Prediction intelligence in context-aware applications

Christos Anagnostopoulos, Panagiotis Mpougiouris, Stathes Hadjiefthymiades
Pervasive Computing Research Group,
Department of Informatics and Telecommunications, University of Athens
Panepistimioupolis, Ilisia
Athens 15784, Greece
bleu@di.uoa.gr, daeipkb@mail.daedalus.gr, shadj@di.uoa.gr

Abstract -- **Mobile applications are required to operate in ubiquitous environments of dynamic nature. Specifically, the availability of resources and services may vary significantly during a typical session of system operation. As a consequence, mobile applications need to be capable of adapting to these changes to ensure the best possible level of service to the user. Therefore, such adaptive applications may have pre-evaluated the appropriate knowledge of their environment to act efficiently. Such knowledge is not known a priori, so information prediction and proactivity should enhance and extend the functionality of such applications in order to be adaptable to the future changes of their underlying computational environment. In this paper, we discuss and evaluate such a context prediction algorithm.**

Index Terms – **Context awareness, Context prediction, Context model, Spatial prediction, Context proactivity.**

I. Introduction

In contemporary mobile environments, we encounter enormous computational complexity in contextual information modelling. The retrieval of such information will depend on spatial variants, time, history of interaction, and a range of factors that are not provided explicitly, but do exist implicitly in the ambient environment. Such information is collectively referred to as context. Context awareness allows an *entity* to adapt to its environment, which offers a number of advantages and possibilities for new applications. One of the more intuitive advantages is ease of use; if devices can adapt to their situation, they can engage in more efficient user interaction and proactivity. Other advantages may be less obvious, but context awareness can also lead to reduced energy consumption and, thus, to longer battery life of mobile devices. Another option for improving the usability of such *entities* is to make them proactive: anticipating user action enables a new class of applications to be developed. Moreover, spatial prediction can be used to improve performance for resource reservation in wireless networks and facilitates the possibility of providing desired spatial-oriented services by preparing and feeding them with the appropriate contextual information in advance. Predicted context-aware applications can address context pre-evaluation aspects introducing innovative proactive services, such as alerts related to traffic conditions or occurrences of accidents. Some publications [13,14] already covered the topic of location prediction for different granularities of location information. The concept of predicting the whole context on the level of abstract contextual *identifiers* with on-line algorithms with mechanisms incorporated in the mobile device, as discussed in this paper, is novel. This concept can be seen as a special case of context awareness, which considers the past, present or future context of an entity.

As defined in [2], context aware applications could implement query interfaces that pertain to past, current or future (predicted) knowledge. Prediction allows applications to consume more time in the preparation and presentation of services, especially those involving complex and time-consuming tasks (e.g., mobile e-commerce) and to ensure that only desired services are delivered to end users. Moreover, it is highly significant that context can be pre-evaluated and provided to consumers in a transparent manner. Context can be used effectively to constrain retrieval of information thereby reducing the complexity of the retrieval process. Thus, it is imperative that a context predictive model is introduced. Such a model should be compliant to a generic context, providing a level of abstraction among heterogeneous data models. Several data models for moving objects [3] have been proposed in the literature along with discussions on how to solve issues associated with point location management. In this paper, a profile for a predictive data model, namely a Predictive Context Object (PCO), is proposed. The PCO utilizes specially adapted interpolation formulae with efficient computational complexity. The PCO is implemented focusing on spatial context, which is considered as the basic contextual information in mobile ubiquitous environments.

Our objective is not only to recognize the current context of an entity, but also to predict the future context and, thus, enable proactivity of the entity. Mobile wireless networks are emerging rapidly as a key technology of the information infrastructure. The proactivity of a service is based on the principle of determining the appropriate contextual information ahead of time. By inheriting a predictive context model, the functionality of mobile computing environments is enhanced. The PCO provides a model for spatial information (i.e., Global Positioning System data) and utilizes Cubic Bezier Splines as a value prediction /estimation scheme. Context capturing and prediction should be embedded in information appliances with limited resources. This limits significantly the set of applicable algorithms. It should be pointed out that our model does not perform any kind of probability-based and/or statistical estimation to calculate predictions. On the contrary, it is based on the assumption that certain amount of error will infiltrate to its calculations. Bearing that in mind, our concerns should be in the direction of choosing efficient prediction algorithms [1] and applying them to suitable contexts. In this paper, the proposed PCO is modelled in UML defining semantics of constraints. Focusing on spatial context, several simulation results are presented in order for the PCO to be evaluated against different factors of the prediction algorithm, such as knowledge depth (i.e., history window length of data), predicted future depth (i.e., future window length of prediction). We also introduce an important extension mechanism of this context model, which enables us to predict contextual information that is not available beforehand, by means of contextual inference.

The rest of this article is structured as follows: In section II we briefly depict our concept model (PCO) using the Unified Modelling Language (UML) and represent it through a mathematical formula. In section III we implement the PCO to evaluate the different parameters of the prediction algorithm (knowledge depth, predicted future depth). The results are then presented and analysed. In section IV we refer to an extension mechanism of the PCO and how to express its semantics with a certain ontology language. Finally, conclusions and directions for further work in the area are provided.

II. Predictive Context Object

A. *UML Profile for PCO*

Our concept of context prediction is based on how to define data models with parameters that hold the contextual information. The word "context" is defined as the interrelated conditions in which something exists or occurs in Merriam-Webster's Collegiate Dictionary [4]. While this is a general definition, it does not help much in understanding the concept in a ubiquitous environment. Usage of the word context tends to be rather vague because everything in the world happens in a certain context. We need to focus on the context used by applications in mobile computing. Schilit in [5] divides context into three categories, the *Computing* context (e.g. network connectivity, communication costs, communication bandwidth, nearby resources such as printers, displays), *User* context (e.g. user's profile, preferences, location, even current social situation) and *Physical* context (e.g. lightning, noise levels, traffic conditions, temperature). Time is also an important and natural context for many applications. Since it is hard to fit into any of the above three kinds of context we propose to add a fourth context category as: *Derivable* context, such as every physical (time) or conceptual (activity) parameter whose value may be independent of any context of the aforementioned categories. We can introduce the notion of the *Predictive* context as a special kind of Derivable context, when the Computing, User and Physical contexts are recorded across a certain time interval. Such context specifies the predicted values of every kind of context.

Furthermore, we can define the context more accurately as the set of environmental states that either determines an application's behaviour or in which an application event occurs.

We try to model the Predictive Context Object conforming to a specific profile [15] expressed in UML [6]. We define the classifier *Observable* as the variable of interest (e.g. Longitude, Temperature, and Bandwidth) related to a set of *Entities* (e.g. Human, Terminal). The Observable could belong to any kind of the aforementioned contexts. It implicitly maintains specific knowledge concerning its environment. Every entity is implicitly bound to a set of Observables in order to support context awareness. We introduce the classifier *Ichnos* that describes the current behaviour and status of an entity. The Ichnos classifier is the context model for the related entity. The main topic of this paper focuses on specific prediction mechanisms, which attempt to predict the behaviours of entities depending on past knowledge, as it is maintained into Ichnos. The Ichnos could be a member of the Predictive context. The profile of such prediction mechanisms incorporates a constraint-based classifier, called *Derivable*. The Derivable associates to an Observable and specifies, through prediction algorithms, the Observable's range of values (e.g., The Derivable *time* specifies the future values of the Observable *Longitude*). In Fig. 1 we illustrate a UML meta-model related to Predictive Context Objects for a set of well-defined entities of interest.

In this model, the class Entity is the object whose context should be monitored. An Entity is explicitly bound to a set of Ichne determining the current contextual information. The Entity is semantically interrelated with another Entity through the meta-association *relatesWith*. Such meta-association defines that an entity is related to another entity regarding their current context. Currently two entities are related if there exists a *semantic* function (e.g. a *nearTo* relation expressing the Euclidian distance) between their Ichne classifiers. We designed this data model in order to predict the next value of the Ichnos. The class Observable is a variable that is incorporated in Ichnos and expresses the physical (location information) or conceptual (mobility pattern) behavior of the Ichnos. The value range of the Observable is specified by a Derivable, which in turn, is independent of any Observable contained into a certain Ichnos.

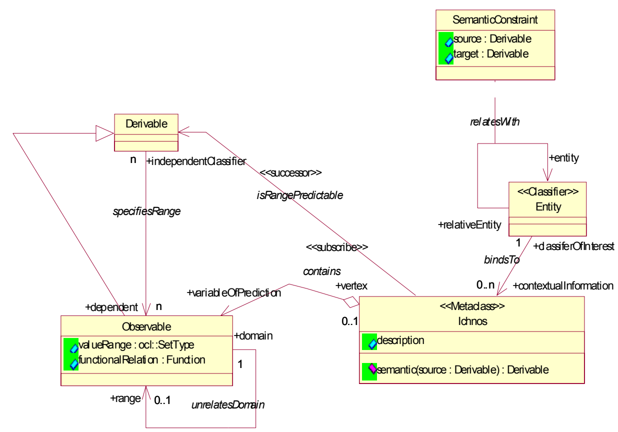

Fig. 1. UML Profile for Predictive Context Object

B. Mathematical Formulation

By representing this model through a mathematical formulation, we accept that the classifier Observable *obs* has as domain of definition any contextual variable and as value range the set $V^{obs} \in \Re^n$. The classifier Derivable *drv* may have as an input field any contextual variable whose value will determine the *obs*'s value range. This means that:

$\exists$ *dependent:* $V^{obs} \times V^{drv} \rightarrow \{true,false\}$,

$\forall$ obs $\in$ Observable $\wedge \forall$ drv $\in$ Derivable

The *dependent* function maps to logic (Boolean) values. This describes that the values of an *obs* depend on the current value of the *drv*. If *dependent* (obs,drv) = true, $\forall$ (obs,drv) $\in \{V^{obs} \times V^{drv}\}$ then the V^{obs} is specified by the V^{drv}. Let us define *prediction*$_{obs}$(drv) as the prediction of the value of the *obs* if the following Boolean statement holds:

$$(dependent(\text{obs},\text{drv}) = \text{true}) \wedge (\exists\, \text{prediction}_{obs}\text{: Derivable} \rightarrow V^{obs}{}_{predicted} \subseteq V^{obs})$$

Applying *prediction*$_{obs}$(drv) to an *obs*, aggregated to an Ichnos and bound to an Entity, we could predict the Entity's future behaviour or contextual state. In order to be able to deduct an Entity's future contextual behaviour, we firstly need to define the notion of Ichnos. Let us define Ichnos as the following set:

$$\begin{aligned}&\text{Ichnos} = \{\text{Observable o} \cup \text{Derivable w} \mid \\ &(\wedge_i\, dependent(\text{i},\text{w}) = \text{true}) \wedge \\ &(\exists\, prediction_i(\text{w}) \in V^{i}{}_{predicted} \subseteq V^{i}) \wedge \\ &(\vee_i\, dependent(\text{i},\text{j}) = \text{false}),\ \forall \text{i,j} \in \text{Observable, i} \neq \text{j} \\ &\wedge \text{ w is } unique\}\end{aligned}$$

In other words, Ichnos is a set of Observables and a Derivable such that there is no dependency between the Observables but there is dependency connecting the Observables to the given, unique Derivable.

A context history Q_e, associated to an entity *e*, is defined as a set of Ichne. Each Ichnos maintains the contextual information of the *e,* for a specific Derivable. Such contextual information is the real (captured) information, not the predicted. If we denote I_{prd} the predicted Ichnos for the entity e, and I_{cpr} the captured Ichnos for the same entity, then the Q_e, is the set $\{(I_{cpr}, \Delta)_p\}$, where p is the history window length (i.e., past context values contributing to the prediction of new) and Δ is an n-dimensional vector, denoting the estimated error for the value of the k^{th} Observable inside the Ichnos, (k=1..#Observables). Q_e is periodically updated with a new Ichnos is added to its container. When we correlate context histories Q_{ent} associated to *ent* entities we might take advantage of the Ichnes' *semantic* function, in order to predict and infer unknown context for the contained context residing into every context history.

III. APPLICATION ON SPATIAL CONTEXT

By modelling predictive contextual information with the PCO concept, we could focus on the context that maintains spatial information (spatial context as User context). Let us define the *Longitude* and *Latitude* instances of the Observable classifier. We consider that such instances are independent with regard to their value ranges (i.e., *dependent*(Longitude, Latitude) = false). Consider the derivable *time* t, which specifies the value range of every instance (*dependent*(Longitude, t) = true). An Ichnos object is constructed by the aforementioned observables. Such an Ichnos maintains the spatial context of an entity. We employ an interpolation algorithm i.e. Cubic Bezier Splines [7], which interpolates time with Longitude and Latitude respectively. Finally, the Ichnos *PlanarLocation* = {{Longitude, Latitude}, {time}} is defined. This maintains the planar context of the bound entity.

We have tried to demonstrate spatial context prediction through simulation. Our domain of interest - the input - will be a GPS trace file. The scenario is quite simple: A user equipped with a GPS-enabled mobile device is driving around a certain neighbourhood of our city for an extensive time period. We collect the trace that his/her device provides - this is how the input file is produced. As the user is moving, his/her new position information is updated every second and we try to predict his/her future location(s) each time. The trace is formulated according to the GPS specifications and comprises of consecutive rows of data that determine the user's position (*PlanarLocation* Predictive Context) on the earth at approximately 1-second intervals.

We chose to implement our prediction mechanism with three different interpolation methods. These are Newton's Divided Differences, Lagrange's Interpolation [7] and Cubic Bezier Splines. Their original domain of application is to try to calculate an estimate of a function's value (e.g. y=f(x)) when only some of its values are available, within a given range of the parameter x. We expand their usage beyond this restrictive range of x. According to our model, y may be related to Observable Longitude and x to the Derivable of y. For instance, when the following pairs of (x,y) are known: $\{(x_1,y_1), (x_2,y_2),...(x_n,y_n)\}$ one can get an estimate of $y_i=f(x_i)$, $x_1<x_i<x_n$, using interpolation methods, without knowing the function f. We will utilize such pairs in order to get estimates for $x_i>x_n$. This actually means that we will try to predict the user's future location based on his/her past locations.

The first two methods (Newton's and Lagrange's) proved inappropriate for this purpose, as they induce significant errors, due to their inherent oscillatory behaviour, when the amount of (x,y) pairs increases. This was confirmed by our simulation results. On the other hand, Cubic Bezier Splines proved quite promising, as will be demonstrated below, while being extremely efficient in terms of complexity – O(n), where n is the length of the history window.

There were two major issues that our simulation tried to investigate. The first was the amount of knowledge (context history) that should be used by the system so as to be able to make reliable estimations. That is the actual number of (x,y) pairs used every time by the method to yield a prediction. Let us call it Q_u (context history for user entity). Should Q_u be large? In other words, should our system "remember" many past user locations before trying to predict the future ones? The question concerning Q_u is not primarily whether big Q_u's would require extremely large execution time, but also how it affects the prediction's error.

The second one was the capability of our system to predict deeply into the future. Given a set of past locations, how many seconds in the future could it provide predictions that bear acceptable errors? If we know where the user was at t_1, t_2, t_3, t_4, is it possible to estimate where they will be at $t = t_{current}+K$ with $K>1$?

The question now becomes "what is the most efficient combination of Q_u (knowledge of the past) and K (future depth) that gives the least errors?". For this, we progressed our simulation by investigating the performance of the algorithm for large value ranges of Q_u and K. The algorithm used is illustrated, in pseudocode, in Fig. 2 and appears graphically in Fig. 3.

```
counter = 0;
while(user_is_moving) {
   get_current_location;
   if (counter < Q) {
      wait_for_more_user_locations;
   } else if (counter < Q+K) {
      calculate_Kth_user_location_from_now;
   } else {
      calculate_error_in_current_prediction;
      calculate_Kth_user_location_from_now;
   }
   counter = counter + 1;
}
```

Fig. 2. Algorithm for predicting the K^{th} Ichnos for history window length Q

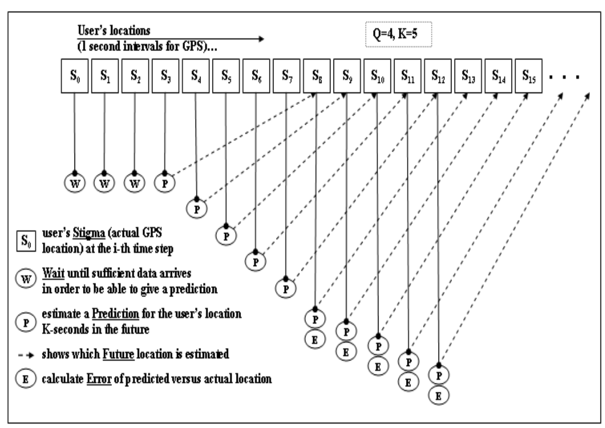

Fig. 3. Graphical representation of the Q+K prediction

The results of the simulation are presented in Fig. 4. In the x-axis one can see the values of Q_u length. The y-axis represents the Mean Error Percentage (MEP) of the user's future location estimations for the first 4 values of K. As MEP we define the mean value of the per cent ratio $(x_{pr}-x_{rl})/x_{pr}$ where x_{pr} is the predicted and x_{rl} is the real (captured) value of the x variable. The observed error increases as one tries to predict deeper into the future. It is notable, though, that the MEP of the estimations lies far beneath 1% for values of K=1, 2, 3, 4. This means that Cubic Splines enabled us to predict the user's future spatial context with relatively good accuracy.

What is also remarkable is the fact that the MEP does not decline as Q_u length grows. This means that no matter how large our knowledge base was (the user's past locations that we take under consideration for the prediction), the MEP remained almost the same! Consequently, a quite short Q_u would be adequate for fast and reliable prediction attempts. Therefore, the value of the Q_u and the specific depth K could be introduced to the *PlanarLocation* Ichnos extending its definition with the *dependent*(Q_u,K) = true. This means that Q_u produces the least MEP for certain range of K. For constant Q_u length, the MEP value is increasing as the K value increases. This is the Computing context, which feeds the Predictive context with the appropriate information in order to predict the user's location.

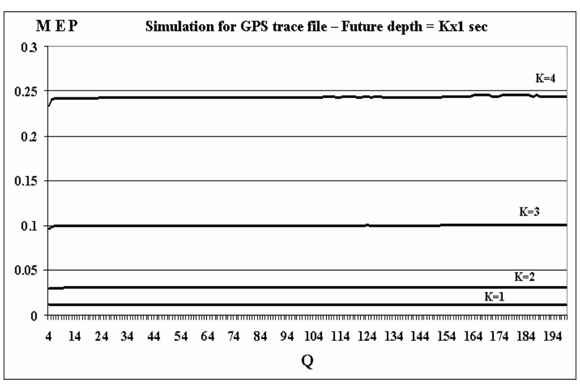

Fig. 4. Mean error percentage related to Q for certain K values (simulation for GPS)

For reasons of generality and diversity in our study of the PCO we also performed our simulation with another type of location Ichne. The source came from IBM's City Simulator [16]. The results of the simulation can be seen in Fig. 5.

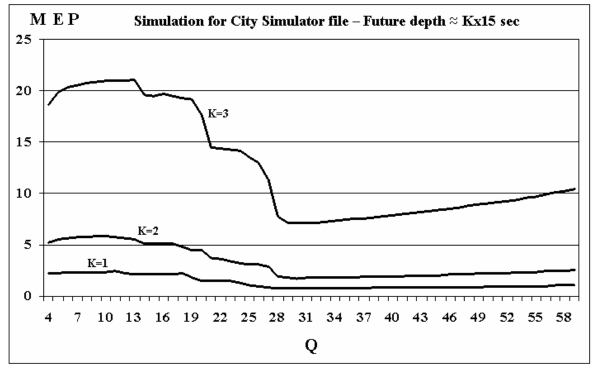

Fig. 5. Mean error percentage related to Q for certain K values (simulation for City Simulator)

It is noticeable that in the case of the City Simulator the simulation produced errors varying from 2% to 5% for values of K=1, 2. However, such values of K refer to predictions 15 seconds and 30 seconds into the future respectively. For predictions of up to 30 seconds into the future, the MEP remained below 10%. This is rather an important inference regarding a city plan. We assume that the *dependent*(Longitude,Latitude)=false, which means that the value of the Longitude is independent with the value of the Latitude. If *dependent*(Longitude,Latitude)=true, which means that the value of the Longitude is dependent with the value of the Latitude and we meet such relation in a city plan, then we could use a 3D Bezier Splines and the Ichnos classifier has to be extended in order to support such dependency.

At this point one should stress the contrast of using mathematical methods, such as Cubic Splines, in order to implement predictive algorithms, instead of statistical and/or probabilistic methods like *ProbInitial* and *ProbUpdate* algorithm in [14] and *Path Prediction Algorithm* in [13]. The errors introduced by mathematical methods can be analysed and estimated through exhaustive simulations for various data and parameters, thus generating concrete knowledge of their behaviour and how they could be exploited.

On the contrary, the statistical probabilistic methods introduce errors that render such methods useful only under certain circumstances. One has the option to decide what is more functional for their intended use: the estimation that a certain context parameter might have Value_1, for example, with a% probability or the knowledge that this parameter has Value_1 ± e% error. We propose the PCO in combination with a mathematical prediction algorithm (Cubic Splines) so as to facilitate context aware applications with a reliable predictive model.

IV. Extension Mechanism

A. Dependency Inference Operator over Ichnos

Using this model we were capable of retrieving contextual information from other already predicted contextual information. An extension mechanism of this model is based on how to exploit predicted knowledge in order to determine the Observable's value. We can define operators between Ichne over their *semantic* functions. We define the operator *Dependency Inference* (DI) between two Ichne I_1 and I_2 related to specific derivable I_1.drv and I_2.drv respectively, as a new Ichnos for which the following statements and constraints hold:

{Ichnos I | *dependent* (I_2.obs, I_1.drv) = true
$\wedge$ (I.obs = I_2.obs)
$\wedge$ (I.drv = I_1.drv)}

The I_1 and I_2 Ichne are not necessarily bound to the same Entity, whilst the I_2.drv is semantically interrelated with I_1.obs. We call the following function a *Dependency Inference* (*DI*),

DI: Ichnos $\times$ Ichnos $\rightarrow$ Ichnos

To make this mapping feasible I_1.obs and I_2.drv must be of the same type. This constraint, expressed in OCL [8], is listed below:

```
context Entity inv:
self.semanticConstraint[relativeEntity].
target.oclIsKindOf(self.bindsTo.semantic
(self.semanticConstraint[entity].source).type)
```

Let us envisage that we are interested in the user's planar location and the terminal's available bandwidth. The planar location and the bandwidth are instances of Ichnos. We can predict the future planar location value (i.e., the future longitude and latitude of the user). The Ichnos *LocBand* = {{Bandwidth}, {Longitude}} describes the contextual information of the terminal's available Bandwidth regarding its Longitude value (Physical context for the terminal). So, the *dependent* (Bandwidth, Longitude) = true. Let us suppose that there exists a *semantic* function between the user and the terminal entity (e.g. user *nearTo* terminal), regarding their Observable Longitude. Then, we could predict the terminal's Bandwidth through the user's Longitude if we consider that terminal.Bandwidth's Derivable (in this case the Longitude) is *semantic*(user.Longitude), as the result of the *semantic* function (e.g. Euclidian distance between user and terminal is above a specified threshold that means that user is indeed *nearTo* terminal). The Derivable of the user's Longitude is time. Now, we can infer that the Derivable of the Bandwidth is time. This means that:
dependent(Bandwidth,Longitude)=
dependent(Bandwidth,*semantic*(Longitude))=
dependent(Bandwidth,time) = true

We can predict the future value of the Bandwidth from the time because of the interrelated behavior of Bandwidth's entity (terminal) with the Longitude's entity (user) and the fact that LocBand's Derivable is PlaneLocation's Observable. The LocBand context is then a Predictive Context because one can exploit it in order to predict the values of its Observables. Using the *DI* function, we do not only achieve new predictive context but we can also use it in order to transform and merge the context from one kind to another. In this case, the user's context is merged to the terminal's physical context for prediction purposes. This merging technique is employed if and only if there exists a semantic association between the involved entities. Fig. 7 illustrates three spaces: the user's horizontal movement, the terminal's available bandwidth related to its horizontal location, and the *nearTo* function related to the linear distance D of the user and the terminal. When the user's context history Q_u intersects with the terminal's context history Q_t over the Observable Longitude and their Euclidean distance $D \leq D_{threshold}$ then the DI operator is applied in order to predict the terminal's available bandwidth for certain K time depth. $D_{threshold}$ indicates the piece of contextual information that the user *relates with* the specific terminal (user is *near to* this terminal). If D is systematically higher to the $D_{threshold}$ then it could be deduced that the user uses this terminal and the user's movement determines the terminal's available bandwidth.

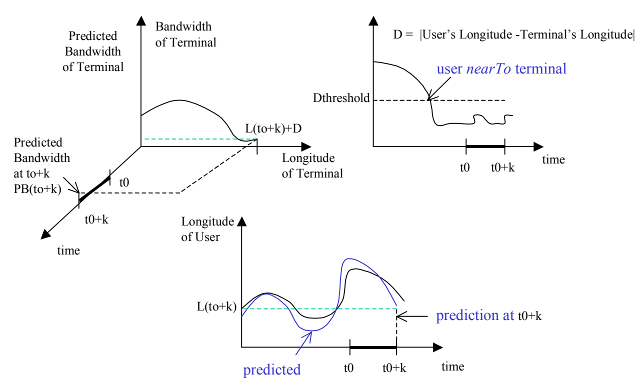

Fig. 7 .The use of DI operator nearTo for further prediction

B. Ontological expression of Ichnos

We can model and express the semantic function of the Ichnos as an instance of the meta-association *relatesWith.* Such an instance could be expressed in a Semantic Web Language, such as OWL [9], which resolves interoperability issues among heterogeneous ubiquitous systems. According to our data model we can apply the DI operator over the *semantic* function to set of Ichne, contained into a Q_e. Applying such an operator will enrich the context awareness in terms of inference, by capturing or predicting contextual information previously unspecified. Fig. 8 depicts an OWL graph with the relationships among Ichne and Entities for the aforementioned example.

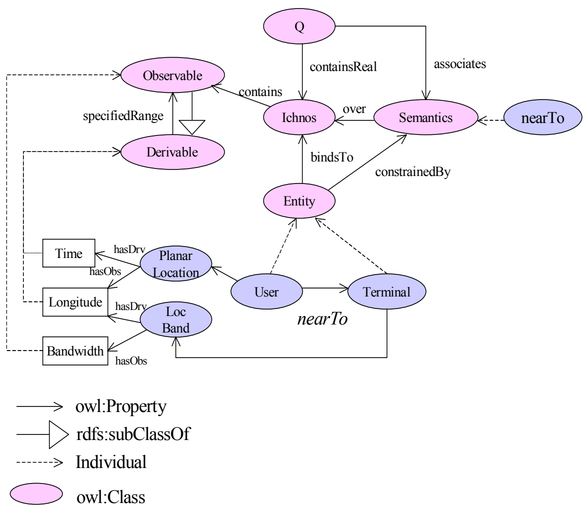

Fig. 8. Ontology representation of the PCO

V. FUTURE WORK

In this paper, we proposed a context prediction algorithm related to interpolation mechanism. We also defined a UML Profile for modelling Predictive Context Objects that generate the notion of the Predictive context. Such context includes any information associated with any variant whose value we want to predict. This variant, called Observable is time-dependent. The Predictive context may be merged into well-defined contexts (Physical, User, and Computing). Through the combination of several context values, we may generate a more powerful understanding of the current situation of an entity. Current contexts (capturing information up to now) act as indices into other sources of contextual information. By integrating such information with its Predictive context, we can implement proactive procedures for an application. For example, by knowing the current location and current time, together with the user's calendar, and by estimating the future location, the application will predict of the user's future social situation, such as having a meeting, waiting in the airport, and so on. The PCO model is defined as a "ground" model that may build more complex type of Predictive context variants by using the notion of a semantic function as specified into Ichnos classifier. The total error, e_{total}, of the spatial context prediction may be considered as a complex Ichnos. Such an Ichnos relates to the error on longitude, e_{long}, and the error on latitude, $e_{lati,}$ and defines a *semantic* function between such errors as the vector $e_{total} = [e_{long}, e_{lati}]$. Our future work is based on how to introduce more semantics and merging techniques into Ichnos classifier for modelling complex data types that represent the Predictive context. Moreover, we intend to improve the prediction algorithms that we have used so far (namely Cubic Splines) bearing in mind facts like user direction and velocity vectors for spatial context.

REFERENCES

[1] Jan Petzold, Faruk Bagei, Woflgang Trumler, and Theo Ungerer, "*Context Prediction Based on Branch Prediction Methods,"* Institut fur Informatik, D-86135, Universitat Augsburg, Report 2003-14, 2003

[2] Ouri Wolfson, "*Moving Objects Information Management:The Database Challenge*", Department of Computer Science, University of Illinois at Chicago,2002

[3] Goce Trajcevski, Ouri Wolfson, Cao Hu, Hai Lin, Fengli Zhang, Naphtali Rishe, "*Managing Uncertain Trajectories of Moving Objects with Domino*", Department of Computer Science, University of Illinois at Chicago,2002

[4] Merriam-Webster's Collegiate Dictionary, [www.m-w.com].

[5] Bill N. Schilit, Norman Adams, Roy Want, "*Context-Aware Computing Applications*",1994

[6] Unified Modeling Language,[www.uml.org].

[7] Steven C. Chapra, Raymond P. Canale "*Numerical Methods for Engineers (fourth edition)*",McGraw-Hill,2002

[8] Object Constraint Language,OCL, [www.omg.org/cgi-bin/doc?ptc/2003-10-14].

[9] Ontology Web Language,OWL,[www.w3.org/ 2004/OWL].

[10] Amiya Bhattacharya, Sajal K.Das, "*LeZi-Update:An Information-Theoretic Approach to Track Mobile Users in PCS Networks*", Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom99,1999

[11] Jan Petzold, Faruk Bagci, Wolfgang Trumbler, Theo Ungerer, "*Global and Local State Context Prediction*", University of Augsburg, Institute of Computer Science,2003

[12] Rene Mayrhofer, "*An Architecture for Context Prediction*", Institut fuer Pervasive Computing, Johannes Kepler Universitaet Linz,2004

[13] Miltiadis Kyriakakos, Nikolaos Frangiadakis, Stathes Hadjiefthymiades,Lazaros Merakos, "*Enhanced Path Prediction For Network Resource Management In Wireless LANs*", IEEE Wireless Communications, December 2003.

[14] Hassan A. Karimi and Xiog Liu,"*A Predictive Location Model for Location-Based Services*", ACM-GIS'03,NewOrleans,Louisianna,USA, 2003

[15] UML Profile for Enterprise Distributed Object Computing [www.omg.org/technology/ documents/formal/edoc.htm].

[16] J. Kaufman, J. Myllymaki, J. Jackson, "*City Simulator V2.0*", IBM alhaWorks, December 2001.