A context-aware intentional service prediction mechanism in PIS

Salma Najar, Manuele Kirsch Pinheiro, Carine Souveyet

A context-aware intentional service prediction mechanism in PIS

Salma Najar, Manuele Kirsch Pinheiro, Carine Souveyet
Centre de Recherche en Informatique - University Paris1 Panthéon-Sorbonne
90 rue de Tolbiac, 75013 Paris – France
Salma.Najar@malix.univ-paris1.fr,
{Manuele.Kirsch-Pinheiro, Carine.Souveyet}@univ-paris1.fr

***Abstract*—Pervasive Information System (PIS) represents a new generation of Information Systems (IS) available anytime, anywhere in a pervasive environment. In this paper, we propose to enhance PIS transparency and efficiency through a context-aware intentional service prediction approach. This approach allows anticipating user's future needs, offering and recommending him the most suitable service in a transparent and discrete way. We detail in this paper our service prediction mechanism and present encouraging experimental results demonstrating our proposition.**

Keywords-pervasive information system; service orientation; intention; context-aware; prediction; clustering; classification

I. INTRODUCTION

Pervasive Information Systems (PIS) represent an evolution of Information Systems (IS) towards the integration of pervasive environments. Currently, pervasive environments are mainly reactive. Decisions are taken solely based on the current context. Indeed, research in the anticipatory and proactive behavior on pervasive environment, notably by the prediction of the user's future situation, is hardly done. By avoiding focusing on the prediction, current systems lack an important element in the search for transparency and homogeneity. Besides, this research does not consider the user's intentions behind each service request. Consequently, many opportunities can be offered to the user, even if he/she is not always able to understand what is proposed to him and why. Thus, PIS remains too complex for the users, who are just interested in satisfying their needs (and not on how it is done).

We believe that in order to achieve transparency necessary to handle such pervasive environments, the PIS must reduce the user's understanding effort. They must hide the complexity of the multiple available services. This will be possible thanks to a user-centered vision. This vision can be achieved through a service prediction mechanism capable of anticipating future user's needs, improving system proactivity, and thereby contributing to improving the transparency necessary for PIS.

Our purpose is to predict the user's future intention based on his/her context, in order to offer him the most suitable service considering his/her incoming needs. This approach considers PIS through the notion of *intention* that can be seen as the goal that we want to achieve without saying how to perform it [10]. An intention represents a requirement that a user wants to be satisfied without really care about how to perform it or what service allows him to do so. An intention emerges on a giving context. The *context* represents any information that can be used to characterize the situation of an entity [5].

Based on this information, we propose a context-aware intentional service prediction mechanism. The main purpose of such approach is to provide to the user a service that can fulfill his/her needs in a fairly understandable and non-intrusive way, reducing user's understanding effort. This prediction mechanism is based on the assumption that, even in a dynamic and frequently changing Pervasive Information System, common situations can be found. Based on this assumption, this prediction mechanism considers a set of time series representing observed user's situations. Thus, we are able to track and store these situations in a history, after each successful discovery process. By analyzing this history, a prediction mechanism can learn user's behavior in a dynamic environment, and therefore deduce his/her future intention and the most appropriate service that satisfy it.

This paper is organized as follows: the *section II* introduces our vision of an user-centered contextual PIS, while *section III* details the proposed context-aware intentional service prediction mechanism. The *section IV* presents the implementation and the experiments results of the service prediction. The *section V* presents an overview on related works. Finally, we conclude in the *section VI*.

II. A USER-CENTERED CONTEXTUAL VISION OF PIS

In this paper, we introduce our new vision of Pervasive Information Systems [11] [15]. This user-centered vision of PIS allows focusing on the user's needs through an intentional approach. It considers the PIS and their elements both in terms of IS and of pervasive service systems, observing their control, intentionality and context-awareness requirements. This is in order to ensure the necessary transparency and understanding for the design and for the development of PIS.

This vision is characterized by its *context orientation,* which allows a better management of the heterogeneity and dynamics that characterize the pervasive environment. Moreover, in the perspective to better satisfy user's needs and to be at her/his level, our vision is based on an *intention orientation*. Thus, PIS can, on the one hand, better

understand the user's needs, and on the other hand, better meet her/his needs in the most appropriate manner.

Thus, we exploit, in our vision, the close relationship between the notions of intention, context and service, shown in Figure 1. We consider that the satisfaction of the user's intention in a PIS depends on the context in which this user is. For us, the environment directly impacts how to meet the intentions, and so the choice of services to be performed. Therefore, by combining intentional and contextual approaches in service orientation, we propose a new user-centered vision of transparent and non-intrusive PIS that is understandable to the user.

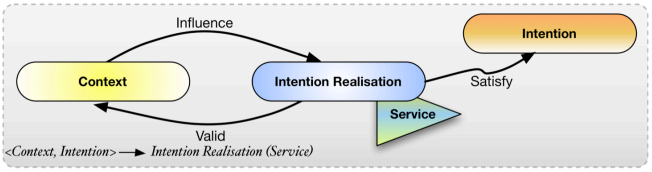

Figure 1. The close relation between context, intention and service

In this paper, we present a context-aware intentional prediction mechanism, presented in the next section, as a part of this vision. This is in order to enhance the PIS transparency, efficiency and proactivity through a user-centered contextual vision of PIS, hiding technical details.

III. CONTEXT-AWARE INTENTIONAL SERVICES PREDICTION MECHANISM

In this paper, we propose a new an approach for predicting the future user's intention (I_U) in a given context (Cx_U). This approach intends for proactively provide a service (Sv_i) that can fulfill the user's future needs. Indeed, this approach is based on the assumption that common situations (S) can be detected, even in a dynamic and frequently changing Pervasive Information System. Based on this assumption, this prediction mechanism considers a set of time series representing a user's observed situation. These observations, represented by the triplet <intention, context, service>, are time stamped and stored in a database after each services discovery process. Thus, by analyzing this history ($\mathcal{H}$), the prediction mechanism can learn the user's behavior model ($\mathcal{M}c$) in a dynamic environment and thus deduce its next intention.

Two main processes compose this intention prediction mechanism: the *learning process* and *the prediction process*, as illustrated in Figure 2. In the learning process, similar situations (S) are grouped into clusters, during the phase of clustering. These clusters are organized as states of a state machine, by the classification phase. It aims at representing, from the recognized clusters, the user's behavior model ($\mathcal{M}c$) based the observed clusters. By interpreting situation changes as a trajectory of states, we can anticipate his/her future needs. Therefore, the intention prediction process is based on the user's behavior model ($\mathcal{M}c$), on the current user's intention (I_U) and the current user's context (Cx_U). Based on this information, the prediction process allows predicting the user's future needs.

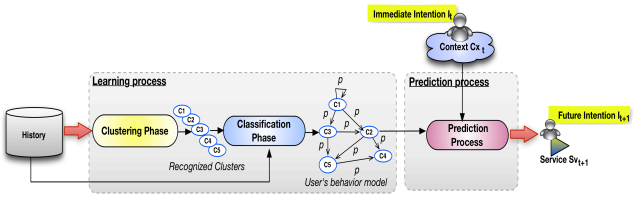

Figure 2. Service Prediction Mechanism

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

A. Trace Management

The history $\mathcal{H}$ is composed by all the results obtained by the service discovery process [15], thereby forming the traces used for prediction. The service discovery process compares the current user's intention (I_U) and context (Cx_U) with those proposed by the available services, in order to propose user the most appropriate service (Sv_i). We define the notion of situation (S_i) as the user's intention (I_U), in a given context (Cx_U), satisfied by a specific service (Sv_i).

$$Si = \{< I_U, Cx_U, Sv_i > \mid \forall\, i \in [1, n],\ I_U, Cx_U, Sv_i \in H \wedge TimeStamp\,(I_U, Cx_U, Sv_i) = ti\} \quad (1)$$

The prediction mechanism is based not only on the current user's situation, but also on its previously observed situations. These observations are saved for future needs. We refer to time series of observed situations as the user's history ($\mathcal{H}$). Each time series represents a time stamped observed situation, as illustrate the Table I.

TABLE I. THE STRUCTURE OF THE USER'S HISTORY

Time/Date	Intention	Context	Service
t_1	I_{U_1}	Cx_{U_1}	Sv_1
t_2	I_{U_2}	Cx_{U_2}	Sv_2
…	..	..	..
t_i	I_{U_i}	Cx_{U_i}	Sv_i
…	..	..	..
t_n	I_{U_n}	Cx_U	Sv_n

Whenever a service is selected, the user's situation is registered on the user's history in order to keep a trace of the user's past situations. The intention (I_U) is represented as an XML schema containing two mandatory elements, namely the verb and the target. Both verb and target are described by ontologies representing respectively significant actions made available by PIS and the objects considered by the actions [14]. The context (Cx_U) is also represented as an XML schema containing the context description. Such descriptions follows an ontology-based context model on which context elements are described by an entity (corresponding to the entities whose context is observed) and a scope representing what is observed (location, memory, etc.) [14]. Finally, the service (Sv_i) represents the name of the service selected to satisfy this intention in this context. Services are described using an extension of OWL-S we have proposed in [14], in

which intention satisfied by the service and the context in which this intention is considered are described.

In the history, the traces represent user's situations (S_i) recorded at a given time. We introduce the notion of observation (O_{Si}), representing a situation of the user S_i observed at the time t_i.

$$O_{Si} = \{<S_i, t_i> \mid \forall i \in [1, n], S_i \in \mathcal{H} \wedge TimeStamp(S_i) = t_i\} \quad (2)$$

Then, we define the history $\mathcal{H}$ as a set of all the observed situations O_{Si} ordered according to their time of occurrence.

$$\mathcal{H} = \{O_{Si}\}, i \in [1,n], \text{with } n \text{ the history size} \quad (3)$$

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

B. *Learning Process*

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

The learning process is based on the analysis of the history ($\mathcal{H}$). We proceed by grouping the different observed situations (O_{Si}) into clusters (CL) of similar situations and then, learn the user's behavior model. It is responsible for dynamically determining the user's behavior model ($\mathcal{M}c$) (*classification*), which illustrates the user's habits, from the recognized clusters (*clustering*). It is with that this process is triggered independently of the prediction process, and may be seen as a background task.

1) *Clustering*

The first phase of our prediction mechanism is the *clustering* of user's traces. As the user interacts daily with PIS, some of his/her situations may be recurrent. These recurring situations can be expressed with similar contexts and intentions. The role of *clustering* here is to consider the relevance of these situations, which can be grouped by similarities in clusters. A *cluster* represents then a set of situations, sharing some similarity between their intentions and contexts. It gathers recurring and similar situations. The cluster analysis allows a better representation of the user's habits, because they are more relevant to treat than separate situations.

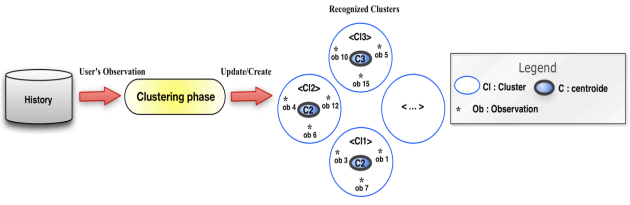

Figure 3. The clustering phase

The input of this phase corresponds to vectors representing user's situations stored in the history (Table I) The main role of clustering, as shown in Figure 3, is to detect recurrent observations among all situations previously observed and grouped in a cluster. A *cluster* consists of a *centroid* and a set of *observations*. The *centroid* represents the identifier of the cluster which symbolizes the observation the most similar to all the observations grouped in this cluster. The *centroid* is defined by the triplet $<I_v, CX_v, Sv_i>$.

In fact, the clustering is responsible for determining the situation that is the closest to a set of situations corresponding to highly similar intentions in quite similar context. This provides us with a powerful mechanism to evaluate the user's intention. A user can express the same intention in a slightly different way by using verbs and targets that are semantically similar enough. Based on verb and target ontologies, we perform a semantic matching between two intentions in order to determine their degree of similarity. On the other hand, the user's context represents highly heterogeneous data. Thus, to compare two context descriptions, we combine a semantic matching between the context elements (scope and entity should be semantically similar) and similarity measures that compare the values of context element. Therefore, the clustering will help to find these situations and represent them by one common situation that is closest to all the members of the same cluster.

In order to fully represent a situation, we attach the selected service for the couple <*Intention*, *Context*>. We are aware that this represents a strong constraint (the concept situation is necessarily coupled to a particular service), but it opens a significant performance advantage, since it is not required to launch the service discovery mechanism during the prediction process. Thus, it is important to regularly update the clusters in order to have the service that best meets the couple intention and context of the situation.

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

2) *Classification*

A user's behavior model intends to reflect the interaction between a user and the PIS and its dynamics. Nonetheless, the user cannot be accurately described in advance. Therefore, a dynamic user's behavior model is necessary. It must be able to adapt to user's change and take into account the probabilistic nature of his behavior.

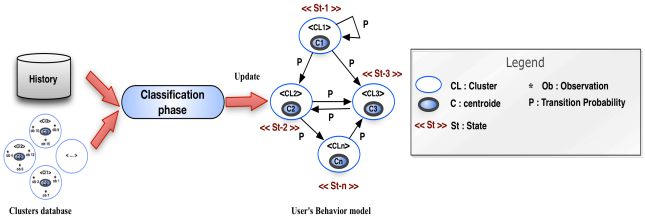

Figure 4. The classification phase

From the clusters recognized in the clustering phase and the history, the classification step determines and maintains a user's behavior model, as illustrated in Figure 4. This phase represents the user's behavior as a set of states with a transition probability. Each *state* is represented by the centroid of a cluster. Each *probability* is calculated based on the history and determines the probability of moving from

one state to another (i.e. the probability that a situation belonging to a cluster A will be followed by a situation belonging to a cluster B).

Several classification techniques exist: Bayesian network (BN) [7], Markov Chain [6], Hidden Markov Model (HMM) [19], Support Vector Machines (SVM) [3], etc. Similar to [12][20], we consider Markov chains [6] as the more suitable method for context classification thanks to its unsupervised and online characteristic. Moreover, Markov chains are able to classify multidimensional and heterogeneous data, which is necessary for classifying intention-context cluster as these are proposed on this paper.

Therefore, Markov [6] chains are the most suitable candidates for PIS. It is a well-known method for representing a stochastic process in discrete time with discrete state space. We represent the Markov chains model ($\mathcal{M}c$) as the doublet $\mathcal{M}c = (St, p)$, with St representing the different states and $p \in [0,1]$ the transition probability.

At a given time t, the user is in a state St_i. In PIS, the user's intention and his context may change. Therefore, the user moves from the state St_i to 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 transition from St_i to St_j divided by the number of all the possible transitions from St_i. This probability is represented in (4).

$$p_{St_i St_j} = P(X_{t+1} = St_j | X_t = St_i) = \frac{N_{St_i St_j}}{N_{St_i St_k}} \quad (4)$$

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

C. Prediction Process

The purpose of this prediction process is to predict the future user's intention in order to propose him the next service. This way, the user would not have to actively request it. This process is triggered when a service discovery process is performed successfully.

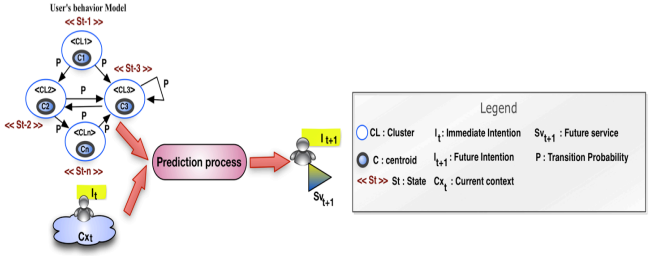

Figure 5. The prediction process

As illustrated by Figure 5, the services prediction process is based on the user's behavior model ($\mathcal{M}c$), which is updated during the previous phase of *classification*. The prediction process is, then, responsible to find the state (St_i) from the model ($\mathcal{M}c$), which is the closest (i.e. semantically similar) of the current user's situation, and to deduce the following situation, which is the most probable. More specifically, and as shown in Figure 6, this process compares semantically, for each cluster represented as a state in the model $\mathcal{M}c$, the immediate intention of the user ($I_{\mathcal{U}}$) with the intention of the state (centroid of a cluster) (I_{Sti}). Then, it compares semantically the current user's context ($Cx_{\mathcal{U}}$) with the context of the state (Cx_{Sti}). If the final score (degree of the contextual and intentional matching) is acceptable (above a certain threshold), then the state is selected as a candidate. Once this processing is done on the set of states in the model $\mathcal{M}c$, and then the state having the highest score is retained. The future service to be offered to the user represents the service of the state, which is held at the end.

Algorithm 1 Context-aware Intentional Service Prediction

```
1.  Procedure ServicePrediction (I_U, Cx_U, Mc)
2.  Result = ∅, State_ranked = ∅
3.  State_Observed = ∅
4.  State_Successor = ∅
5.  Sv_state = ∅, I_state = ∅, Cx_state = ∅
6.  Score = 0
7.  ID = GetStatesID (Mc)
8.  For each id ∈ ID do
9.      Score = Match (I_U, Cx_U, id)
10.     State_ranked.add(id, score)
11. End for
12. If State_ranked is not empty Then
13.     State_Observed = MaxScore(S_ranked)
14.     State_Successor = FindNextState(State_Observed, Mc)
15.     Sv_state = GetService (State_Successor)
16.     I_state = GetIntention (State_Successor)
17.     Cx_state = GetContext (State_Successor)
18.     Result.add(Sv_state, I_state, Cx_state)
19. End If
20. Return Result
21. End procedure
```

Figure 6. The service prediction algorithm

The Figure 6 details the proposed algorithm for predicting the future user's intention and consequently the most appropriate next service. The line 9 of Figure 6 shows the first step of the prediction process. It illustrates the semantic matching between the intention and context of each state of the model with the user's immediate intention and context. First, this step is based on a semantic matching between the user's intentions and the intention of the state. As mentioned above, an intention consists of a verb and a target. The semantic matching of intentions is therefore based on ontologies describing these elements in order to calculate the matching score between them. Then, the algorithm performs a semantic matching between the user's context description and the context descriptions of the different states of the model. This matching is based on a domain-specific ontology and on similarity measures between the values of context (see [15] for more details).

The final matching score represents the sum of the intention matching score and the context matching score. This information is stored with the state identifier. Going through all the states of the model, we can determine the state the most similar to the current user's situation (line 13). If a state is identified, the algorithm select the next state

based on the transition probabilities (line 14). This transition probability must exceed a certain threshold. If several successor states are retrieved, then the one having the highest transition probability is chosen (if there is more than one next state having the highest transition probability, then we choose arbitrary one of them). By this choice, we derive the successor state, which represents the future user's intention in a given context. Thus, we anticipate the user's future needs by offering him the most appropriate service that can interest him.

IV. IMPLEMENTATION AND EVALUATION

We present in this section the implementation of our context-aware intentional service prediction. Then, we discuss our experimental results.

A. Implementation

The services prediction mechanism, proposed in section IV, is based on a history that contain user's traces, the recognized clusters and the user's behavior model. We feed this database by a set of observations stored as traces. These observations represent a set of intentions that the services of the test collection OWLS-TC2 [16] are able to satisfy in the field of travel. Then, we add to these intentions, different contextual descriptions and the service identifier that can meet this intention in the context of use. We identified 10 different context descriptions. Then, we assigned arbitrarily this context description to the defined intentions. These traces were created fictitiously because it was difficult to convince companies to provide us their real data (respect and protection of privacy). The recognized clusters and the user's behavior model are maintained from this history.

This services prediction mechanism was implemented using Java language. It follows the same implementation structure like the service discovery mechanism [15], where the implementations of the processes are organized around three main implementations of interfaces. The *Manager* interface (*IclusteringManager, IClassificationManager* and *IservicePredictionManager*) represents the entry point of the processes. Then, the *PersistenceManager* interface (only used in the clustering and prediction process) acts as a facade between the component that implements the *manager* interface and the ontology directory, which allows access and loading ontologies. Finally, the *Engine* interface (*IclusteringEngine, IclusteringEngine* and *Iprediction Engine*) proposes the necessary methods to implement the different proposed algorithms, such as (i) the classification of the recognized clusters, (ii) the clustering of the user's situations; and (iii) the prediction of the user's future intention and the most appropriate service.

The clustering and prediction algorithms implementations are based on our OWL-S extended API [14], Jena [9] and the reasoner Pellet [18]. These impelentations use two classes, namely *"ContextMatching"* to determine the contextual matching score and the *"IntentionMatching"* class to determine the intentional matching score.

B. Evaluation

As part of our experiments, we deployed our algorithms on a machine Intel Core i5 1.3 GHz with 4 GB memory. As mentioned earlier in this paper, the evaluation of these algorithms has been performed on a semantic directory containing a set of domain-based ontologies and on the constructed history.

The purpose of our experiments is to evaluate the validity of our algorithms and their feasibility. Two main observations emerge from this experiment: (i) *Scalability*: whether the processing time is reasonable; (ii) *Result quality*: whether the algorithm can effectively select the most appropriate services.

In order to evaluate these two measures, we formulate 7 user's requests relatives to the travel domain. These request are represented by the user's intention and his current context, as illustrates in Table II.

TABLE II. USER REQUESTS : AN INTENTION EMERGED IN A GIVEN CONTEXT

Requête	Intention	Current Context
Req 1	- Reserve Five Star European Hotel	**Context User 1 :** -DateTime.Season = Winter, -Profile.Age = 21, -DateTime.Time = Morning, -Location.City = France -Resource.Screen = 16, -Device = Intel Core 2 duo -Resource.Memory = 2048, -Resource.Network = Ethernet
Req 2	- Book-up Lodge	**Context User 2 :** -DateTime.Season = Summer, -Profile.Age = 23 -DateTime.Time = Evening, -Location.City = Tanzania -Resource.Network = Wifi, -Device = Intel Core 2 duo -Resource.Memory = 1024, -Resource.Screen =15
Req 3	- Search BedAndBreakfast	**Context User 1**
Req 4	- Find Contact	**Context User 3** -DateTime.Season = Winter, -DateTime.Time = Night -Profile.Age = 16, -Profile.Expertise = Low -Profile.Role = Student, -Location.City = Mexique -Location.Country = USA, -Resource.Network = Wifi -Device = Iphone 4, -Resource.Memory = 16 -Resource.Screen = 3
Req 5	- Search Destination	**Context User 1**
Req 6	- Locate Surfing Destination	**Context User 4** -DateTime.Season = Summer, -Profile.Age = 24 -DateTime.Time = Morning, -Profile.Expertise = High -Profile.Role = Student, -Location.City = Hawai -Location.Country = USA, -Resource.Network = 3G -Device = Ipad 2, -Resource.Memory = 32 -Resource.Screen = 9
Req 7	Look for Surfing Course	**Context User 5** -DateTime.Season = Summer, -Location.City = Germany -DateTime.Time = Evening, -Profile.Expertise = High -Profile.Role = Student, -Device = Intel Core 2 duo -Profile.Age = 23, -Resource.Network = Ethernet, -Resource.Memory = 2048, -Resource.Screen = 16

These requests are formalized in different ways. We described situations where the elements of the user's intention are not described in the intention ontologies. Nevertheless, there are a set of clusters and states of the user's behavior model that can satisfy this intention in the current user's context (*Req2* and *Req7*). In addition, we described situations where the elements for the user's intention are described in the intention ontologies. These situations are specified in order to demonstrate that the threshold setting, defined by the system designer in the clustering and prediction algorithms, can eliminate some clusters and states even if they are able to satisfy the user's situation (*Req1*, *Req5* and *Req6*). And finally, we evaluated the requests *Req3* and *Req4*, which describe intentions that

can be satisfied by a set of clusters and states in the current user's context, which represents a complex context.

We measure the scalability of our algorithms with respect to the number of clusters, observations in the history and states of the user's behavior model, by measuring the average processing time.

1) Scalability

We measure the scalability of our algorithms with respect to the number of clusters, observations in the history and states of the user's behavior model, by measuring the average processing time.

The execution time of the clustering algorithm was measured by varying the number of clusters already recognized in the database between 7 and 186 clusters. This time represents the average execution time taken by the clustering algorithm to determine which cluster an observation belongs. As illustrated in Figure 7, the execution time following a polynomial trend of degree three varying from 3,6 *s* for 7 clusters to 6,3 *s* for 186 clusters. However, even if this time is a higher, we can observe that despite the fact that we have increased the number of clusters over twenty six times, the response time has only increased by a little more than one and half times. In addition, one of our perspectives is to improve the execution time by optimizing our development code by using the Java threads, for example.

Concerning the execution time of the classification algorithm, it is measured by varying the number of observations already grouped in clusters in the database between 10 and 200 observations. This time represents the average execution time taken by the classification algorithm to dynamically update the user's behavior model. As illustrated in Figure 7, the execution time follows a polynomial trend ranging from 39 *ms* for 10 observations to 398 *ms* for 200 observations. However, even if this algorithm does not take much time to dynamically update the Markov chains, we can notice that rose almost 10 times by increasing the number of observations of twenty times. Thus, introducing the parallel processing in the algorithm can also optimize this.

Finally, the execution time of the prediction algorithm is measured by varying the number of states in the user's behavior model, stored in the database, between 7 and 168 states. This time represents the average execution time set to predict the next service that satisfies a future user's intention according to his immediate intention and current context. As illustrated in Figure 7, the execution time following a polynomial trend of degree three from 2,04 *s* for 7 states to 5,28 *s* for 168 states. We increased the number of states over twenty five times, while the execution time has only increased by two and half times. This allows us to validate the feasibility and scalability of our algorithm. However, these results can be optimized, such as the last two algorithms of clustering and classification.

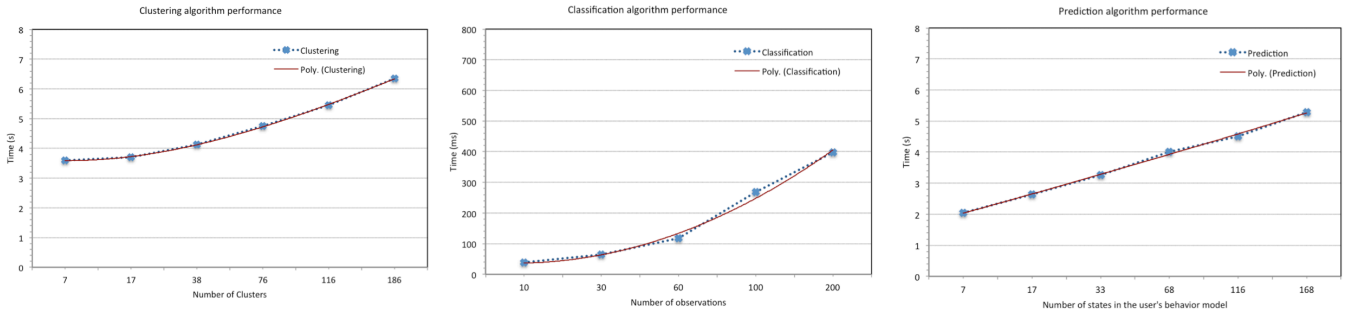

Figure 7. Clustering, classification and prediction algorithms performances

These results allow us to deduce that our algorithms provide a good scalability. However, the performance of our algorithms, especially the clustering and prediction is rather average. But, we remain confident since these algorithms can be optimized in order to improve the execution time. These optimizations are one of our short-term perspectives.

2) Result Quality

In order to measure the quality of the result, we cover the two most useful quality metrics: *precision* and *recall*. These two measures are defined in terms of a set of retrieved item and a set of relevant items. The *precision* represents how well a system retrieves only the relevant services, while the *recall* measures the ability of a system to retrieve all the relevant service [21]. The definition of recall and precision measures are defined as follows:

$$Precision = \frac{|\{relevants\ items\} \cap \{retrived\ items\}|}{\{retrived\ items\}}, Rappel = \frac{|\{relevants\ items\} \cap \{retrived\ items\}|}{\{relevants\ items\}}$$

We consider that the precision and recall metrics are important factors in the analysis of the learning (clustering and classification) and prediction mechanisms. The results presented in Figure 8 indicate that the three algorithms have a satisfying level of precision. The mean precision of the clustering and prediction algorithms is around 88%, whereas the classification algorithm is 100%.

These results indicate that the clustering algorithm is more likely to recognize the right cluster that represents the

user's observation. The classification algorithm represents exactly the dynamics of the user. The prediction algorithm, meanwhile, has a higher chance to select the most appropriate service that satisfy the future user's intention in a similar context to his current context. However, this good results of precision are accompanied by less interesting results (but still satisfaying) on recall, as illustrated in Figure 8. We observe that the average rate of the recall, for the clustering and prediction algorithms, is around 75%.

These results can be explained, for example, by the evaluation of situations where the elements of intention are not described in ontologies, while it exists in the database clusters/states similar to that intention in the user's context. In this case, clustering algorithms and prediction algorithms returns any results. This affects the results quality and return a recall of 0%. In addition, when situations are described by intentions whose verbs and/or targets are fairly generic or specific, some clusters/states that can respond to the trace or to the immediate user's intention in the current context will not be selected. Thus, in some cases we obtain reminders that are below 25%.

The analysis of these results, illustrated in Figure 8, demonstrates that our proposition presents a more interesting result with a higher quality. However, it is important to note that we cannot get that good results only if the system designer: 1) establishes a rich and complete description the different ontologies used; and 2) choose the best threshold setting. We believe that our service prediction mechanism allows selecting the most appropriate future service. And that is due to both its intentional approach, which is more transparent to users, and contextual approach that limits clusters and states to those that are valid.

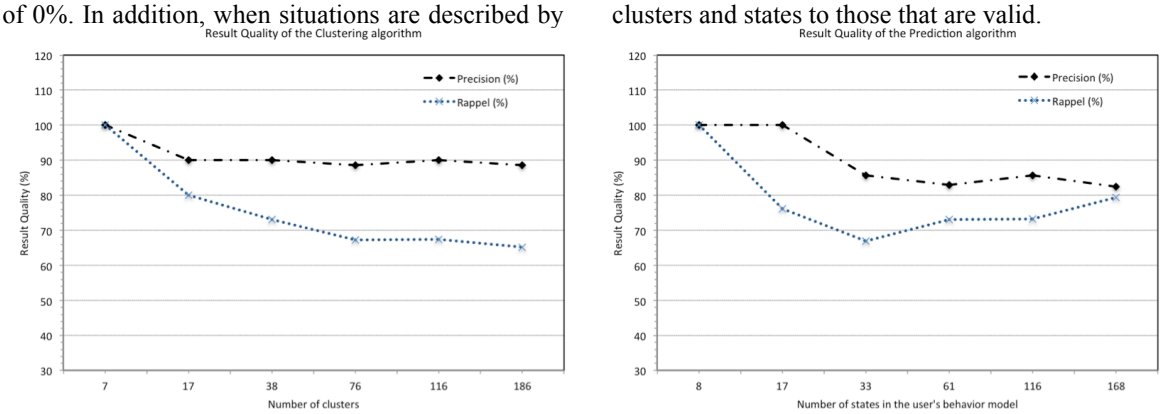

Figure 8. Quality of result of the Clustering and prediction algorithms

V. RELATED WORK

In order to supply users with the desired services, different research on *context prediction* and *context based recommendation systems* are proposed. Mayrhofer [12], Sigg et al. [20] and Meiners et al. [13], for example, propose major contributions towards generic context prediction. Mayrhofer [12] proposes an unsupervised classification, which tries to find previously unknown classes from input data. Sigg et al. [20] provide an alignment method based on typical pattern and on alignment technique in order to deduce missing low-level context information. Moreover, Meiners et al. [13] present a generic and structured context prediction approach based on (1) the incorporation of the domain knowledge at design time and (2) the selection of multiple exchangeable prediction techniques.

These context prediction approaches try to predict user's next context based on the user's current context and history. However, none of these works consider the services a user invokes on a given context, neither the real need behind it. Against these work, we focus first on the real user's needs and try to anticipate it by predicting his future intentions in similar context. This is in order to propose him the most suitable service that can interest him.

Concerning the second aspect, we have many contributions on recommendation systems, which aim to propose services based on the user's context. Adomavicius & Tuzhilin [1] and Panniello et al. [17] propose to categorize recommendation approaches into three categories: 1) *pre-filtering*, where the contextual information is used to filter out irrelevant ratings before they are used for computing recommendations; (2) *post-filtering*, where the contextual information is used after the standard non-contextual recommendation methods; and (3) *contextual modelling*, where the contextual information is used inside the recommendation algorithms with the user and item data.

For example, Baltrunas and Ricci [2] introduce a technique for context-aware collaborative filtering called "Item Splitting". In this approach, items experienced in two alternative contextual conditions are "split" into two items, which are then used in the rating prediction algorithm. Moreover, Cremonesi et al. [4] propose a technique that relies on classical and post-filters recommendation based on contextual information. This technique use association rules to identify the most significant correlations between context and item. These rules are then used to filter the predictions performed by traditional recommender systems. Recently, Hussein et al. [8] introduces a software framework for the development of context-aware and hybrid recommenders. They introduce the Hybreed framework based on Dynamic Contextualization approach.

All these works try to anticipate user's needs in order to offer him more transparency. Hence, most recommendation systems propose a next service to users based solely on their

context information, without considering the user's requirements behind a service, i.e., its intentions. They propose available services to the user, ignoring why this service is needed.

VI. CONCLUSION

Nowadays, our environment is characterized by the evolution of pervasive technologies. However, PIS that derived from using IS on pervasive environments still complex, requiring an important user's understanding effort.

Therefore, we propose a user-centered vision of PIS based on a context-aware intentional prediction approach, in order to hide PIS complexity. This approach allows us to anticipate the future user's needs. By this approach, we believe contributing to the improvement of PIS transparency and productivity through a user-centered view.

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

The, we implement the service prediction mechanism. This service prediction mechanism is evaluated according to two aspects. The first one is the *scalability,* which allows analysing the feasibility of the proposed mechanism especially with respect to the growing of the service repository. The second one is the *result quality* based on two measures in services field called *precision* and *recall*. These two measures are used to analyse whether our mechanism really reached its goal. The experiments demonstrate that our algorithms can be able to find, in a reasonable time, the most appropriate results that can fulfill user's intention in a given context, with the lowest rate of "false-positive". But, to in order to have a good result, the system designer should describe the different ontologies used in a rich and complete manner and choose the best threshold setting.

This intention prediction mechanism highlights the anticipatory and proactive behavior of our proposed vision of PIS. We strongly believe that an intentional prediction approach can answer to transparency and homogeneity requirements, necessary for fully acceptation of PIS. Moreover, evaluating the user acceptation of the proposal requires applying it in a real case study. Such evaluation should consider the final user's point of view. It should consider the user acceptance, considering the prediction mechanism, as well as the level of transparency perceived by these users. As a future work, we expect to evaluate our approach in a large-scale in order to validate its usefulness and compare it with the existing techniques.

REFERENCES

[1] G. Adomavicius, and A. Tuzhilin, "Context-aware recommender systems," In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, Recommender Systems Handbook, Springer, pp. 217-253, 2011

[2] L. Baltrunas, and F. Ricci, "Experimental evaluation of context-dependent collaborative filtering using item splitting,". User Modeling and User-Adapted Interaction: Special issue on Context-Aware Recommender Systems, 2013.

[3] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery," 2(2), pp. 121–167, 1998

[4] P. Cremonesi, P. Garza, E. Quintarelli and R. Turrin, "Top-N recommendations on Unpopular Items with Contextual Knowledge," In 3rd Workshop on Context-Aware Recommender Systems (CARS) at ACM Recommender Systems, Chicago, USA, 2011

[5] A. Dey, "Understanding and using context," Personal and Ubiquitous Computing, Vol. 5 n°1, pp. 4-7, 2001

[6] W. Feller, "An Introduction to Probability Theory and its Applications," New Jersey: Wiley, 1968

[7] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian Network Classifiers," Machine Learning, 29(2-3), pp. 131- 163, 1997

[8] T. Hussein, T. Linder, W. Gaulke, and J. Ziegler, "Hybreed: A software framework for developing context-aware hybrid recommender systems," User Modeling and User-Adapted Interaction, 24(1-2), pp.121-174, 2014

[9] Jena Semantic Web Framework, http://jena.sourceforge.net/

[10] R.S. Kaabi, and C. Souveyet, "Capturing intentional services with business process maps," 1st IEEE Int Conference on Research Challenges in Information Science (RCIS), pp. 309-318, 2007

[11] M. Kirsch Pinheiro, B. Le Grand, C. Souveyet, and S. Najar, "Espace de Services : Vers une formalisation des Systèmes d'Information Pervasifs," XXXIème Congrès INFORSID 2013: Informatique des Organisations et Systèmes d'Information et de Décision, Paris, 2013

[12] R. Mayrhofer, An Architecture for Context Prediction. PhD thesis, Johannes Kepler University of Linz, 2004

[13] M. Meiners, S. Zaplata, and W. Lamersdorf, "Structured context prediction: A generic approach". In R. Kapitza F. Eliassen (Ed.), Proceedings of the 10th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS), pp. 84–97 IFIP, 6: Springer, 2010

[14] S. Najar, M. Kirsch-Pinheiro, and C. Souveyet, "The influence of context on intentional service," 5th Int. IEEE Workshop on Requirements Engineerings for Services (REFS'11) - IEEE Conference on Computers, Software, and Applications (COMPSAC'11), Munich, Germany, pp. 470 – 475, 2011

[15] S. Najar, M. Kirsch Pinheiro, C. Souveyet, and A. Steffenel, "Service Discovery Mechanism for an Intentional Pervasive Information System," International Conference on Web Services (ICWS), Honolulu : United States, pp. 520-527, 2012

[16] OWLS-TC: http://www.semwebcentral.org/projects/owls-tc/

[17] U. Panniello, A. Tuzhilin, and M. Gorgoglione, "Comparing context-aware recommender systems in terms of accuracy and diversity: Which contextual modeling, pre-filtering and post-filtering methods perform the best," User Modeling and User-Adapted Interaction. Special issue on Context-Aware Recommender Systems, 2013

[18] Pellet, http://www.mindswap.org/2003/pellet/ index.shtml.

[19] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE, 77(2), pp. 257–286, 1989

[20] S. Sigg, S. Haseloff, and K. David, "An Alignment Approach for Context Prediction Tasks in UbiComp Environments," IEEE Pervasive Computing, 9(4), pp. 90-97, 2010

[21] H. Xiao, Y. Zou, J. Ng, and L. Nigul, "An Approach for Context-aware Service Discovery and Recommendation," IEEE Int. Conf on Web Services (ICWS), Miami, pp.163-170, 2010