as any information that can be used to characterize the situation of an entity (a person, place, or object considered as relevant to the interaction between a user and an application)

Intention can be defined as a user's requirement that represents a goal that a user wants to be satisfied by a service without saying how to perform it.

Service Prediction is the process allowing the anticipation of the user's needs and the selection for him the service that can interest him and that can answer to his future needs.

Service Discovery is the process allowing to find and to select a service, among the available ones, that answers to an immediate user's request.

Clustering is the process that tries to group a set of object into clusters whose members are similar in some way.

Classification is the process that determines, based on predefined set of cluster, which cluster a new object belongs to