UNISINOS

Programa Interdisciplinar de Pós-Graduação em

Computação Aplicada

Mestrado Acadêmico

João Henrique da Rosa

ORACON: An Adaptive Model for Contexts Prediction

São Leopoldo, 2013

UNIVERSIDADE DO VALE DO RIO DOS SINOS — UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA
NÍVEL MESTRADO

JOÃO HENRIQUE DA ROSA

ORACON: AN ADAPTIVE MODEL FOR CONTEXTS PREDICTION

SÃO LEOPOLDO
2013

João Henrique da Rosa

ORACON: AN ADAPTIVE MODEL FOR CONTEXTS PREDICTION

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos — UNISINOS

Advisor:
Prof. Dr. Jorge Luis Victória Barbosa

São Leopoldo
2013

Ficha catalográfica

R788o Rosa, João Henrique da
Oracon : an adaptive model for contexts prediction / por João Henrique da Rosa. – 2013.
103 f. : il., 30cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, 2013.
Advisor: Prof. Dr. Jorge Victória Barbosa

1. Context-awareness. 2. Contexts prediction. 3. Prediction algorithms. I. Título.

CDU 004.421

Catalogação na Fonte:
Bibliotecária Vanessa Borges Nunes - CRB 10/1556

ACKNOWLEDGEMENTS

- First of all I thank this strange and unknown strength of the universe, called God by many, which gave me the opportunity of being alive and having the chance of meeting such great people and having such unbelievable experiences.
- I thank my girlfriend who has always been making everything in my life brighter.
- I thank all my family, been included my mother, father, and siblings-in-law, for all the love, appreciation, and support.
- I thank my friends for all great times together.
- I thank Dr. Jorge Barbosa who has advised me for the last four years.
- I thank Dr. Cristiano Costa who has always showed himself opened for helping and teaching.
- I thank Dr. Claudio Geyer for participating of the evaluation board of my thesis.
- I thank Eduardo Reis for his commitment and dedication.
- I thank my friend Rodrigo Remor who assisted me in many times of difficulties.
- I thank the National Counsel of Technological and Scientific Development (CNPq) for funding this research.
- I thank everyone who directly and indirectly contributed for the conclusion of this work. It was and will always be a really great pleasure to work with such dedicate and amazing people.

“*Nothing in life has any meaning, except the meaning you give to it.*”
“*Get excited about your problems. Don’t hope to get to a place where you have no problems. Create higher quality problems!*”
“*A decision made from fear is always the wrong decision.*”
“*Happily achieve instead of achieving to be happy.*”
Anthony Robbins

ABSTRACT

Contexts prediction has been receiving considerable attention in the last years. Furthermore, this area seems to be the next logical step in context-aware computing, which, until a few years ago, had been concerned more with the present and the past temporal dimensions. There are many works regarding models for contexts prediction. Nevertheless, most of them employ the same algorithm for all cases. In other words, we did not find any approach that automatically decides the best prediction method according to the situation. Therefore, we propose the ORACON model. ORACON adapts itself in order to apply the best algorithm to the case. Moreover, the model supports other important aspects of ubicomp, such as, context formal representation and privacy. In this thesis, we describe the ORACON design and evaluate the model through two experiments, one using real data and the other employing simulated information.

Keywords: Context-awareness. Contexts prediction. Prediction algorithms.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

The evolution of mobile devices and high-speed wireless networks has been stimulating researches related to Mobile Computing (DIAZ; MERINO; RIVAS, 2010; SATYANARAYANAN et al., 2009). In this area, the improvement and proliferation of Location Systems (HIGHTOWER; BORRIELLO, 2001; HIGHTOWER; LAMARCA; SMITH, 2006) have motivated the adoption of solutions that consider the user's precise location in the providing of services (Location-Based Services (VAUGHAN-NICHOLS, 2009; DEY A. K. HIGHTOWER J., 2010). However, lately, mobile applications have also become to take into account the user's current context to distribute content and services (Context Awareness (BALDAUF M., 2007; BARBOSA J. L. V., 2007; HOAREAU; SATOH, 2009; LEWIS et al., 2010)). According to DEY; ABOWD; SALBER (2001) context is: "any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves."

Context-awareness is a very relevant aspect for ubiquitous computing (or ubicomp for short) (COSTA; YAMIN; GEYER, 2008). However, ubicomp is a broader research area, which also concerns other issues, such as, heterogeneity, scalability, privacy and trust, mobility, invisibility, transparent user interaction, among others. The model presented in this work regards context-aware computing. Nonetheless, as this area is an important pillar for ubiquitous computing, the model consequently empowers the implementation of ubicomp. Context-aware computing enables applications taking decisions based on users' data (for example, their profiles) and their present contexts (HOAREAU; SATOH, 2009; LEE; PARK; LEE, 2009).

However, DEY; ABOWD; SALBER (2001) also briefly described the importance of using contexts history in the decision-making process. Moreover, in previous works, we studied the value of considering users' past actions performed in the contexts visited during a period, such as, the activities did, the applications used, the contents accessed, and any other possible data (SILVA et al., 2010). This information helped to improve the distribution of content and services in context-aware environments, because applications were using an additional and more complete information source. In other words, applications passed to use the contexts history in conjunction with the current context information and users' profiles to take decisions (SILVA et al., 2010). Many other authors also approached the use of contexts histories in the decision making process (KALATZIS et al., 2008; HONG et al., 2009; CIARAMELLA et al., 2010; MANTORO; MUATAZ; AYU, 2010; BAUR et al., 2010; MANIKANDAN et al., 2011).

With the use of the users' current contexts (**present**) and their contexts histories (**past**), applications already have a reasonable information source to base their decisions. Nonetheless, in order to become proactive and act before the context has actually changed, future contexts have to be predicted (KONIG et al., 2011). This have motivated researchers to study the use of another temporal aspect; the **future** (MAYRHOFER, 2005; SIGG et al., 2011; VOIGTMANN;

LAU; DAVID, 2011; FOLL; HERRMANN; ROTHERMEL, 2011). The obtainment of the users' future contexts is made through predictions techniques. Based on users' histories and their current contexts, algorithms predict the contexts that probably will describe the users' future situations (SIGG; HASELOFF; DAVID, 2010).

1.1 Motivation

The essential part of a prediction model is the algorithm used in the predictions. There are many methods for this task. However, there is no single approach best suitable for all cases. In his doctoral dissertation, SIGG (2008) discussed the requirements for contexts prediction and analyzed various algorithms. The most important attributes identified were high prediction accuracy and high prediction horizon. The prediction horizon is the number of contexts ahead. Other important aspects described were: processing complexity, memory consumption, and applicability for numerical and non-numerical contexts.

According to these requisites, the most suitable algorithms were: Alignment and Markov (SIGG, 2008). In cases where the prediction search space was short and the demanded prediction horizon was high, the Alignment method outperformed Markov. The search space is the entity's contexts history, which is used in the prediction process. On the other hand, for low prediction horizons and high search spaces, the Markov approach bet Alignment.

Later on, KONIG et al. (2011) enhanced the Alignment approach using correlation among different context sources. KONIG et al. (2011) analytically proved that the enhanced Alignment has equal or better accuracy then the standard Alignment. Nevertheless, both the standard and the enhanced versions have a limitation, which was pointed out by VOIGTMANN; LAU; DAVID (2011). That disadvantage occurs when entities have a completely new contexts sequence, describing their current actions.

The reason for this shortcoming is due to the way the algorithm works. It takes the last contexts sequence of an entity (e.g. the last five contexts) and tries to find in its contexts history that same sequence or the best approximation for it. After the best approximation is found, the contexts following it are used as prediction. Thus, when the entity has a completely new contexts sequence, the methods do not find a good approximation in the history.

Aiming to solve that problem, VOIGTMANN; LAU; DAVID (2011) proposed the Collaboration approach. This method uses the correlated contexts histories of many entities to make prediction for one of them. Thus, in cases that the entity has a completely new contexts sequence, the Collaboration algorithm searches for approximations in the correlated entities' histories. However, this approach also has a drawback. It has to find the exactly match of the last contexts sequence in the entities' histories. In other words, the Collaboration method does not use an approximation technique as Alignment does. The own authors recognize the importance of adding this feature, which they call fuzziness, to their approach (VOIGTMANN; LAU; DAVID, 2011).

As we can see, there are many algorithms for contexts prediction. However, there is no single method that is the best for all cases. On the contrary, as the prediction scenario changes, the most suitable algorithm also varies. In literature, there are many works regarding models for prediction. Nevertheless, most of them employ the same algorithm for all situations. In other words, we did not find any approach that automatically decides the best method according to the case.

FOLL; HERRMANN; ROTHERMEL (2011), for example, proposed a model for prediction that uses Semi-Markov Chains (BAIER; KATOEN, 2008) to represent all users' contexts, which is then used to make predictions. MEINERS; ZAPLATA; LAMERSDORF (2010) implemented a framework that enables the employment of the best algorithm to the case. However, it requires that the developer specifies it at design time. SIGG (2008) designed a prediction architecture that uses the alignment method to make predictions for all situations.

Thus, as there is no single algorithm best suitable for all cases and we did not find any model that can automatically choose the best method, we propose ORACON. ORACON automatically decides the most appropriate approach according to the situation. Moreover, ORACON supports other important aspects of ubicomp that are not considered by the related works, such as, context formal representation and privacy.

For the information specification feature, we consider that ubiquitous scenarios are highly dynamic, that is, applications can interact with a great number of different and unknown applications all the time (DEY; ABOWD; SALBER, 2001). Hence, it is fundamental to define a context representation, so that different systems can communicate. However, we did not find any attempt to cover this demand in the analyzed prediction models.

The privacy aspect is approached in two ways. The first discusses which applications and entities will have access to the predictions made for a specific entity. In a ubiquitous environment, users interact with a great number of applications and other users all the time (COSTA; YAMIN; GEYER, 2008). Thus, it is reasonable to consider that programs will want to use predictions made for other applications or entities. Nonetheless, we did not find this study in any of the researched works.

The second aspect of privacy is mainly concerned with algorithms that use multiple entities' histories to make prediction for a single one. Many entities do not wish to share their entire histories. Some of them may want to divide only a certain amount of their data or nothing at all. Therefore, it is important to consider how to control privacy related to this characteristic.

1.2 Problems and questions

Below we list the aspects that were not fully explored in the studied related works and that ORACON aims to address:

1. How to adopt an **adaptive strategy** in order to apply the most suitable algorithm according to the case?

2. Which **context formal representation** to use, so that different applications can communicate with the model?

3. How to control **privacy**, so that algorithms, such as Collaboration, can be applied, and entities can have access to other entities' predictions?

1.3 Objectives

The general objective of this thesis is to propose, implement, and evaluate an adaptive model for contexts prediction. The main contribution of the work regards the capability of choosing the best method according to the situation. However, the model also aims to support a formal representation of context and a privacy mechanism that enables entities to choose which of their data or prediction can be shared. The specific objectives are listed below:

- To design a model;
- To describe experiments to evaluate the model and to show its functionalities;
- To implement a prototype;
- To assess the model by using the prototype in the detailed experiments.

1.4 Methodology

Initially, we researched the context-aware computing area, mainly works related to the use of the users' current contexts (**present**) as well as the use of their contexts histories (**past**) to take decisions. In a second moment, we identified gaps in the studied researchers. The initial idea for this proposal was to design a management model for storing contexts histories. However, we found many advanced works approaching this subject. Therefore, we decided to move the efforts to the study of the **future** dimension of context-awareness.

Although there are also many works regarding this theme, we found some clear possibilities of contribution. Thus, the searches for related works were intensified and the researches questions and problems were defined. In the sequence, we studied forms of solving the identified problems and the adaptive model for contexts prediction (named ORACON) was design. Thus, the following steps of this research were: to describe evaluation scenarios, to implement a prototype, and to test the model.

1.5 Outline

This thesis is organized into eight chapters. The second presents background concepts. Chapter three compares related works and describes the relevant aspects for contexts prediction models. The fourth chapter discusses the context formal representation used in ORACON. The fifth chapter details the contexts predictions algorithms supported by the model. The sixth chapter discusses the ORACON model. The seventh deals with the evaluation experiments. And finally the eighth chapter presents final considerations and future works.

2 BACKGROUND AND BASIC CONCEPTS

In this chapter, we approach some basic concepts related to our work. The chapter is divided into five sections. The first provides an overview about Ubiquitous Computing and Context Awareness. The second section presents works related to Context Modeling and Ubiquitous User Modeling. In addition, it describes how those two areas are connected. The third section approaches Contexts History. The fourth section presents concepts related to Contexts Time Series. Finally, the fifth section describes the contexts predictions concepts.

2.1 Ubiquitous Computing and Context Awareness

According to Weiser, the definition of a ubiquitous computing system (or *ubicomp* for short) is based on two fundamental aspects, which are: ubiquity and transparency (WEISER, 1993). Ubiquity implies that the interaction with the system is available wherever the user needs it. Transparency expresses that the application is non-intrusive and is integrated into the everyday environment. Similarly to Weiser's definition, SALBER; DEY; ABOWD (1998) described two concepts that supply a clear boundary for ubicomp and express its relationship with other research areas, such as, mobile computing, augmented reality, and wearable computing. Those two dimensions are: mobility and interface transparency. Mobility is the degree of freedom of the user to move around, when interacting with the application. And interface transparency is related to the system's interface and denotes the conscious effort required either for operating the system or for understanding its output.

Many works in ubicomp lead toward the development of interactive environments, which enable the mobility of both users and devices. The ubicomp concept, as described by COEN et al. (1999), is closely related to intelligent environments enriched by computers embedded in everyday objects, such as, blackboards, tables, chairs, which are enriched by sensors capable of obtaining context data. According to DEY; ABOWD; SALBER (2001) context is: "any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves."

Some applications use context information to take decisions, such as, to adapt the format of learning contents according to the user's device and bandwidth or to discover business opportunities among the people within the same shopping mall. The use of the entity's context to take decision is defined as Context Awareness (LEWIS et al., 2010). Context-aware applications have to be able to operate in highly dynamic environments and placing minimal demands on user attention. They aim to meet those requirements by adapting to selected aspects of the context, such as, the current location, time and user activities (HOAREAU; SATOH, 2009).

2.2 Context Modeling and Ubiquitous User Modeling

There are many works concerned on modeling context information (BETTINI et al., 2010). In 2001, DEY; ABOWD; SALBER (2001) provided a classical categorization for context data. They argued that context-aware applications look at the who's, where's, when's and what's (that is, what the user is doing) of entities and use this information to determine why the situation is occurring. Hence, the authors proposed four basic categories to model context, which are: (1) identity; (2) location; (3) time; and (4) activity. These context types not only answer the questions of who, where, when, and what, but also act as indices into other sources of contextual information.

For instance, given a person's identity, an application can acquire many pieces of related information, such as, phone numbers, addresses, email addresses, birth date, list of friends, and relationships to other people in the environment. Considering the entity's location, it is possible to determine what other objects or people are near the entity and what activity is occurring in her surroundings. Therefore, DEY; ABOWD; SALBER (2001) concludes that the primary pieces of context for one entity can be used as indices to find secondary context (e.g., the email address) for that same entity as well as primary context for other related entities (e.g., other people in the same location). In addition, they explain that the secondary pieces of context share a common characteristic. They can be indexed by primary context because they are attributes of the entity with primary context.

Nonetheless, along the years, other context modeling approaches were proposed. An example is the Composite Capabilities/Preference Profile (CC/PP) model (KLYNE et al., 2005), which is a markup-based W3C standard for description of mobile devices. Other example is the Context Modeling Language (CML), which is a tool to assist designers with the task of exploring and specifying the context requirements of a context-aware application (HOAREAU; SATOH, 2009). CML provides modeling constructs for describing types of information, their classifications, relevant quality metadata, and dependencies amongst different types of information.

In a similar way, HECKMANN (2005) described the *ubiquitous user modeling* concept. The author explained that the user's behavior is constantly tracked at any time, at any location and in any interaction context. Moreover, the various user models are shared, merged and integrated on demand. The author defined the term as follows:

"Ubiquitous user modeling describes ongoing modeling and exploitation of user behavior with a variety of systems that share their user models." (HECKMANN, 2005)

This description is closely related to the context modeling ideas. In fact, in studies conducted during the research, we noticed that the ubiquitous user modeling notion includes the concepts of context modeling and goes further, approaching additional issues, such as, challenges of scalability, scrutability, privacy, decentralization, communication, and integration (HECKMANN, 2005). According to Viviani; Bennani; Egyed-Zsigmond (2010), user modeling plays a fundamental role in context-aware environments and it represents the basis for cross-system personalization and interaction. The shared user models can either be used for mutual or for individual adaptation purposes (CARMAGNOLA; CENA, 2009). The objective is to enable isolated user modeling applications to exchange partial user models with each other. Thus, the challenge is the semantic integration of the distributed heterogeneous partial user models to enable long term user modeling.

There are many works approaching ubiquitous user modeling. HECKMANN; KRUEGER (2003), for example, introduced the idea of using sharable data structures containing user features and preferences, aiming to enable personalized interactions of users with different devices. They proposed an XML-based user modeling mark-up language (named UserML) as a platform for communication in ubiquitous environments. In this approach, the authors considered aspects, such as, privacy and the right of every human for introspection and control of their collected data. Furthermore, other important characteristic of UserML is the decupling from semantics. UserML specifies the syntax of information, whereas external ontologies describe the semantic.

NIEDEREE et al. (2004) introduced the Unified User Context Model (UUCM), which is a centralized and extensible multi-dimensional user model for aggregating the partial user models collected by individual personalization systems. The personalization systems have to build upon their user models the UUCM structure. Later on, the same authors suggested the use of ontologies for the standardization of user models and for easing information exchange between applications (MEHTA et al., 2005).

In this context, HECKMANN (2005) proposed an extensive approach for ontology-based representation of user models by introducing the General User Modeling Ontology (GUMO). Viviani; Bennani; Egyed-Zsigmond (2010) mentioned that: "GUMO seems to be the most comprehensive user modeling ontology proposed". The author also developed a new architecture employing UserML and GUMO to address the problem of the uniform interpretation of decentralized user models. In addition, HECKMANN (2005) proposed the UbisWorld ontology to describe the user's context. UbisWorld extended the Blocks World (SLANEY; THIéBAUX, 2001) and the context toolkit of DEY; ABOWD; SALBER (2001) to the special needs of contextualized interaction in ubicomp environments with user modeling and privacy.

As we presented in Section 1.2, one of the problems identified in the studied works on contexts prediction was the absence of a specification for context representation, so that different applications can communicate with the prediction models. Therefore, ORACON aims to fulfill that gap. For this reason, we studied the Context Modeling and Ubiquitous User Modeling areas, aiming to identify the works that could be used to specify the context information in ORACON. In Chapter 4 we detail the chosen context formal representation for the model, which are the languages and the ontologies of HECKMANN (2005), due to the ontological approach and the decoupling from syntax and semantic.

2.3 Contexts History

Context-aware architectures that use not only present contexts, but also measurements about the past, need to store observed contexts for further use. The concept that implements this idea is contexts history. In literature, there are many works that approach contexts history (SILVA et al., 2010; DRIVER; CLARKE, 2008; ASHLEY, 2008). Moreover, some proposals refer to the history as Trail (SILVA et al., 2010; DRIVER; CLARKE, 2008). The first work found that employed the term trail to represent a history was BUSH; WANG (1945). He envisioned a machine, called Memex, which would record all users experiences. Memex would operate as the human mind, using associations. When the mind has one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in accordance with some intricate web of trails carried by the cells of the brain. Therefore, Memex would enable selection by association, rather than by indexing (BUSH; WANG, 1945).

There are works on history focusing, specifically, on life logging (DOHERTY et al., 2011; SELLEN; WHITTAKER, 2010; BELIMPASAKIS; ROIMELA; YOU, 2009). Their main purpose is to enhance human memory by using the capabilities of computers. They can record chat conversations, documents, location information, photographic, audio, e-mail and video content using cameras and microphones, and many other types of personal and environmental data (GYORBIRO; FABIAN; HOMANYI, 2009). Some life logging systems aid users to remember past events, providing different forms of visualization and access to the recorded contents (GEMMELL et al., 2002). For example, there are works that focus, specifically, on helping people with episodic memory impairment (LEE; DEY, 2008).

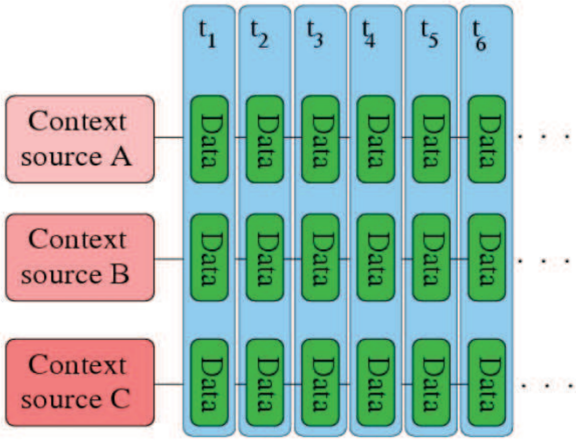
In contrast, there are works that approach contexts history focused on assisting systems to personalize services and contents according to the users' previous choices (HONG et al., 2009; MANIKANDAN et al., 2011; MANTORO; MUATAZ; AYU, 2010; CIARAMELLA et al., 2010). They usually have a well defined domain representation through an ontology (HONG et al., 2009; SILVA et al., 2010) or a entity-relationship model (MANIKANDAN et al., 2011). The domain definition facilitates queries and reasoning to discover users' preferences based on their past actions (SONG et al., 2010).

2.4 Contexts Time Series

Contexts histories are also referred to as contexts time series (SIGG, 2008). In fact, a time series can represent the entire history or only a part of it. In the prediction area, the time series term is more often used than the contexts history. A definition of time series is given by BROCKWELL; DAVIS (2002). According to them, it is a set of observations ξ_{t1}, .. ., ξ_{tn} with ξ_{ti} recorded at a specific time interval *t*. Note that they refer to time intervals instead of points in time. This is due the fact that context sources measure at time intervals rather than at time instants (SIGG, 2008). Two arbitrary time intervals t_i and t_j are assumed either identical or non-overlapping.

A time series can be either discrete or continuous (BROWN; BOVEY; CHEN, 1997). The first one is that in which the observations ξ_{ti} are taken at discrete time intervals. And the second series is obtained when observations are recorded continuously over some time. Other important notion is the time series elements. Measures from context sources that share the same timestamp are grouped into a time series element. Thus, the length of a time series is defined by its number of elements. Each element can contain more than one context source's measure. And the number of different sources' samples inside an element determines the dimension of the series. Figure 1 presents a multi-dimensional time series.

Figure 1: Multi-Dimensional Time Series



Source: SIGG (2008)

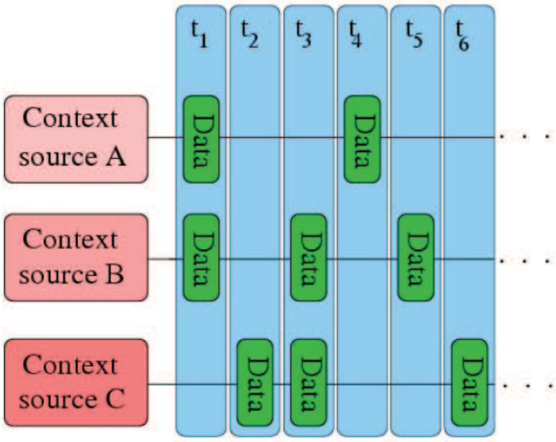
However, observe that a time series obtained from a realist scenario can contain only parts of information about the context in a time interval, see Figure 2 for an illustration. That can occurs basically for two reasons. The first one is due context sources with different sampling frequencies. And the second cause is a momentary defect in a context source. Therefore, a realist time series most probably do not match the simple generic pattern shown in Figure 1.

Other complications rise with realist series (SIGG, 2008). For instance, consider that someone wants to compare the series presented in Figure 2 with any other realist time series in order to discover similar contexts patterns. The problem is that in most cases no sub-sequence of sufficient length, regarding the same context source, can be found. To solve this problem, it is possible to interpolate all missing values in every time series or extrapolate it if the missing information is younger (older) than all sampled values. Nevertheless, this usually increases additional noise (errors) in the input data.

2.5 Contexts Prediction

Most works on context awareness consider the entities' present or their past situations. Nonetheless, some researchers also take into account the future temporal aspect. The latter

Figure 2: Real Multi-Dimensional Time Series



Source: SIGG (2008)

case of context computing is usually referred to as contexts prediction, forecasting or proactivity (SIGG, 2008). The contexts prediction term is most prominently used in conjunction with context awareness, whereas proactivity was originally coined for software agents and forecasting is most often found in relation to stochastic time series analysis.

Moreover, the term prediction is used with different meanings. MULVENNA BROWN P. (2000), for example, employs it to describe the automatic triggering of actions when some context becomes active. MULVENNA et al. (2006) and LEICHTENSTERN; LUCA; RUKZIO (2005), in contrast, use this term to describe the process of inferring a context from outputs of context's sensors. Nevertheless, in this thesis, prediction describes an operation that infers future contexts from past and present contexts (SIGG et al., 2011; KONIG et al., 2011; FOLL; HERRMANN; ROTHERMEL, 2011). According to SIGG (2008), contexts prediction can be used by applications to extend the knowledge about an observed context into the future. In other words, to adapt their behavior to events that will probably occur in the future.

A benefit of contexts prediction is that it enables systems to perform actions on behalf of the entity, that is, applications become proactive. Nonetheless, this is a delicate issue. Consider, for example, that a program decides to buy flight tickets when a potentially interesting conference will be held. The conference is really very interesting to the user and the system thus has determined that she will attend it, but what happens if her budget is not enough? According to MAYRHOFER (2005), predictions of future events will necessarily be imprecise, and in some cases they might even be impossible. Thus, it is extremely important to avoid potential problems caused by erroneous predictions. SIGG (2008) suggests that systems that exploit predictions should not automatically triggering actions that can cause serious real world effects whenever a prediction is uncertain. The same authors summarized areas in which the effects of erroneous predictions tend to be limited, which are: reconfiguration, accident prevention, alerting, and planning aid.

For contexts prediction to be possible there is a basic condition that need to be satisfied, which is the occurrence of typical patterns in the contexts history (SIGG, 2008). According to ANDERSON (2001), reproducible, typical human behavior patterns exist. In fact, in the cognitive psychology area, they are referred to as scripts (ANDERSON, 2001). A script describes the actions and circumstances that characterize a specific context or typical context pattern. Moreover, these scripts are similar even for groups of individuals; although small alterations might exist for individuals from different cultures or societies (ANDERSON, 2001). Patterns can be observed in many areas. ARGILAGA; JONSSON (2003), for instance, perceived typical behaviors in team-sport games, such as, soccer. In addition, KRSUL (1994) described how to recognize the software programmer of a piece of programming code based on her programming style. In the next subsections, it is presented some important definitions for the contexts prediction task provided by SIGG (2008).

2.5.1 Search problem

SIGG (2008) defined contexts prediction as a search problem, which was presented as:

Definition 1. *A search problem* Π *is described as:*
the set of valid inputs Λ_Π
for $I \in \Lambda_\Pi$ *the set* $\Omega_\pi(I)$ *of solutions*
An algorithm solves the search problem Π *if it calculates for* $I \in \Lambda_\Pi$ *an element* $\Omega_\pi(I)$ *if* $\Omega_\pi(I) \neq \emptyset$ *and rejects otherwise.*

In contexts prediction, the set of valid input Λ_Π is given by the set of currently observed contexts, and the set $\Omega_\pi(I)$ is given by the contexts that might happen in the future. The set of solutions can constantly change in the observed context evolution. The process that is responsible for the creation of the context evolution is called π. The set of solutions is influenced by this process. Considering that the number of input parameters for the process is enormous and they are mostly unknown, it is assumed that the process is probabilistic. The task of a prediction algorithm is to find a sequence in the environment that, at a given point of time, most likely describes the continuation of the currently observed contexts in the future. The contexts prediction task is, therefore, to find a function f that approximates the process π.

2.5.2 Prediction Quality and Accuracy

SIGG (2008) also provided a definition for the quality of prediction, which is described as:

Definition 2. *Let T denote a time series and* $d : T\,X\,T \rightarrow \Re$ *a distance metric. The quality of a prediction is measured by the distance of the predicted contexts time series to the contexts time series that is actually observed in the predicted time interval.*

According to the author, the goal of the prediction algorithm is to minimize the distance from the predicted contexts to the currently observed ones. Therefore, an optimal prediction has zero distance between the predicted and the observed contexts. A distance measure represented in a Euclidean space is the sum of the Euclidean distance between predicted and the observed contexts. Nevertheless, the total value of this distance measure is dependent on the number contexts considered.

There are two metrics commonly employed to calculate the distance between two time series, they are: the 'Root of the Mean Square Error' (RMSE) and the BIAS metric. For a predicted time series of size n, these metrics are described as:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(p_i - d_i)^2}{n}}$$

$$\text{BIAS} = \frac{\sum_{i=1}^{n}|p_i - d_i|}{n}$$

In the formulae, p_i represents the predicted context at time i, whereas d_i is the value that actually occurs at time i.

On the other hand, the accuracy of a prediction algorithm is based on the quality definition, as shown below:

Definition 3. *For any contexts prediction algorithm A, the prediction accuracy is given by the approximation quality d_A if the algorithm produces predictions whose quality is bounded from above by d_A.*

3 RELATED WORKS

In the literature, there are many works regarding models for contexts prediction. Nevertheless, most of them employ the same prediction algorithm for all cases. In other words, we did not find any approach that automatically decides the best method according to the situation. Furthermore, other important aspects of ubicomp were not explored in the related works, such as, context formal representation and privacy. This chapter is divided into five sections. Sections 3.1, 3.2, 3.3, and 3.4 describe the related works, and Section 3.5 compares them and details the comparison aspects.

3.1 Mayrhofer's Architecture

MAYRHOFER (2004) proposed an architecture for contexts prediction that is based on five steps separated by simple interfaces. The proposed stages are: sensor data acquisition, feature extraction, classification, labeling, and prediction. Due to the exact definition of the interfaces between the steps, they are mostly self-contained and can be exchanged independently. Below, the operation of each step of the architecture is presented:

- *sensor data acquisition* - it provides data streams (time series) of measurements (raw sensor data). Usually some physical values, such as, the incoming Radio Frequency signals are the base for the measurements;

- *feature extraction* - it extracts information from raw sensor data using domain-specific methods. In this step, the available data is deliberately simplified, transformed or even expanded;

- *classification* - it tries to find common patterns in the feature space, which are called classes;

- *labeling* - it assigns to the classes or combinations of classes descriptive names, aiming to easy the presentation of the detected contexts to the users;

- *prediction* - it predicts future contexts.

The different processing steps can be regarded as filters, transforming input values to output values. No central context repository is necessary; every step is independent and performs online data processing. Thus, the architecture is well suited for resource limited information appliances, but does not exclude the integration of complex modules with high demands on computational resources. When available, powerful processing or storage components can be used. Although modules from different steps (e.g. a classifier and a predictor) might use extensive data storage facilities, such as, a central database server, they can do so independently.

However, in the proposal of MAYRHOFER (2004), contexts prediction is based only on high-level contexts. In addition, the author does not approach aspects regarding privacy and specification for context information. And most importantly, the work does not have any mechanism to support an adaptive strategy for contexts prediction.

3.2 Sigg's Architecture

SIGG (2008) proposed an architecture for contexts prediction. In his work, he considered that prediction is composed of several interconnected operations. Furthermore, he argued that a general architecture should support the addition and the interchange of these operations. The operations, therefore, are viewed as interchangeable modules that constitute the computational possibilities of the architecture. The modules can be chosen and organized by the designers according to the functionalities that they desire. There are two types of modules, the optional and the obligatory. The optional modules provide facets that are not vital for the proper operation of the architecture, but they might add further context sources or improve existing ones. The obligatory modules, on the contrary, are necessary for the correct operation of the architecture at a given task. The author considers that the following modules are vital for contexts prediction to work as expected: the prediction algorithm, a learning process, the contexts history, and the rule dataset.

The architecture is divided into four layers, which are: data acquisition, prediction, interpretation, and application. Furthermore, the architecture enables prediction on low-level and high-level context elements alike. For low-level, the prediction module is placed between the context interpretation and the acquisition layer. However, for approaches based on high level, the prediction module placed between the interpretation and the application layer. Therefore, the architecture supports different context abstraction levels. In addition, it also has a learning mechanism. Nevertheless, SIGG (2008) does not describe any adaptive mechanism for prediction neither consider privacy or specification for context information.

3.3 The Structured Contexts Prediction Framework

The Structured Contexts Prediction architecture was implemented in a framework that aims to ease the employment of existing prediction methods (MEINERS; ZAPLATA; LAMERSDORF, 2010). It is based on two major principles. The first is the use of knowledge about the application domain as valuable information that has to be incorporated by the developers at design time. And the second principle is the combined application of multiple and exchangeable prediction algorithms. Thus, the framework allows developers to selected and combine suitable methods to ensure accuracy and efficiency of domain-specific demands.

The knowledge about the application domain is referred to as prediction model. This model specifies the way predictions have to be performed and sets up the prediction system. In other words, it assigns algorithms to each variable in order to predict its value. For example, consider that a user telephoning is represented by a Boolean variable. This variable's value is predicted by a method that uses the values of other variables, such as, the time of day and the position of the user, which are again predicted by their variables' methods.

Initially, the framework established a reference configuration mainly based on linear regression and probability tables. However, the set of available strategies can be extended by implementing new and possibly application-dependent approaches. Probability tables refer to methods that store occurrence frequencies of variable/value combinations as knowledge. The properties of the two strategies complement one another and are therefore well suited for adaptive application. From the developer's viewpoint, the whole process of using the framework consists of two parts. The first is the development of the prediction model at design time. And the second part is the retrieval of predictions by the respective application at runtime.

On the one hand, the Structured Contexts Prediction architecture does regard privacy neither context formal representation. On the other hand, it does support an adaptive mechanism for contexts prediction. However, this mechanism is manual, that is, the designer needs to choose at design time the most suitable algorithms for predictions. Furthermore, the architecture also has a learning component and support low-level context data.

3.4 The PreCon Model

PreCon is a model for contexts prediction which applies well-known methods of stochastic model checking (BAIER; KATOEN, 2008) (the same used for the verification of distributed communication protocols) to the analysis and prediction of human behavior (FOLL; HERRMANN; ROTHERMEL, 2011). The stochastic models, called Stochastic User Models (SUM), are representations of human behavior that are learned from traces of past context changes. The SUMs are represented as Semi-Markov Chains such that the changes in contexts are regarded as a stochastic process. The authors use temporal logics as a query language, enabling applications to specify expressive temporal properties on future context. For a prediction, the model verifies with which probability these properties hold on a given SUMs.

The authors assume that a context recognition system monitors the context of the user and records context traces (time-stamped series of consecutive context changes) in histories. For instance, a context trace may contain information about which activities have been executed at what time and location. The context traces are given as input to the learning algorithm, which processes them to obtain an SUM. PreCon can answer time-dependent queries, for example, will the user be executing activity A at location X within the next 10 minutes?

Nevertheless, the PreCon model does not approach the aspects of privacy neither specification for context. Moreover, the model does not describe any adaptive mechanism for the prediction task. However, PreCon has a learning capacity and supports low-level context.

3.5 Comparison among the related works

Table 1 shows the comparison among the related works. As we can see, none of the studies supports automatic adaptive approach, context formal representation, and privacy. Although the structured architecture of MEINERS; ZAPLATA; LAMERSDORF (2010) has an adaptive strategy, it is manual, that is, the designer needs to choose at design time the most suitable algorithms for predictions. Furthermore, we included other two comparison characteristics (i.e. low and high context levels and learning capability) that we considered relevant for a contexts prediction model. The comparison aspects are described in details in the following subsections.

Table 1: Comparison among the related works

	Adaptive Approach	Context Formal Representation	Privacy	Low and high context levels	Learning Capability
Mayrhofer's model	No	No	No	No	Yes
Sigg's architecture	No	No	No	Yes	Yes
Structured architecture	Manual	No	No	Yes	Yes
PreCon model	No	No	No	Yes	Yes

Source: Made by the author

3.5.1 Adaptive Approach

The essential part of a prediction model is the algorithms used in the predictions. There are many methods for this task. However, there is no single approach best suitable for all cases. The Alignment method, for example, had the best performance considering high accuracy and high prediction horizon. Nonetheless, for low prediction horizons and high search spaces, the Markov approach outperformed that technique in many cases (SIGG, 2008).

Furthermore, in cases where the entity had a new contexts sequence, the Collaboration approach was better than the Alignment (VOIGTMANN; LAU; DAVID, 2011). Notwithstanding, it also has a shortcoming, because it does not support fuzziness, i.e. it does not apply approximation techniques to find matches in the contexts history. Therefore, in this work we consider that the adoption of an adaptive strategy is the most reasonable approach for contexts prediction models.

3.5.2 Context Formal Representation

Ubiquitous environments are highly dynamic, that is, applications can interact with a great number of different and unknown applications all the time (WEISER, 1991). Hence, it is fundamental to define a formal representation for context, so that different systems can communicate. It is possible to note the importance of a formal representation by analyzing the studies on context modeling and ubiquitous user modeling.

In the context modeling area, for instance, DEY; ABOWD; SALBER (2001) provided a classical categorization for context, in which they proposed four basic categories that act as indices to other sources of contextual information. Other example is the Composite Capabilities/Preference Profile (CC/PP) model (KLYNE et al., 2005), which is a markup-based W3C standard for description of mobile devices. Furthermore, there is the Context Modeling Language (CML), which is a tool to assist designers with the task of exploring and specifying the context requirements of a context-aware application (HOAREAU; SATOH, 2009).

On the ubiquitous user modeling side, HECKMANN; KRUEGER (2003) proposed an XML-based user modeling mark-up language (named UserML) as a platform for communication in ubiquitous environments, which supports personalized interactions of users with different devices and privacy. These works aim to model ubiquitous scenarios in a way that different applications from distinct domains be able to communicate and understand each other's concepts. Therefore, specification context representation is considered as an important characteristic for a prediction model.

3.5.3 Privacy

According to LANGHEINRICH (2009), privacy and data protection have always been closely related to what is technically feasible. For instance, at the end of the 19^{th} century, the invention of modern photography made people to rethink the concept of legal privacy protection. At the beginning of the 20^{th} century, laws had to be reinterpreted again to consider the possibilities of modern telecommunications. And in the 1960s and 1970s, the implementation of techniques to make the USA government more efficient through the use of modern databases required yet another update of privacy laws. In these cases, technology changed what was possible in the everyday and thus prompted a realignment of the notion of privacy.

The dawn of ubiquitous computing promises the next revolution of "smart things". Using miniature sensors, cheap microchips, and wireless communication, computer technology can penetrate our everyday lives in a completely unobtrusive manner. In the same way, real world facts can be mapped on a computer with an unprecedented reliability and efficiency. Thus, the boundary between the real and virtual world seems to disappear. According to LANGHEINRICH (2009), data protection and privacy is all about the mapping between the real and the virtual world. In his 1991 Scientific American article, Mark Weiser already identified privacy as one of the biggest challenges of ubicomp: "Perhaps key among the social issues that embodied virtuality will engender is privacy: hundreds of computers in every room, all capable of sensing people near them and linked by high-speed networks, have the potential to make totalitarianism up to now seem like sheerest anarchy." (WEISER, 1991).

Therefore, privacy is considered as an essential aspect for contexts prediction models. Privacy is regarded in two ways. Firstly, it is assumed that entities will want to use predictions made for other entities. For instance, a specific entity house may want to receive prediction about its owner's location (other entity) in order to customize the environment to her taste. Thus, it is necessary a mechanism to control that kind of privacy. In second place, it is necessary to consider algorithms that use multiple entities' histories to make prediction for a single one. Many entities do not wish to share their entire histories. Some of them may want to divide only a certain amount of their data or nothing at all. Therefore, it is needed a mechanism to support privacy related to this characteristic.

3.5.4 Low and high context levels

The context abstraction levels in the different stages of context processing are often referred to as high-level, low-level, and raw data (SIGG, 2008). WANT et al. (1995), for example, provided a rough distinction between low-level and higher-level contexts. Low-level is employed to describe data obtained directly from sensors, whereas the high-level definition is used for processed information. This processing can be, for example, semantic reasoning, an interpretation, data calibration or noise removal.

Other classification of context abstraction levels was provided in the work of MANTYJARVI; TUTKIMUSKESKUS (2003). According to them, a condition represents low-level context, whereas an activity describes high-level information. The lowest abstraction level can be, for example, 20°C or 80% humidity, which could also be referred to as 'warm' or 'high humidity' respectively. A high-level context, on the contrary, is an activity, such as, 'having lunch'.

According to these definitions, higher-level contexts are obtained by further processing lower level data. Nonetheless, SIGG (2008) introduced an alternative distinction about context abstraction, which is based on the amount of processing applied to contexts. Therefore, following this model, the context abstraction rises with the amount of processing applied. In fact, the number of distinct context abstractions is not restricted to any finite set, such as, for instance, 'raw data', 'low level', and 'high level'. Actually, it is expected a fine grained transition among context abstractions.

SIGG (2008) divided the context processing task into three operations, which are: the acquisition of the data, its interpretation, and the prediction itself. The amount of interpretation employed in the information determines its abstraction level, i.e. the more interpretation applied, the higher the context level. Moreover, the authors studied the impact of applying the prediction before and after the interpretation step for the alignment algorithm (SIGG; HASELOFF; DAVID, 2010). They also considered other attributes to calculate the accuracy, which are: (1) the dimensions of the input time series; (2) the length of the contexts history; and (3) the accuracy of the context interpretation.

As result, SIGG (2008) noticed that a higher prediction accuracy for an increased dimension of the input time series can be achieved when contexts prediction is applied after the context interpretation process. An opposite result was obtained for the length of the contexts history. Using an increasing contexts history size, the prediction accuracy is higher when contexts prediction is applied prior to the context interpretation stage.

Moreover, the authors realized that the accuracy of the context interpretation has a significant impact on the contexts prediction accuracy. For increasing error probabilities of the context interpretation operation, they noticed a tendency that the prediction accuracy is higher when prediction is applied prior to the context interpretation process.

Summarizing the outcomes obtained by SIGG (2008), in cases where the context interpretation operation has difficult to handle noisy input data, it is more favorable to employ contexts prediction in advance of context interpretation. Nevertheless, when context interpretation is highly accurate, the application of contexts prediction after the context interpretation might produce improved contexts prediction accuracy. Therefore, from this study, it is possible to identify one desirable characteristic for contexts prediction models. This aspect refers to the ability of supporting predictions from the lowest to the highest context abstraction level.

3.5.5 Learning Capability

In ubiquitous scenarios, environments rapidly change on a microscopic level (e.g. single context sources) as well as on a macroscopic degree (e.g. the behaviors and habits of humans that gradually change over time) (SIGG, 2008). There are cases where acts not frequently performed, such as, a change between jobs or an uncommon displacement, might impose sudden and drastic macroscopic environmental modifications. Therefore, to keep high prediction accuracy in this changing environment an adaptation mechanism is required. In other words, a learning capability is a fundamental aspect for contexts prediction architectures.

The learning mechanism might extract rules or functions from the contexts history. According to SIGG (2008), there are many properties that might be obtained and are important for the description of the context evolution, which are:

- Trends in numerical contexts time series;
- Context patterns that repeatedly occur;
- The absence or presence of specific context sources;
- The length of context durations or the frequency of context changes.

4 CONTEXT FORMAL REPRESENTATION OF HECKMANN

In this chapter, we describe the context formal representation used in ORACON. As we discussed in Section 2.2, there are many researches for modeling context as well as ubiquitous users. However, considering aspects, such as, ontological approach and decoupling from syntax and semantic, the work of HECKMANN (2005) was considered the most complete approach to represent entities' contexts. HECKMANN (2005) proposed two markup languages (SituationML and SituationQL) and two ontologies (GUMO and UbisWorld). SituationML carries information about entities model entries and context data. It is a uniform structure for representing any entities' models and their contexts. SituationQL is a query language that allows retrieval of data specified in the format of SituationML.

In this chapter, we describe in details SituationML, SituationQL, GUMO, and UbisWorld. The chapter is divided into four sections. The first one presents the SituationML language. The second section describes the SituationQL language. The third approaches the GUMO ontology. And the fourth and last section deals with the UbisWorld ontology.

4.1 The SituationML Language

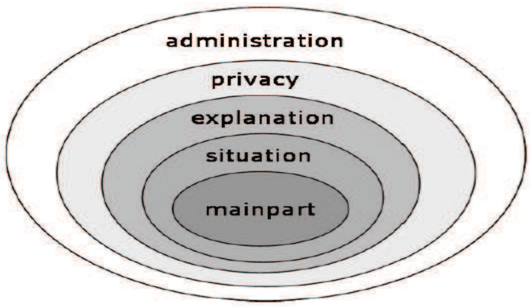
The SituationML language is also referred to as ContextML or UserML, since it covers the conceptual purpose of user models and context models (HECKMANN, 2005). The ML at the end of the terms stands for Markup Language. The SituationML is, in fact, a XML-based technological realization of the abstract model called SituationReports and SituationalStatements, which inherent ideas from the three different areas: User Modeling, Ubiquitous Computing, and Semantic Web. To understand the SituationalStatements model, HECKMANN (2005) described a scenario annotated with semantic meta-information, which is:

"An inference system (creator) deduces that Peter (subject) is now (start) most probably (confidence) under high (object) time pressure (predicate), because (evidence) he is near the duty-free shop (position) of the airport (location), while boarding of his flight closes in a few minutes (evidence). Additionally, his walking speed sensors (creator2) report fast walking (predicate2, object2). According to his (owner) privacy settings (privacy), this information is only freely available to preselected people and systems (access)."

The object and predicate names are inherited from the naming of the original RDF triple. The object carries the value and the predicate the attribute. The annotation example shows that it is possible to separate the whole description of this airport scenario into smaller, sentence-like units, for example, a sentence regarding Peter's time pressure, other concerned about Peter's walking speed, and probably a last sentence about the boarding time of Peter's flight. Nevertheless, the author argues that these intuitive semantic roles do not lead straight forward to a clear structure, if it is desired to omit the complexity of natural language.

Some different approaches from knowledge representation researches influenced the design decisions of the SituationalStatements. The main of them is the Resource Description Framework (RDF), with subject-predicate-object triples, reification, collections and all the theory of semantic web in support, as well as other semantic web languages, such as, OIL, DAML or OWL. The analysis of related markup languages and other works led to the creation of the SituationalStatements model. SituationalStatements represent partial descriptions of situations, such as, user model entries, context information, or even low-level sensor data. The main differential of this model is its extensive layered approach, which is presented in the onion model of Figure 3.

Figure 3: Onion model to represent the five layers of situational statements



Source: HECKMANN (2005)

The information is organized in a predefined hierarchy of meta-levels wrapped around the main part. These layers of meta-level information can be seen as a collection of slots or attributes that are arranged in five boxes, which are: mainpart box, situation box, explanation box, privacy box, and administration box. These boxes have an organizing and structuring functionality. Their meanings are presented in Table 2.

The scenario of the passenger at the airport can now be described by the SituationalStatements and the SituationReports representations. The SituationalStatements definition has already been presented, however, the SituationReports concept still needs clarification. A SituationReport is a bag of SituationalStatements, see Figure 4 for an illustration. The airport scenario could be represented by the three SituationalStatements inside a SituationReport, Figure 5. Note that the attribute values of the SituationalStatements are not correctly presented. They are only indicated and should be understood as simplified classes.

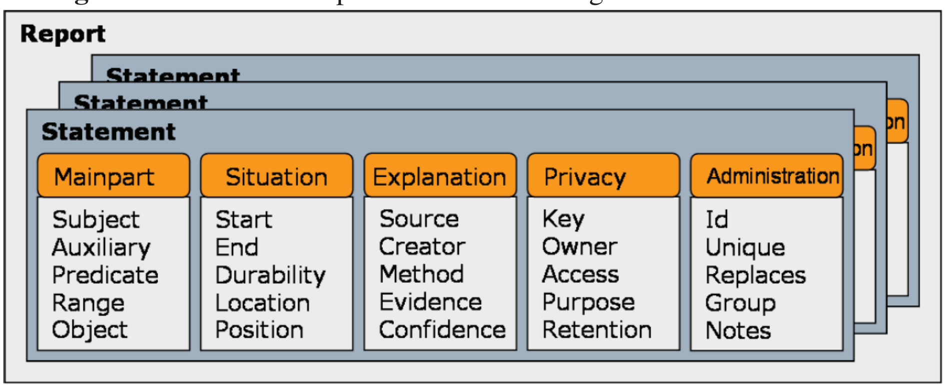
The XML/RDF technological realization of the abstract models SituationReports and SituationalStatements is called SituationML. In addition, it is also referred to as either UserML or ContextML. Figure 6 presents the representation for SituationalStatements in SituationML. Note again that the attribute values of the SituationML are not correctly presented. They are only indicated and should be understood as simplified classes.

Table 2: Description of the SituationalStatements boxes

Mainpart	*the basic, rdf-based five-tuple of situational statement attributes*
subject	the main entity the statement is about
auxiliary	the auxiliary part of the predicate, such as, "hasProperty" or "hasInterest"
predicate	the dimension or category of the entity, such as, "Walking" or "Sleeping"
range	the range of the object attribute, defaults are possible
object	the value for the subject-auxiliary-predicate triple of the statement
Situation	*temporal and spatial constraints, restricting the mainpart*
start	the point of time when this statement was given
end	the point of time when this statement is no longer valid
durability	the qualitative time span of how long the statement is valid
location	the qualitative spatial location where this statement takes place
position	the quantitative spatial location, the exact coordinates
Explanation	*a collection of explanatory meta attributes*
source	the origin where the statement is stored
creator	the person or system that is responsible for the creation of this statement
method	the manner by which the statement was created
evidence	a list of evidence that supports the statement
confidence	a number that displays the creator's expected truth of the statement
Privacy	*a collection of privacy meta attributes, controlling the distribution*
key	optional encrypted security key that can be attached to the statement
owner	the person or system that may manage the distribution
access	the class of users or systems that are allowed to use the statement
purpose	the qualitative purpose for which the statement may be used
retention	the qualitative time of how long the statement may be kept or used
Administration	*a collection of administrative meta attributes*
id	a locally unique identifier for referencing the statement in the database
unique	a globally unique identifier for referencing the statement
replaces	a unique identifier of another statement that has to be replaced
group	rough classification of the statement, such as, "UserModel" or "SensorData"
notes	an additional attribute with free semantics, can serve as a variable

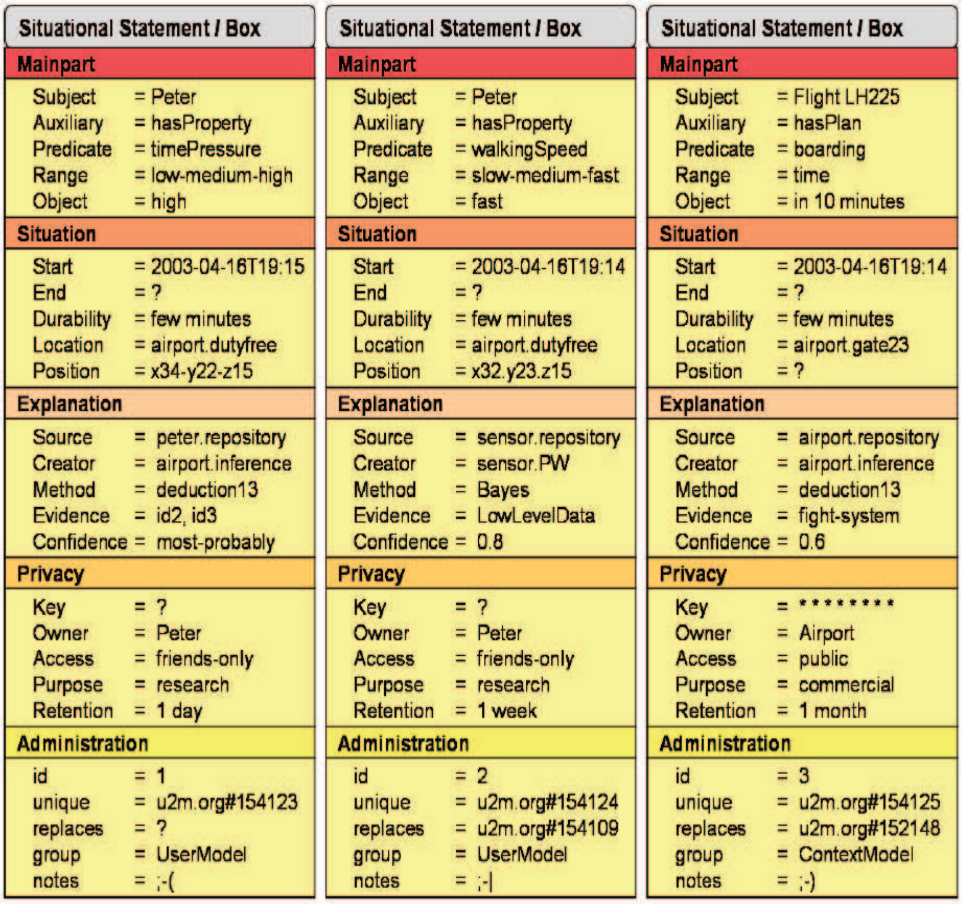
Source: HECKMANN (2005)

Figure 4: A SituationReport is defined as a bag of SituationalStatements



Source: HECKMANN (2005)

Figure 5: SituationReport with three SituationalStatements from the airport scenario



Source: HECKMANN (2005)

4.2 The SituationQL Language

Besides of defining a format for the histories' information, we needed to provide a protocol for the data retrieval. For this task, we decided to use the SituationalQL language, which is also called either UserQL or ContextQL (HECKMANN, 2005). The QL at the end of the terms stands for Query Language. The SituationalQL is, actually, a XML-based technological realization of the abstract model called SituationalQueries. SituationalQueries form the counterpart to SituationalStatements, since each situation attribute finds a corresponding attribute in the model of SituationalQueries.

The SituationalQueries model is composed of three groups of attributes, which are: **match**, **filter**, and **control**, which are presented in Table 3. These groups correspond to the macro-steps of the query evaluation process, presented in Figure 7. The attributes of the select step, shown in the figure, are described together with the control attributes. The **select step** chooses the repositories to which the query is applied, since it is possible to have distributed contexts histories. The **match step** returns all statements that match the corresponding query attributes. Moreover, it integrates semantic functionality, such as, the ontological sameAs and the spatial closeBy. The **filter step** filters out further unwanted statements.

Figure 6: SituationML representation

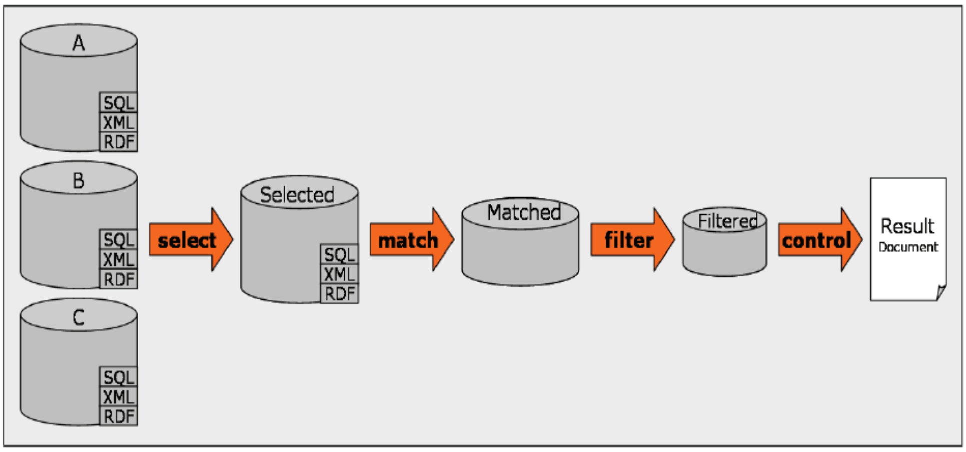
```
Situational Statement / XML (Mix)

<statement>
        <mainpart
                subject = "a1"
                auxiliary ="a2"
                predicate = "a3"
                range = "a4"
                object = "a5" />
        <situation
                start = "a6"
                end = "a7"
                durability = "a8"
                location = "a9"
                position = "a10" />
        <explanation
                source = "a11"
                creator = "a12"
                method = "a13"
                evidence = "a14"
                confidence = "a15" />
        <privacy
                key = "a16"
                owner = "a17"
                access = "a18"
                purpose = "a19"
                retention = "a20" />
        <administration
                id = "a21"
                unique = "a22"
                replaces = "a23"
                group = "a24"
                notes = "a25" />
</statement>
```

Source: HECKMANN (2005)

The **control step** performs conflict resolution and transforms the final statements into the returned result, which has the format of a SituationReport. Each statement is individually checked if it passes the privacy filter, the confidence filter and the temporal filter. The privacy filter checks if the *access* from SituationML is either set to public, private, or a set of friends. If it is set to a set of friends, the process verifies if there is a friendship between the *requestor* from SituationQL and the *owner* from the SituationML. On the other hand, if it is set to private the process checks if the *requestor* attribute from SituationQL is the same as the *owner* property from SituationML.

A list of SituationalQueries is referred to as SituationRequest, see Figure 8 for an illustration. A SituationRequest is sent to the External Histories, which resolve each query after the other in a row (in parallel) and returns the resulting SituationReport. The XML technological realization of the abstract models SituationalQueries is called SituationQL, ContextQL, or UserQL. Figure 9 shows an example of a SituationQL in XML. All attributes are defined by

Figure 7: Macro-steps of the query evaluation process



Source: HECKMANN (2005)

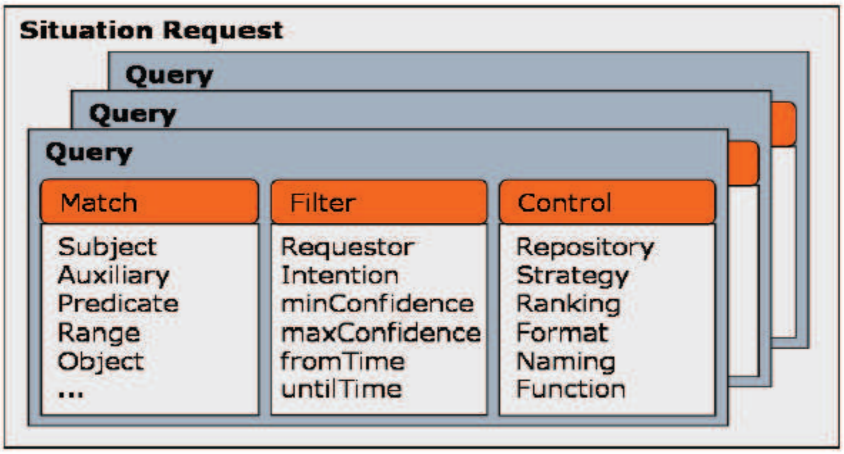
Table 3: Attributes of SituationalQueries with default values

Match	*all attributes introduced for SituationalStatements*
subject	selecting the main statement entity, default: any
auxiliary	selecting the auxiliary part of the property, default: any
predicate	selecting the predicate part of the property, default: any
range	selecting the range part of the property, default: any
object	selecting the object, default: any
id	selecting the statement by id, default: any
group	selecting the group of statements, default: UserModel
location	selecting the spatial extension of the statements, default: any
...	see table 2 for a complete list of attributes since every situational statement attribute can be used as a matching attribute in the query
Filter	*a collection of filter attributes*
requestor	the requesting user or system, default: anonymous
intention	what is intended to be done with the statement, default: commercial
minConfidence	minimal confidence value that must hold, default: 0
maxConfidence	maximal confidence value that must hold, default: 1
fromTime	start of the time interval, default: whenever
untilTime	end of the time interval, default: now
Control	*a collection of control attributes*
repository	the chosen, respondent situation container, default: system's choice
strategy	conflict resolution strategies, default: latestOnly
ranking	sorting and ranking of the results, default: newestFirst
naming	manipulating the appearance of the names, default: longName
format	manipulating the appearance of the XML format, default: UserML
function	applying evaluation functions to the results, default: none

Source: HECKMANN (2005)

XML-elements with their original name, whereas the corresponding attribute groups are omitted. SituationQL define up to 37 attributes, nonetheless the average query will be short, once empty elements can be omitted. The variables from q1 to q37 can carry ordinary RDF node values but also more complex UbisExpressions (HECKMANN, 2005).

Figure 8: A SituationRequest consists of a set of SituationalQueries



Source: HECKMANN (2005)

4.3 GUMO - the General User Model Ontology

The SituationalStatements concept of dividing the descriptions of user model dimensions into the three parts (i.e. auxiliary, predicate, and range) influenced the construction of the General User Model Ontology (GUMO for short) HECKMANN (2005), see Figure 10. Using this approach, if someone wants to describe his or her interest in football, for instance, he or she could divide this so-called user model dimension into the auxiliary part (has interest), the category part (football), and the range part (low-medium-high), as illustrated in Figure 11. Other example is a system that wants to describe a user's knowledge about Beethoven's Symphonies. In this case, it could divide this knowledge into the triple "has knowledge", "Beethoven's Symphonies", and "poor-average-good-excellent", as shown in Figure 12.

Therefore, the implication for GUMO of these examples is a clear separation in the modeling of user model auxiliaries, predicate classes, and special ranges. HECKMANN (2005) identified a group of auxiliaries, which is presented in Table 4. This listing is not intended to be complete; on the contrary, it is a start point with which most of the important user model statements can be realized.

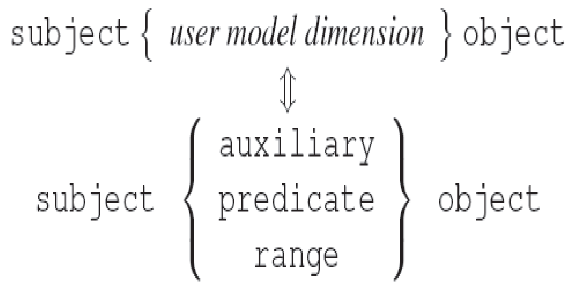
The auxiliary "has Property", for instance, describes more user-centric predicates, which are called Basic User Dimensions. The auxiliaries "has Interest" and "has Preference", on the other hand, are mainly directed to model users' interests, such as, music categories or film genres. Usually these auxiliaries lead to domain-dependent predicates. The auxiliaries "has Regularity" and "has Done" corresponds to the so-called Usage Data as defined by ALFRED (2001) and the

Figure 9: SituationQL Language

```
<query>
      <!-- Match Attributes -->
      <subject>       q1   </subject>
      <auxiliary>     q2   </auxiliary>
      <predicate>     q3   </predicate>
      <range>         q4   </range>
      ...             ...  ...
      <unique>        q22  </unique>
      <replaces>      q23  </replaces>
      <group>         q24  </group>
      <notes>         q25  </notes>

      <!-- Filter Attributes -->
      <requestor>     q26  </requestor>
      <intention>     q27  </intention>
      <minConfidence> q28  </minConfidence>
      <maxConfidence> q29  </maxConfidence>
      <fromTime>      q30  </fromTime>
      <untilTime>     q31  </untilTime>

      <!-- Control Attributes -->
      <repository>    q32  </repository>
      <strategy>      q33  </strategy>
      <ranking>       q34  </ranking>
      <naming>        q35  </naming>
      <format>        q36  </format>
      <function>      q37  </function>
</query>
```

Source: HECKMANN (2005)

Figure 10: User Dimensions in SituationalStatements



Source: HECKMANN (2005)

Low Level Sensor Data. The auxiliary "has Location" is related to a spatial ontology, such as, the Spatial Ontology from UbisWorld, discussed in Section 4.4.

The **Basic User Dimensions** categorization in GUMO describes user-centric predicates.

Figure 11: GUMO User Dimensions of Football Interest

$$\mathtt{subject}\left\{\begin{array}{rcl} \mathtt{auxiliary} & = & \text{has interest} \\ \mathtt{predicate} & = & \text{football} \\ \mathtt{range} & = & \text{low-medium-high} \end{array}\right\}\mathtt{object}$$

Source: HECKMANN (2005)

Figure 12: GUMO User Dimensions of Beethoven's Symphonies knowledge

$$\mathtt{subject}\left\{\begin{array}{rcl} \mathtt{auxiliary} & = & \text{has knowledge} \\ \mathtt{predicate} & = & \text{Beethoven's symphonies} \\ \mathtt{range} & = & \text{poor-average-good-excellent} \end{array}\right\}\mathtt{object}$$

Source: HECKMANN (2005)

Table 4: List of User Model Auxiliaries

Group	**Name**	**Id**
UserModelAuxiliary	has Property	600100
UserModelAuxiliary	has Interest	600110
UserModelAuxiliary	has Believe	600120
UserModelAuxiliary	has Knowledge	600130
UserModelAuxiliary	has Preference	600140
UserModelAuxiliary	has Regularity	600150
UserModelAuxiliary	has Plan	600160
UserModelAuxiliary	has Goal	600170
UserModelAuxiliary	has Done	600180
UserModelAuxiliary	has Location	600190

Source: HECKMANN (2005)

Furthermore, it is based on the work of JAMESON (2001). However, it differs from this approach because in GUMO the modeling of the dimensions can be split into auxiliaries, predicates, and ranges. This strategy leads to powerful interplay between the SituationalStatements and the ontology. Figure 13 shows various groups of basic dimensions that were modeled. Nevertheless, the complete list can be found in the dissertation of HECKMANN (2005).

The **Domain Dependent User Dimensions** are related to predicates describing users' interests, such as, music categories or film genres. It differs from Basic User Model Dimensions with respect to the required additional general-world knowledge. For instance, consider that it is desired to express someone's interest in certain film category or certain wine types, in those

Figure 13: Some groups of basic user dimensions

Source: HECKMANN (2005)

cases it is necessary a domain ontology for film as well as wines. However, note that any concept in the whole world is a potential candidate for expressing user model data about interests, preferences, or knowledge. Therefore, GUMO architecture is opened to any external ontology and it expresses user model data with the modularized SituationalStatements. Nevertheless, GUMO also supports interest categories as combining elements between GUMO and external ontologies.

The GUMO interest categories form a large listing of interest and preference groups, such as, film genres, music trends, sports, pc-game genres, environmental topics, among others. Figure14 presents the main categories supported by GUMO and Figure 15 shows preference settings within the museum's. The entire list of the interest categories can be found in the dissertation of HECKMANN (2005).

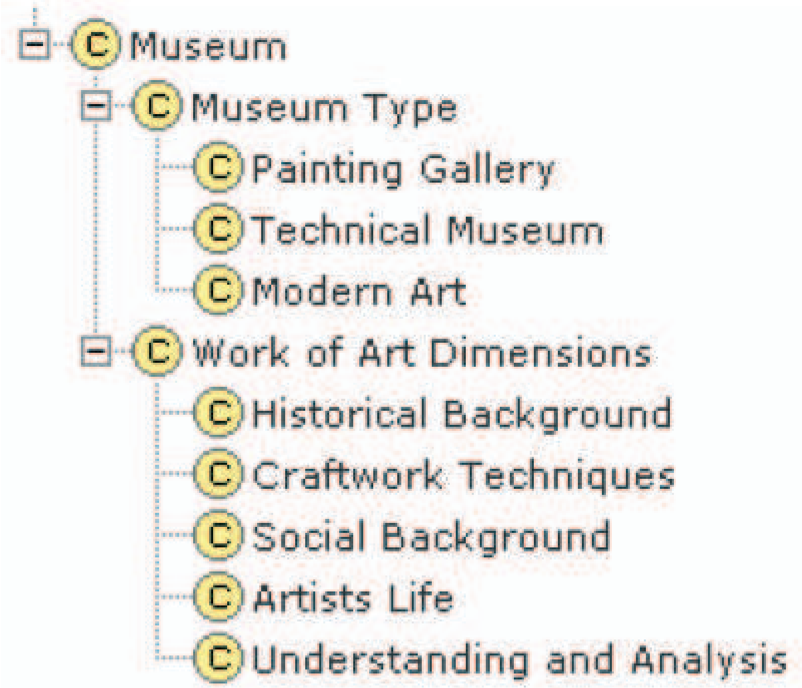
4.4 The UbisWorld ontology

UbisWorld extended the Blocks World (SLANEY; THIéBAUX, 2001) and the context toolkit (DEY; ABOWD; SALBER, 2001) to the special needs of contextualized interaction in ubicomp environments with user modeling and privacy. According to HECKMANN (2005), Ubis abbreviates the Ubiquitous term, whereas the postfix World indicates that the approach tries to be

Figure 14: Interest Categories Supported by GUMO

Source: HECKMANN (2005)

Figure 15: Categories in the Museum Domain



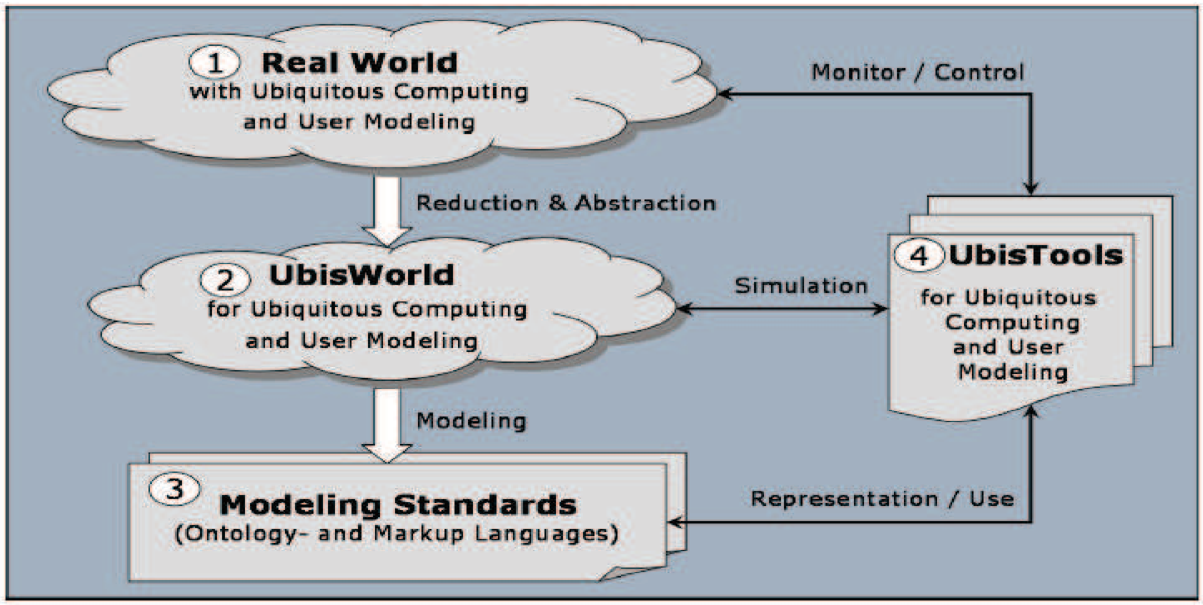
Source: HECKMANN (2005)

very broad and it refers to the blocks world. HECKMANN (2005) described UbisWorld as a collection of concepts and models for location and time, for interaction and context that are all prepared for ontological representation and data collection (HECKMANN, 2003).

UbisWorld can be used to describe some parts of the real world, such as, an office, a shop, a museum, or an airport. It represents people, objects, locations as well as times, events, and their properties and features. Besides of the representational function, UbisWorld can be employed for simulation, inspection and control. Figure 16 presents the conceptual view of the real world.

Cloud number (named Real World) in Figure 16 represents the points of interest of the real

Figure 16: Conceptual view of the real world



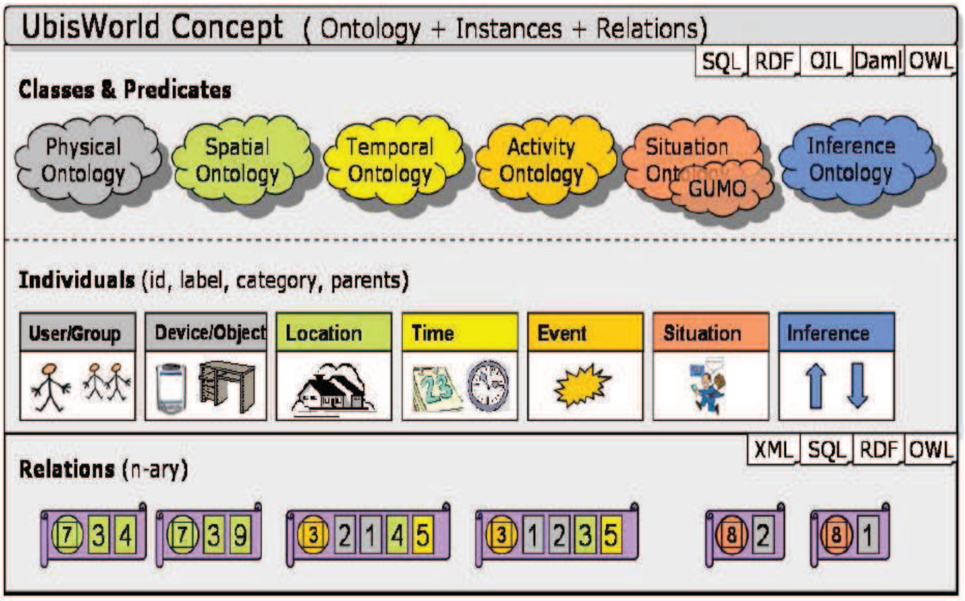
Source: HECKMANN (2005)

world with ubiquitous computing and user modeling. UbisWorld, presented as cloud number two, is realized by reduction and abstraction from the real world. The modeling standards (such as, markup languages and ontology definitions) are derived from UbisWorld abstractions. UbisTools, on one hand, operates on the level of UbisWorld and, on the other hand, it operates on the level of the real world.

HECKMANN (2005) provided a small example to illustrate the interrelation between a simulated world and a real world, which is: *"Imagine that there is a room in the real world with two doors, two light switches at each door and one light at the ceiling. All these elements will be represented in an abstract manner in the corresponding UbisWorld model. The UbisTools could simulate the light-on light-off behavior of the real room, such that if the virtual light switch gets pressed the virtual light in UbisWorld shines, independently from the real world. Secondly, the UbisWorld room could be used to monitor the real room in such a way that every time when the real switch has been used, the virtual light shows the status "shining". As a third possibility, the UbisWorld room could be used to control the real world room for example by turning the real light on or off, every time when the virtual light switch is used."* It is possible to learn from this example that the ontology engineering of UbisWorld is independent from the task of representation, simulation, monitoring or even control of the real world.

UbisWorld is composed of specialized partial ontologies, rather than one ontology for all aspects. Figure 17 presents in one diagram the elements of the UbisWorld concept, which consists of classes and predicates, of individuals, and of relations. The classes and predicates are defined in six additive ontologies, which are: the physical ontology, the spatial ontology, the temporal ontology, the activity ontology, the situation ontology with situation describing dimensions that also cover the general user model ontology GUMO, and the inference ontology that models the computational and intelligent behavior in ubiquitous computing environments.

Figure 17: UbisWorld



Source: HECKMANN (2005)

The *Physical Ontology* introduces physical objects. Physical elements are people, devices, objects, furniture, goods, food, and so on. The *Spatial Ontology* deals with the spatial arrangement of physical objects in ubicomp environments. The third partial ontology in UbisWorld is the *Temporal Ontology*. It has a clear model of time and time-intervals, since most statements are related over the temporal dimension. The *Activity Ontology* describes the changes in the world and the most prominent one is change of location. The *Situation Ontology* describes attributes, parameters or properties about users, systems, locations or activities. A situational parameter for a location could for example be the Noise Level, the Weather conditions, or the available Light. A situational parameter for a person could for instance be his/her Blood Pressure, his/her Cognitive Load or his/her Interests. The defined general user model ontology GUMO can be considered as being part of the Situation Ontology. A situational parameter about a device or system could for example be its remaining Battery Power or its Screen Size. Finally the power of computing and intelligent behavior enters the UbisWorld by the Inference Ontology. Inference elements define smart rules or proactive inference processes within intelligent instrumented environments.

5 ALGORITHMS FOR CONTEXTS PREDICTION

In this chapter we choose and describe the algorithms that will be supported by the model. It is divided into two sections. The first compares available contexts prediction algorithms, selecting the most suitable ones to be used in ORACON. And the second section describes in details the chosen methods.

5.1 Comparison of Contexts Prediction Algorithms

SIGG (2008) described some requirements for the contexts prediction and analyzed several algorithms according to them. He observed that there are further statistical methods or variations of the described techniques that could also be used for prediction. Nevertheless, his choice of methods represented the most commonly applied and straightforward approaches. After deciding the most suitable algorithms according to the established requisites, Sigg performed analytical tests with real data from two different domains. The following three paragraphs of this section summarize the decision of algorithms of SIGG (2008), whereas the three last ones describe more recent methods and compare them all.

Figure 18: Algorithms Comparison

	SOM	Markov	SVM	ARMA	Kalman	Align	State
Count of typical patterns	fixed[11]	variable	fixed[11]	0[12]	0[12]	variable	1
Numeric context types	yes	yes[13]	yes	yes	yes	yes	no
Non-numeric context types	no	yes	no	no	no	yes[14]	yes
Complexity[15]	$O\left(k^3\right)$	$O(k\|C\|^2)$	$O(k)$	$O(k\log(k))$	$O(k^4)$	$O(k^3)$	$O(k\|C\|^2)$
Learning ability	yes[16]	yes	yes	no [17]	no[17]	yes	no
Approximate pattern matching	yes	no	yes	no[12]	yes	yes	no
Multi-dim. TS	yes	yes	yes	yes	yes	yes	yes
Discrete data	yes	yes	no	yes	yes	yes	yes
Variable length typical patterns	yes	no	no	no[12]	no[12]	yes	no
Multi-type TS	no	yes	no	no	no	yes	no
Continuous data	no	yes	yes	yes	yes	no	no
Pre-processing	yes	no	yes	no	no	yes	no
Prediction of durations	yes	no[18]	yes	yes	yes	yes	no
Continuous time	no	yes	no	yes	yes	no	no

Source: SIGG (2008)

Figure 18 presents the characteristics and the prediction methods analyzed by SIGG (2008). According to the author, the most important aspects are: (1) applicability to non-numeric contexts as well as to numeric contexts, (2) the learning ability, (3) and the applicability to multi-dimensional and multi-type discrete time series. Sigg argued that the SOM and Markov approaches are very similar. The main difference between them is that SOM is not applicable to non-numeric contexts, whereas Markov support that aspect. Therefore, Markov was selected for the analytical analysis by Sigg. The Support Vector Machines method (SVM for short) does not support non-numeric context types, neither discrete datasets. The Kalman filter and the ARMA method represent statistical methods and they have quite similar features. Even though they do not support non-numerical datasets, Sigg used ARMA as statistical representative in the analytical tests. The State approach does not support numeric context. And the Alignment sustains non-numeric, numeric, and multi- dimensional contexts. Thus, it also was used in the analytical analysis.

Furthermore, other approaches not listed in Figure 18 were also considered. Neural networks, for instance, have no repository of typical contexts, thus its general operation is similar to ARMA. Evolutionary algorithms were considered as computationally too expensive to be applicable to ubiquitous scenarios. JURSA R. (2006), for instance, employed particle swarm optimization (PSO) methods to prediction. Nevertheless, those techniques are known to quickly collapse to local optima in the search space. Therefore, SIGG (2008) considered that approach as not well-suited for prediction. Moreover, the author also mentioned the importance of adapting simulated annealing to contexts prediction. Other approaches that were not examined were the IPAM algorithm (DAVISON; HIRSH, 1998) as well as graph based models (PATTERSON et al., 2003). The IPAM method was not detailed, because it has low accuracy. And the path learning algorithms are very similar to Markov, thus (SIGG; HASELOFF; DAVID, 2010) used Markov as a representative of the graph based models.

After defined the most suitable methods for prediction (namely Alignment, ARMA, and Markov), SIGG (2008) performed a simulation of them with real data and measured their performance. Although Sigg used ARMA in the analytical tests, we disregarded it for ORACON, because it is applicable only for numerical datasets. The domains Sigg chose for the tests were wind power and location prediction. The attributes measured were prediction accuracy and prediction horizon. In cases where the prediction search space was short and the demanded prediction horizon was high, the Alignment method outperformed Markov and ARMA. On the other hand, for low prediction horizons and high search spaces, ARMA and Markov beat Alignment. Therefore, we can note that as we varying the prediction horizon, the best method also changes. In SIGG; HASELOFF; DAVID (2010), Sigg compared PCA and ICA to Alignment and Markov. Both PCA and ICA had similar accuracy and sometimes they were worse than Alignment or Markov.

Later on, KONIG et al. (2011) enhanced the Alignment approach using correlation among different context sources. Thus, only associated sources are used in the prediction, differently from the original proposal where all sources are employed in the process. Although, the enhanced Alignment method also has a drawback, which was identified by VOIGTMANN; LAU;

DAVID (2011). This disadvantage occurs when entities have a completely new contexts sequence, describing their current actions. The reason for this shortcoming is due to the way the algorithm works. It takes the last contexts sequence of an entity (e.g. the last five contexts) and tries to find in its contexts history that same sequence or the best approximation for it. After the best approximation is found, the contexts following it are used as prediction. Thus, when the entity has a completely new context sequence, the Alignment does not find a good approximation in the history.

Aiming to solve that problem, VOIGTMANN; LAU; DAVID (2011) proposed the Collaboration approach. This method uses the correlated contexts histories of many entities to make prediction for one of them. Thus, in cases where the entity has a completely new contexts sequence, the Collaboration algorithm searches for approximations in the correlated entities' histories. VOIGTMANN; LAU; DAVID (2011) carried out experiments comparing Alignment and the Collaborative approach. The second technique achieved better results than the first method, in cases where the entity had a completely new contexts sequence. Nevertheless, the Collaborative approach also has a drawback. It has to find the exact match of the last contexts sequence in the entities' histories. In other words, the Collaboration method does not use an approximation technique as Alignment does. The own authors recognize the importance of adding this aspect, which they call fuzziness, to their approach VOIGTMANN; LAU; DAVID (2011).

From the discussion held in this section, it is possible to note that there is no single algorithm that is the best for all scenarios. On the contrary, varying the environment's configurations, the most suitable method also changes. In the tests of (SIGG, 2008), we noted that the Alignment went well for high prediction horizon and Markov had more accuracy for low horizons. In KONIG et al. (2011), the Enhanced Alignment beat the standard Alignment in cases where we have information of correlated context. And in VOIGTMANN; LAU; DAVID (2011), the Collaboration was better than Alignment in situations of missing information. Therefore, for this proposal, we consider that the most appropriate approach for a prediction model is an adaptive strategy, employing the algorithm proper to the case.

Although (SIGG, 2008) and SIGG; HASELOFF; DAVID (2010) used Markov model in the analysis, this method has the limitation of not predicting the contexts durations. Therefore, we decided to use Semi-Markov chain, which has this property. In addition, (FOLL; HERRMANN; ROTHERMEL, 2011). also employed this in his approach in his model. In ORACON we used the following methods: Semi-Markov, Alignment, Enhanced Alignment, and Collaboration. The next section describes them in details.

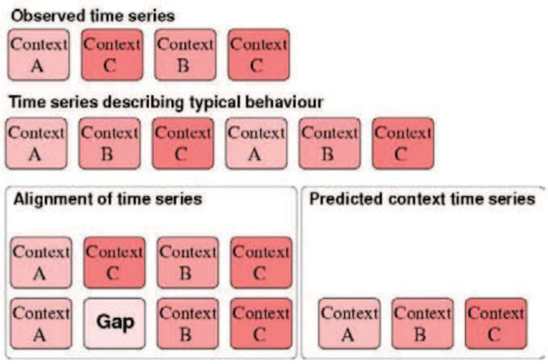
5.2 Description of the Contexts Prediction Algorithms

5.2.1 The Alignment Algorithm

The Alignment algorithm tries to find in two string sequences the most similar sub-sequence by adding gap-symbols to one or the other observed sequence. The method supports multidimensional and multi-type context sequences and has reasonable memory requirements and a scalable error tolerance. The technique is based on the computational biology. In this area, professionals employ it to find approximately matching patterns between RNA or DNA sequences. One of the Alignment's advantages is that it can abstract from process noise in the input sequence. This is possible because the algorithm computes the similarity between observed and typical sequences with a metric that rewards smaller deviations with smaller penalty costs (SIGG; HASELOFF; DAVID, 2010).

The Alignment method requires a set of typical context patterns, which can be created by any pre-processing method (SIGG, 2008). Usually, this pre-processing step is the most important part of the contexts prediction task. An appropriate pre-processing can increase the prediction accuracy and also the prediction speed (KEOGH; PAZZANI, 1998). Typical context patterns need to be extracted from the contexts histories. In this process, characteristic parts of the time series are identified while the noises (irrelevant contexts) are removed. Algorithms for the pre-processing step were described in the works of WEISS; HIRSH (1998); BROCKWELL; DAVIS (2002). This set of typical contexts is named rule base, because it represents the rules that guide the prediction process. In other words, it constitutes the search space (called S) of the Alignment algorithm.

Figure 19: Prediction using the Alignment algorithm



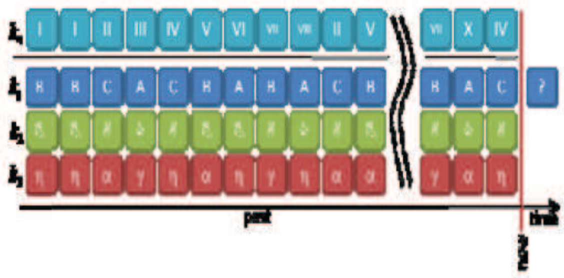
Source: SIGG (2008)

Given a sequence of observed contexts time series elements, the method calculates for every typical contexts time series in S the optimal semi-global alignment that matches the end of the observed time series. The outcome is a set of optimal semi-global alignments. Therefore, Algorithm ranks the semi-global alignments and provides the continuation of the best one as prediction. Figure 19 presents this approach, using one-dimensional contexts time series for ease the understanding.

5.2.2 The Enhanced Alignment Algorithm

In fact, the original Alignment method only makes predictions using one-dimension contexts histories. Nonetheless, in order to handle multidimensional contexts time series, the algorithm merges multidimensional histories into a one-dimensional time series. Then, based on this time series, the algorithm makes the predictions. To convert multi-dimensional data into one-dimension, Alignment divides the history into timeslots. After that, it maps the different symbols, one from each context source, at a certain timeslot and labels them with a new symbol. The mapping itself is made by combining a unique set of symbols, one of each time series, into a new unique symbol. The same mapping is made for the last observed contexts from the different sources. Figure 20 shows the mapping process. The blue, green and red rows k_1 .. . k_3 are the different context patterns in the history, each one of them represents a different context source. On the other hand, the light blue row k_4 consists of the mapped new built unique symbols. Based on this new representation (k_4), the Alignment technique performs the predictions. After that, the predicted symbols are mapped backwards to their original representations.

Figure 20: Mapping process of the Alignment method



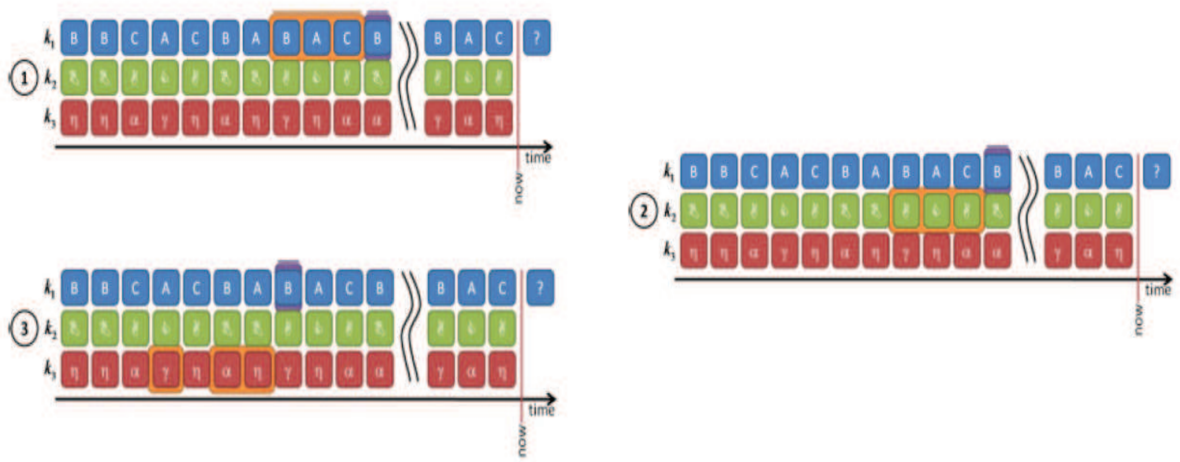
Source: KONIG et al. (2011)

However, KONIG et al. (2011) argued that there is a problem with this approach. According to them, in the mapping process, all errors of the used context sources are preserved. This occurs because the errors result in a new mapped context containing all errors previously dissipated over all context sources. Therefore, to avoid this problem, KONIG et al. (2011) proposed an enhancement in the Alignment algorithm. In this approach, they use the Alignment method to find the place in the history where the last few occurred context have already appeared. Nonetheless, differently from the original Alignment strategy, this new approach does this for every new context source separately. Then, the algorithm takes the following symbols from the context row that should be predicted.

Figure 21 illustrates this technique. In the figure, there are context patterns of three different context sources (k_1, k_2, k_3), and we want to make a prediction for the blue context source (k_1).

Therefore, the algorithm takes the last few observed contexts of all sources. For each one of them, it tries to find the place in the history where they occurred. When the method finds an approximation for the sequence in the history, it takes the following contexts from the context row that should be predicted, in this case k_1. Nonetheless, the main enhancement is in the fact that only correlated context sources are used in this process, differently from the original approach where all sources are assumed to be correlated when mapped into a new symbol. KONIG et al. (2011) gave an analytical proof that their method, in the worst case, performs equals to the original algorithm and in other situations it achieves better results. Additionally, the authors carried out a real-world experiment where the enhancement approach performed better than the original technique. As the enhanced Alignment approach is basically the same as the original proposal, except by the use of correlated context sources, their processing complexity is the same. Furthermore, both can be applied for non-numerical and numerical contexts time series.

Figure 21: The approach of the Enhanced Alignment algorithm to make predictions



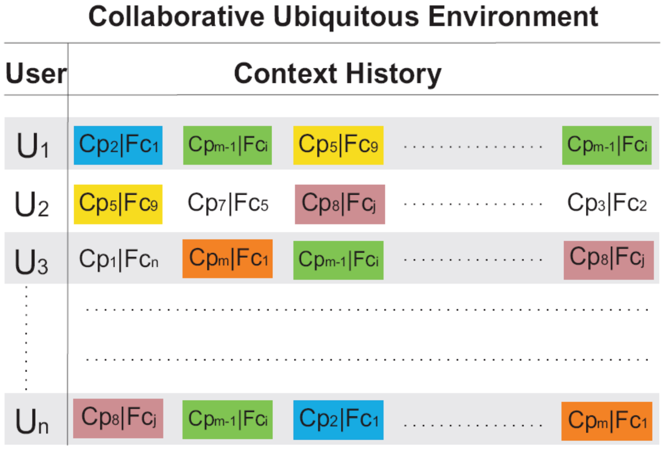
Source: KONIG et al. (2011)

5.2.3 The Collaboration Algorithm

The Collaborative-based Contexts Prediction (CCP) algorithm increases the possibility for making a currently unknown context pattern of a user available to prediction (VOIGTMANN; LAU; DAVID, 2011). It is applicable for both non-numerical and numerical contexts histories. In addition, the algorithm supports multi-types time series. Figure 22 presents the Collaborative Ubiquitous Environment that forms the foundation for the CPP approach. The environment is composed of three different elements, which are: (1) the set of users U $\in$ U; (2) the set of possible context patterns Cp $\in$ CP; and (3) the set of predicted future contexts Fc $\in$ FC. Thus, the history of a user is described by $H_i \subseteq CP$ x FC.

The Collaborative approach uses Higher Order Singular Value Decomposition (HOSVD for short) to enrich the contexts history of a user with additional latent information. Latent infor-

Figure 22: Collaborative Ubiquitous Environment that forms the foundation for the CPP approach



Source: VOIGTMANN; LAU; DAVID (2011)

mation, in this sense, means new parts of the contexts history that were previously and formally unknown and can be used to infer the next future contexts. HOSVD considers existing relations (equal context parts) among the contexts histories of different users in the Collaborative Ubiquitous Environment to obtain the latent data.

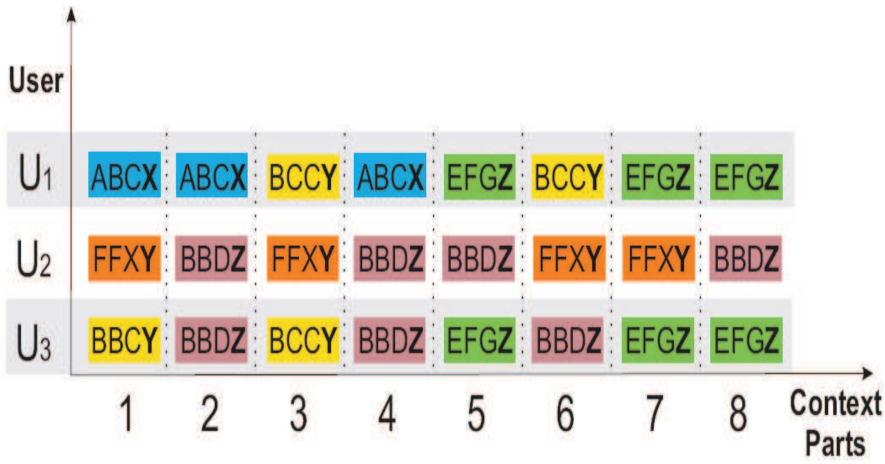
The basic idea of HOSVD is to restrict the dimensionality of each element of a Collaborative Ubiquitous Environment to a specific size in which each element only contains relevant and less noisy information. To achieve this goal, the HOSVD approach uses the n-mode product (KOLDA; BADER, 2009). After that, the downsized information space is used to recalculate the Collaborative Ubiquitous Environment, based on the most relevant information using the n-mode product again.

Moreover, (VOIGTMANN; LAU; DAVID, 2011) provided a practical example on how the Collaboration technique works. In this example, the context patterns are not mapped to multiple future contexts for ease the understanding. Figure 23 presents three users' contexts histories. Each history is divided into eight parts, and each part has an equal window size of four. In addition, equal context parts are marked with the same color. Every context parts consist of three contexts (represented by normal letters) and one future context (characterized by a bold letter). In total, there are five different context parts composed of ten different contexts.

Note that the contexts history of U_1 does not provide information for the context patterns FFX, BBD. In the same way, the history of U_2 does not have data about the patterns ABC, BCC, EFG. Additionally, for the third user U_3, the patterns ABC, FFX have not been recorded yet. Therefore, algorithms, such as Alignment, would have low accuracy making predictions if the last few observed contexts for user U_1 were FFX, BBD. The Collaborative method takes advantage of direct and indirect relations between the users' histories. Direct relations are represented by equal context parts between two users. And an indirect relation between two users (U_1 and U_2) is characterized by a third user U_3 who has similarities with both U_1 and U_2.

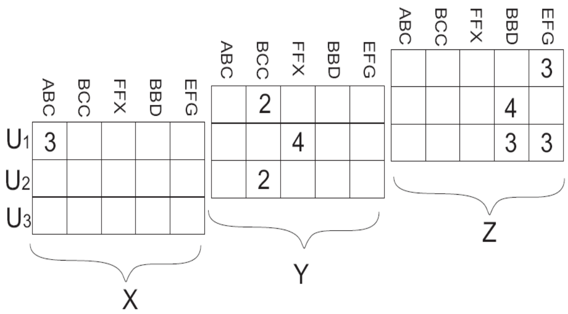
Initially, CCP converts the Collaborative Ubiquitous Environment to a three-order tensor

Figure 23: Contexts histories of three users



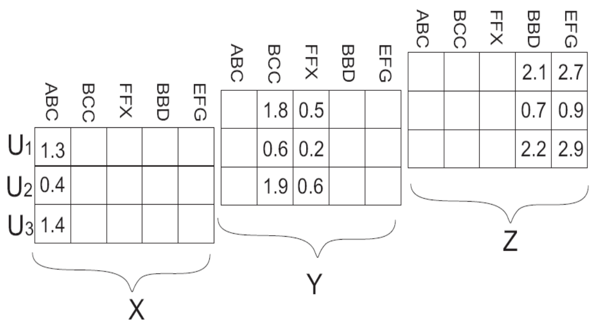
Source: VOIGTMANN; LAU; DAVID (2011)

Figure 24: Three-order tensor **A** of three users, five different context patterns and three different future contexts



Source: VOIGTMANN; LAU; DAVID (2011)

Figure 25: Resulting tensor **A'** with new relations between users, context patterns and future contexts



Source: VOIGTMANN; LAU; DAVID (2011)

structure, described as $A \in \Re^{3x5x3}$ (see Figure 24). In the sequence, the HOSVD is applied to reduce the dimension size of the user to one. Thus, the resulting tensor structure is $\mathbf{A} \in \Re^{1x5x3}$. After that, HOSVD is reapplied to the reduced tensor and new information is obtained, which are shown in Figure 25. This resulting tensor contains new relations among users, context patterns, and predicted contexts. Note that now it is possible to use the new relations provided by the resulted tensor to make a prediction for context pattern FFX or BBD from user U_1. And the same is true for the users U_2 and U_3 regarding patterns that they did not have in the original tensor.

5.2.4 The Semi-Markov Algorithm

Semi-Markov chains are a generalization of Markov and of renewal chains (BARBU; LIMNIOS, 2008). They were independently introduced in 1954 by LéVY (1954), (SMITH, 1955), and TAKACS (1954), who basically proposed the same type of process. Nowadays, Semi-Markov Chains (SMC for short) have become increasingly important in probability and statistical modeling. They are popular for their simplicity and easy applicability to a huge set of problems in various domains. Applications concern queuing theory, reliability and maintenance, survival analysis, performance evaluation, biology, DNA analysis, risk processes, earthquake modeling, etc. A comprehensive description of Semi-Markov Chains is provided by BARBU; LIMNIOS (2008).

SMCs specify a so-called state dwell time, which is an arbitrary probability distribution that is associated with every state transition specifying the amount of time spent in a given state. Formally, a SMC *M* is a 3-tuple defined as:

$$M = (S, p, q)$$

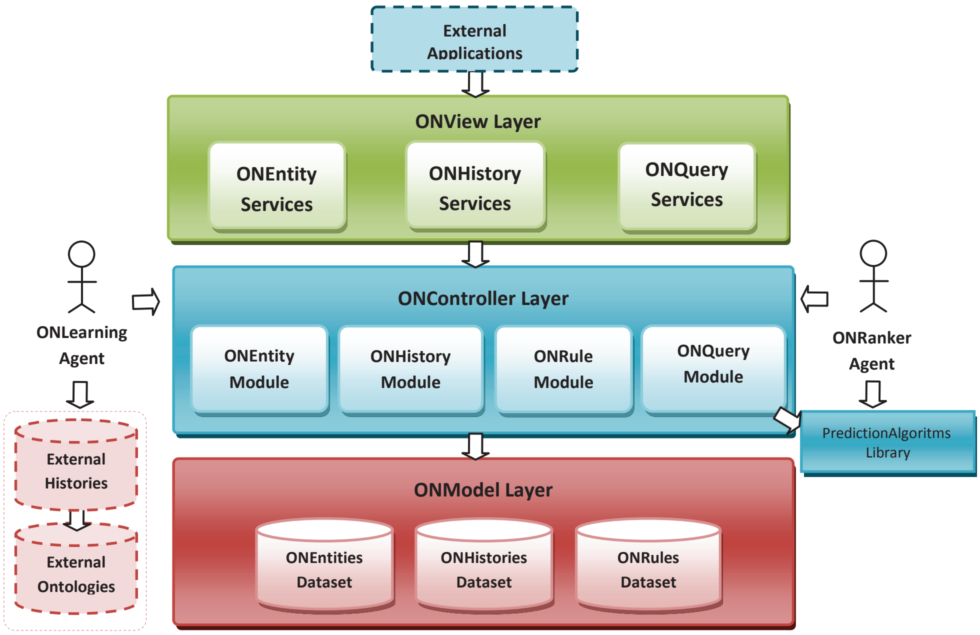
where *S* is the state space, $p : S \times S \rightarrow [0, 1]$ with $\forall s \in S : \sum_{s' \in S} p(s, s') = 1$ is the transition probability function, and $h : (s, s', t) \rightarrow [0, 1]$ with $t \in \Re^{+}$represents the distribution of dwell times associated with a state transition $(s, s') \in S \times S$. The SMC enables us to describe entities' behaviors. At each point in time, an entity is in a state $s \in S$ that is identified by its current context. While the entity acts in the real world, its context changes and its SMC advances to a new state $s' \in S$ representing the new context. s' is called the *successor state* of *s*, and *s'* is visited with a certain probability *p(s, s')*. Before leaving the current state *s*, *s* is active for a limited amount of time (the dwell time described as $h_{s,s'}(t)$). During this time period the entity's context does not change.

6 THE ORACON MODEL

This chapter describes the ORACON model and is divided into seven sections. Section 6.1 presents a general view of the model architecture. Sections 6.2, 6.3, 6.4, 6.5, and 6.6 detail each component of the architecture. And Section 6.7 describes the technologies used in the prototype development.

6.1 The ORACON Architecture

Figure 26: The ORACON Architecture



Source: Made by the author

In this section, we describe the ORACON architecture, which is presented in Figure 26. The architecture is based on the Model-View-Controller (MVC) design pattern. It has three layers, two agents, one library of prediction algorithms, External Histories, External Ontologies, and External Applications. ORACON makes prediction about entities. An entity, in this sense, can be a living being, an object (e.g. a car, a mobile device, an airplane, and even a piece of clothing), or even a location. Each entity can have many applications, modeled as **External Applications**, which can interact with the model in order to obtain predictions.

The entities' contexts histories are represented in the architecture as **External Histories**. ORACON does not approach the obtainment of the histories data. We decided not to approach this characteristic because it is outside the scope of this proposal. In addition, there are many works specifically built for this task, such as, SILVA et al. (2010); ASHLEY (2008); DOHERTY et al. (2011). Therefore, to take advantage of ORACON to obtain predictions, entities need to use an external mechanism to monitor their behaviors and to manage their histories.

Thus, before querying for predictions, the entities need to inform the addresses of the machines keeping their contexts histories, so that ORACON be able to query them to make predictions based on their data. Each entity can have one or many External Histories. To specify their contexts in the histories, entities can use the UbisWorld and the GUMO ontologies represented by the **External Ontologies** component. In addition, it is also possible to use other ontologies to describe domain dependent properties not supported by UbisWorld and GUMO.

The **ONView Layer** has three groups of services that intermediate the communication between the External Applications and the model. The ONEntity services allow the entities to register themselves, their applications and subscriptions in the model. The ONHistory group provides methods to the histories registration. And the ONQuery category enables queries for predictions.

The **ONController Layer** contains the business logic of the model and it is divided into the following modules:

- *ONEntity Module* - it manages the registration of the entities, their applications and subscriptions;
- *ONHistory Module* - it enables entities to register their External Histories and also has methods to access and modify ONHistories Dataset;
- *ONRule Module* - it deals with the creation, reading, updating, and deletion from ONRules Dataset;
- *ONQuery Module* - it processes applications' queries regarding predictions.

The **ONModel Layer** keeps the model information and has these three datasets:

- *ONEntities Dataset* - it keeps the registrations of the entities, their applications and subscriptions;
- *ONHistories Dataset* - it maintains a local copy of the entities' External Histories as well as the conversion of the contexts timeslots to unique numerical identifiers. The conversion of timeslots to numerical identifier is described in Section 5.2.2;
- *ONRules Dataset* - it keeps the rules generated by ONLearning Agent. Rules are used as basis by the algorithms to make predictions.

The **model agents** run autonomously and concurrently to the rest of the model and they are described below:

- *ONLearning Agent* - it constantly monitors the External Histories in order to detect new entries, which are copied to ONHistories Dataset. The agent also generates *rules* or updates the existing ones stored in ONRules Dataset;

- *ONRanker Agent* - it perceives new subscriptions (by analyzing ONEntities Dataset) or changes in the rules (through ONRules Dataset) and ranks for each subscription the best prediction algorithm.

The **PredictionAlgorithms Library** contains the four predictions methods described in Chapter 5, which are: the Alignment, Enhanced Alignment, Semi-Markov, and Collaboration approaches. PredictionAlgorithms Library is used by ONRanker Agent as well as by ONQuery Module. The following sections present each component of the model in details.

6.2 External Histories

The External Histories need to follow a specific standard to represent their data, so that ORACON can understand and process their information. We decided to use the SituationML specification for this purpose (HECKMANN, 2005). SituationML is a markup language designed to describe basically entities' properties (such as, interest, knowledge, motion, activity) and their contexts. The SituationML concepts influenced the way the GUMO and UbisWorld ontologies were designed. However, SituationML can also be used independently from these ontologies. In ORACON, SituationML specifies the structure that the registers from External Histories need to have.

As previously detailed in Section 3.5.3, privacy plays an essential role for a prediction model. The type of privacy regarded here is related to the prediction algorithms that use multiple entities' histories to make prediction for a single one. Note that many entities do not wish to share their entire histories. On the contrary, some of them may want to divide only a certain amount of it or nothing at all.

Therefore, the standard chosen for the contexts history must allow the entities to specify which data they want to share. The SituationML language provides support for that aspect through the group of attributes named *privacy*, see Section 4.1 for a detailed description of SituationML. Using those attributes, more specifically the *access* property, entities can specify for each record of their histories the people or applications that are allowed to use it.

Furthermore, the External Histories also need to support a language for data retrieval, so that ORACON be able to query them. For this task, we used the SituationQL language (HECKMANN, 2005), since it was designed specifically to retrieve data described in the SituationML format. The SituationML and SituationQL languages are detailed in Chapter 4.

6.3 The ONView Layer

The **ONView Layer** has three groups of services (ONEntity, ONHistory, and ONQuery) that intermediate the communication between the External Applications and the model. Below, each subsection describes the general operation of those categories of services. The parameters of the services vary according to the attributes of the datasets presented in Section 6.5. Thus, to avoid duplication of information, we do not specify them in this section.

6.3.1 ONEntity Services

The ONEntity group has four subcategories of services, which deal with the creation, retrieval, update, and deletion of: (1) Entities; (2) Applications; (3) Messages; and (4) Subscriptions. The Entities category has included methods for the entities authentication on the model, so that they can register their application, obtain their messages, and subscribe for predictions.

The subscription for predictions is a mechanism that we created in order to ease the rules generation and algorithms classifications. By subscribing for a prediction, the entity previously manifests its interest in it, thus the model is able to generate rules and compare the algorithms' accuracies. Although it is possible to do that without the subscription, it is more computationally expensive.

For instance, not knowing the context and the time horizon the entity is interested in obtaining predictions, it is necessary to create typical patterns of all sizes containing all contexts for the alignment method. However, if the entity's intention is known, it is possible to generate typical patterns of specific sizes containing the determined context. Notwithstanding, the most difficult part is to classify the best prediction algorithm not knowing the context nor the time horizon the entity wants to have predictions about. As SIGG (2008) analyzed, the most suitable algorithm can vary as the time horizon or the context changes.

In ORACON each application can apply to predictions related to different contexts. In addition, it is possible an entity's application to subscribe for predictions from other entities' contexts. In this case, the model generates a message asking the entities if they allow that applications from other entities obtain prediction regarding them. The messages generated by the model are stored in ONEntities Dataset and can be obtained through ONEntity Services.

6.3.2 ONHistory Services

ONHistory services enable the entities to register their External Histories. As the entities' contexts histories are external to the model, we need to know their information, such as, IP addresses of the machines they are stored and their log in data. Furthermore, ONHistory services also enable the entities to enter correlated histories and contexts. The correlated contexts are needed due to the way the Enhanced Alignment algorithm works. As described in Section 5.2.2, this method uses correlation among different context sources. Therefore, only associated sources are used in the prediction, differently from the original Alignment proposal where all sources are employed in the prediction process.

Nevertheless, we decided not to include in the model the correlation calculation among the different context sources, because of the high computing complexity involved in this process. For example, let us analyze how to calculate this data. To compute correlation, firstly, it is necessary to identify what is a context source. In this work, we used SituationStatements to represent context data. Therefore, the context source is represented by the Predicate attribute of the mainpart category of the SituationStatements.

Each context source determines a set of data, which is composed of the sources' values. For instance, consider Figure 5, presented in Section 4.1, that contains three SituationStatements about an airport scenario. In this example, there are three different context sources, which are: TimePressure, walkingSpeed, and boarding. However, note that only the first two contain information about the same entity, which is Peter. Therefore, to calculate correlation, there are two different set of data, which are: TimePressure, walkingSpeed. Now, it is necessary to obtain the values from these two sets.

The possible values of the first source are: low, medium, and high. And the available values for the second sources are: slow, medium, fast. Thus, before calculating the correlation, it is needed to convert the sources values to numerical representations. Although it seems that this process does not have any problem, it has a serious limitation. This drawback is related to the fact that an ubicomp environment has many context sources. It is not difficult to image a scenario with, for example, ten different context sources. Therefore, to compute the correlation among all these sources, it is necessary calculate correlation 3628800 times (i.e. factorial of 10). And this is to obtain the correlation among the context sources of only one entity, whereas an ubicomp environment can easily have multiples entities interacting.

Moreover, we also decided not to calculate the correlation among the different entities' External Histories, because it occurs the same high-complexity problem that happens with the calculation of the correlation of the entities' context sources. The problem is that in an ubicomp environment there are many entities. Therefore, to compute the correlation among all of them, the processing complexity increases in a factorial degree. Due to those complications, we left outside our model the correlations calculation. Thus, entities need to use an external mechanism to compute that data and then inform it to the model.

6.3.3 ONQuery Services

ONQuery services enable the entities to query the model for predictions. To submit a query, applications need to inform the following attributes:

- *Context* - it specifies the context that the application wants to query for predictions. It is specified in the format of a SituationStatement. If empty, ONQuery Module returns predictions regarding all contexts that might happen within a time limit;

- *Prediction Horizon* - this attribute is used only when the Context is not specified. It describes how much time in the future contexts will be predicted. If empty, the model predicts only one context ahead;

- *Prediction Antecedence* - this attribute is used only when the Context is specified. It represents how much time in advance an application wants to receive a prediction.

The messages returned informing the applications of a prediction is based on the format of SituationML and it presented in Table 5.

Table 5: Description of the prediction message structure

Mainpart	*the basic, rdf-based five-tuple of prediction attributes*
subject	the entity that the prediction is about.
auxiliary	the auxiliary part of the predicate, e.g. "hasProperty" or "hasInterest".
predicate	the dimension or category of the entity, such as, "Walking" or "Working".
range	the range of the object attribute.
object	the value for the subject-auxiliary-predicate triple of the prediction.
Situation	*temporal and spatial values, describing the mainpart of the prediction*
start	the point of time in which this prediction may happen
location	the qualitative spatial location where this prediction may take place.
position	the quantitative spatial location, the exact coordinates.
Explanation	*a collection of explanatory meta attributes*
method	the algorithm used for the prediction.
confidence	a number that displays the probability of the prediction really happen.

Source: Made by the author

6.4 The ONController Layer

The Modules ONEntity, ONHistory, ONRule, and ONQuery are responsible for managing the model datasets, providing basic methods for operations, such as creation, retrieve, update, and deletion, and other specialized functions. Among these modules, the most complex is ONQuery. ONQuery Module processes the applications' queries for predictions. To make a query for predictions, applications do not need to specify in the query the context they are interested. In that case, the next context that might happen in the future will be returned.

Moreover, applications can also make queries related to a specific context. In this case, the model analyzes if there is a subscription (in ONEntities Dataset) regarding that context. If there is not, ORACON tries to make a prediction using the Semi-Markov chain generated for the entity's External History. Otherwise, the model uses the best method ranked for the subscription. In cases that there are no subscription, Semi-Markov chain is used because it does not need subscription to be created, differently from the other methods. If an entity makes the same query more than pre-configured amount of time, ONQuery Module automatically creates a subscription for it, so that the model can choose the most accurate algorithm to make predictions for that case.

6.5 The ONModel Layer

ORACON has three datasets, which maintain the information necessary for the proper operation of the model. Each dataset does not necessarily correspond to a single database; they only represent a clear separation in the data structure of ORACON. Actually, each one of the three datasets can be modeled by a set of tables in a same database. Nevertheless, this discussion is held in Section 6.7. In the following subsections, we describe the information the datasets store.

6.5.1 ONEntities Dataset

ONEntities Dataset stores the entities' registrations, including general information and subscriptions for predictions. This section is divided into two subsections. The first approaches the entity's general information, and the second presents the subscriptions.

6.5.1.1 Entities' General Information

In the entities' general information are included their description, applications, messages generated by the model, and a log of their queries. Table 6 shows the entities' descriptions. Table 7 presents the data structure of the applications. Table 8 describes the entities' messages attributes. And Table 9 groups the attributes of the queries log.

Table 6: Entities' description in ONEntities Dataset

Identifier	it keeps entity's identifier to log in on the model.
Password	it stores the entity's password with the SHA-1 cryptographic hash function applied over it.
Name	it maintains the entity's name.
Type	it keeps the entity's type according to the UbisWorld ontology.

Source: Made by the author

Table 7: Entities' applications in ONEntities Dataset

Entity	it stores the entity who is logged in on the application.
Name	it is the name of the application.
Description	it holds a brief description of the application.

Source: Made by the author

Table 8: Entities' messages in ONEntities Dataset

Entity	it represents the entity which the message belongs to.
Message	it keeps the message.

Source: Made by the author

Table 9: Entities' queries log in ONEntities Dataset

Application	it represents the application through which the query was submitted.
Query	it holds the query sent.
DateTime	it is the date and time the query was made.
Response	it stores the response returned to the application.

Source: Made by the author

6.5.1.2 Entities' Subscriptions

Each subscription is composed of two categories of attributes, which are: *prediction* and *management*. The attributes of the prediction group, presented in Table 10, are specified by the entity and describe the predictions that its applications are interested. The attributes of the management group, described in Table 11, are used by the model to control the processes of learning and algorithms ranking.

6.5.2 The ONHistories Dataset

Each entity can have one or more External Histories. In ONHistories Dataset, we store the access information of the entities' External Histories, e.g. the IP addresses of the machines maintaining them and their log in information. Those attributes are presented in Table 12. Furthermore, ONHistories Dataset stores a copy of the entities' histories as well as information about correlated contexts and correlated histories. Table 13 shows the data of the correlated contexts. Table 14 presents the information of the correlated histories. Table 15 describes the copy of the entity's history of contexts.

6.5.3 The ONRules Dataset

ONRules Dataset contains the rules that ONLearning Agent constructs to guide ONQuery Module and ONRanker Agent. The syntax in which these rules are represented depends on the prediction algorithm. The following subsections describe the rules format for the Alignment, Enhanced Alignment, Semi-Markov, and Collaboration algorithms.

Table 10: *Prediction* attributes of the entities' subscription in ONEntities Dataset

Subscribed Applications	it contains the applications of the entities that are interested in the prediction.
Context	it specifies the context that the application wants to receive prediction about. The context is specified in the format of a SituationStatement, which is divided into five categories. For these specification, it is used only the three first categories, which are: mainpart, situation, and explanation. With this categories, a application can subscribe for prediction related to anything described in the main part group that happens in a certain context (described by the situation category) with determined confidence (supported by the explanation category).
Prediction Antecedence	it describes approximately how much time in advance an application wants to receive a prediction.
Privacy	it specifies if other entities can query the history for predictions. This attribute can assume three values, which are: private - no one, but the owner, can query the history for predictions, public - anyone can query the history for predictions, and restricted - only specified entities can query the history.
Allowed entities	it groups the entities that can query the history for predictions regarding this subscription.

Source: Made by the author

Table 11: *Management* attributes of the entities' subscription in ONEntities Dataset

LastCreatedRule	it represents the last time that an External History was analyzed by ONLearning Agent to generate rules regarding the subscription. This attribute is used by ONLearning Agent to control the creation of rules.
BestRankedAlgorithm	it contains the algorithm the performed better in the tests made by ONRanker Agent. This attribute is used by ONQuery Module to verify which prediction algorithm is the best to process queries related to the subscription.
AccuracyAlignment	it stores the accuracy of the Alignment algorithm to make predictions for this subscription.
AccuracyEnhanced	it keeps the accuracy of the Enhanced Alignment method to predict for this subscription.
AccuracySemiMarkov	it describes the accuracy of the Semi-Markov approach to make predictions for this subscription.
AccuracyCollaboration	it stores the accuracy of the Collaboration algorithm to make predictions for this subscription.

Source: Made by the author

6.5.3.1 Rules for the Alignment and Enhanced Alignment

Rules for Alignment and Enhanced Alignment are a set of typical context patterns, organized in the format of SituationReports. Each SituationReport contains a weight that describes the

Table 12: Entities' histories in ONHistories Dataset

Entity	it is the owner of the External History.
Address	it maintains the IP address and port number of the machine storing the External History.
Identity	it stores identifier of the entity in its External History.
Password	it keeps the entity's password to access the External History.
Privacy	it specifies if other entities can query the history for predictions. This attribute can assume three values, which are: private - no one, but the owner, can query the history for predictions, public - anyone can query the history for predictions, and restricted - only specified entities can query the history.
Allowed entities	it groups the entities that can query the history for predictions. These entities will be able to query the history for any prediction. To allow an entity to obtain only specific prediction, it is necessary to give it permission in the *Allowed entities* attribute of the entity's Subscription.

Source: Made by the author

Table 13: Correlated Contexts in ONHistories Dataset

Entity	it is the entity that informed the correlated contexts.
Context 1	it indicates the context that is correlated to Context 2.
Context 2	it represents the context that is correlated to Context 1.
Correlated History	it stores the contexts on which two histories have correlation.
Correlation	it is a quantitative value that shows the correlation degree.

Source: Made by the author

Table 14: Correlated Histories in ONHistories Dataset

Entity	it is the entity that informed the correlated histories.
History 1	it indicates the history that is correlated to History 2.
History 2	it represents the history that is correlated to History 1.
Correlation	it is a quantitative value that shows the correlation degree.

Source: Made by the author

Table 15: Copy of the entity's history of contexts in ONHistories Dataset

History	it indicates the entity's history from which this copy was made.
Context Unique ID	it contains a unique identifier for each timeslot of the contexts history. The conversion of timeslots to numerical representation is described in Section 5.2.2.
Context	it maintains a context in the SituationStatement format, so that we can store a copy of the entity's External History.

Source: Made by the author

number of times that it was observed. Thus, ONRules keeps a set of SituationReports for each subscription. The size of the rules, in time, varies according to the antecedence attribute set in the subscriptions. The rules have at least the same amount of time of the antecedence value, and it comes before the context that the application subscribed.

The difference between the rules for the alignment and the enhanced alignment is in the conversion of the contexts timeslots to unique identifier. The Alignment uses the unique identifiers generated from all contexts, whereas Enhanced Alignment uses the identifiers created from the correlated contexts, as detailed in Section 5.2.2. Table 16 shows the attributes of the Alignment and Enhanced Alignment methods.

Table 16: Rules of the Alignment and Enhanced Alignment algorithms

Subscription	it refers to the subscription for which the rules was created.
Weight	it represents how many times the rules repeated in the history.
Situation Report	it groups the SituationStatements (i.e. the contexts) that form the rule.

Source: Made by the author

6.5.3.2 Rules for Semi-Markov algorithm

Rules for the Semi-Markov method are, in fact, a Semi-Markov chain of the entity's history, in which the states represent contexts and the transitions are time distribution probabilities. These transition probabilities reflect the frequencies of the corresponding context changes as observed in the contexts history. The constructed Semi-Markov model is stored in ONRules Dataset and it is updated every time the contexts history is updated. Table 17 presents the attributes of the rules for Semi-Markov method in ONRules Dataset.

Table 17: Rules for the Semi-Markov method

History	it represents the history for which the Semi-Markov chain was created for.
Context 1	it keeps one context of the contexts transition.
Context 2	it maintains the second context of the contexts transition.
Time distribution	it stores time intervals that an entity took to go from one context to another and the time intervals probabilities.

Source: Made by the author

6.5.3.3 Rules for the Collaboration algorithm

Table 18 presents the rules attributes of the collaborative approach. Each rule maintains the subscription for which it was created. In addition, the rule contains an entity, a sequence past contexts, and a sequence of future contexts. This structure enables us to store the resulting matrix of the HOSVD, represented in Figure 25 of Section 5.2.3. Thus, to make predictions using the Collaborative method, ORACON tries to match the entity's current contexts with all sequences of past contexts available in the rules. When sequences match, the model uses the predicted contexts as prediction.

Table 18: Rules for Collaborative method

Subscription	it is the subscription for which this rule was created.
Entity	it represents the entity for which the predicted contexts were calculated given the past contexts sequence.
Past Contexts	it contains a sequence of past contexts in the format of a SituationReport. This sequence has size, in time, of the prediction antecedence attribute of the subscription.
Predicted Contexts	it has the sequence of predicted contexts given the past contexts. This sequence has size, in time, of the prediction horizon attribute of the subscription.

Source: Made by the author

6.6 The Model Agents

To model the ORACON agents, we used some parts of the Prometheus methodology (PADGHAM; WINIKOFF, 2004). Figure 27 presents the system overview diagram, in which each agent is described with their perceptions, actions, and messages. ONLearning Agent, for instance, perceives the register of a new entity's history in ONHistories Dataset or a change in a history in the External Histories and manages the rules in ONRules Dataset. ONRanker Agent detects changes in rules in ONRules Dataset and ranks an algorithm for each subscription. This section is organized into two subsections. Section 6.6.1 describes ONLearning Agent with its capabilities, perceptions, and actions. And Section 6.6.2 details ONRanker Agent.

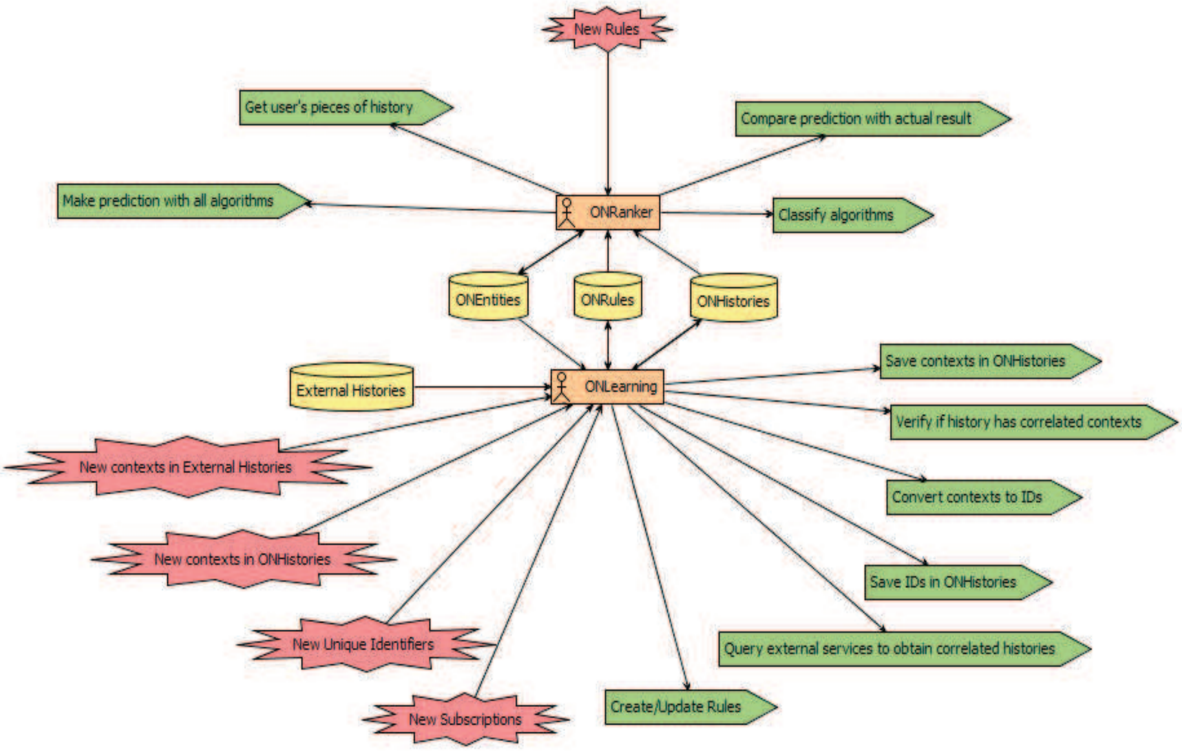
6.6.1 ONLearning Agent

An important aspect for contexts prediction algorithms is the capability to adapt to the changes of the environment (SIGG, 2008). To adjust to those changes, a learning mechanism is necessary. In ORACON model, ONLearning Agent is responsible for the learning task. The agent runs constantly detecting modifications in the entities' External Histories and generating or updating the rules in ONRules Dataset that serves as basis for the prediction algorithms. ONLearning is composed of five capabilities, four perceptions, and six actions, as presented in Figure 28.

The capabilities of the ONLearning Agent are activated by the following perceptions:

- **New contexts in External Histories** - it perceives new entries in the entities' External

Figure 27: Overview Diagram of ORACON Agents



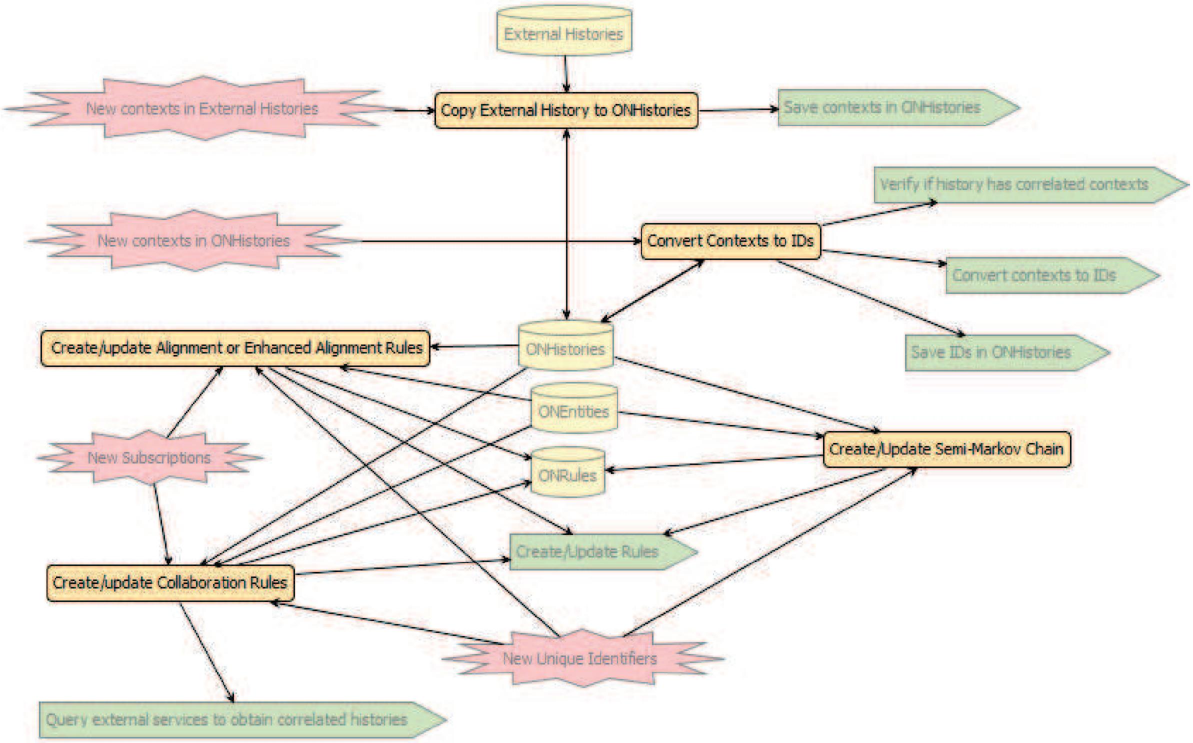
Source: Made by the author

Histories and activates the capability *Copy External History to ONHistories*. To detect new entries, the perception obtains the External Histories addresses from ONHistories Dataset and query them using the SituationQL language, described in Section 4.2. The *fromTime* attribute is set to the last time that that the history was queried. The activated capability gets the new entries and saves them in ONHistories Dataset;

- **New contexts in ONHistories** - this perception realizes the entry of new contexts in ONHistories Dataset and triggers the *Convert Contexts to IDs* capability. This capability verifies if the history has correlated contexts. If it does, the capability converts only the correlated contexts to unique identifiers; otherwise all contexts are converted to the identifiers. Subsequently, it saves the identifier in ONHistories Dataset;

- **New Subscriptions** - it captures new subscriptions in ONEntities Dataset and starts these two capabilities: (1) *Create/update Alignment or Enhanced Alignment Rules*; and (2) *Create/update Collaboration Rules*. Both of them verify if there are contexts with unique identifiers in ONHistories Dataset related to the subscription. If there are not, they stop their executions; otherwise they create rules regarding the subscriptions. The second capability only generates rules if there is information about correlated histories in the model;

Figure 28: ONLearning Agent capabilities



Source: Made by the author

- **New Unique Identifiers** - this perception notices new identifiers in ONHistories and activates these three capabilities: (1) *Create/update Alignment or Enhanced Alignment Rules*; (2) *Create/update Collaboration Rules*; and (3) *Create/Update Semi-Markov Chain*. To detect new identifiers, the perception uses the last time that rules were generated for the subscriptions. This information is kept in ONEntities Dataset together with each subscription in the attribute *LastCreatedRule*. The first and second capabilities check if the new identifiers have the context for which the entity subscribed. If they do not have, nothing is done; on the contrary they create or update their rules. The second capability only manages its rules if there is information about correlated histories in the model. The third capability updates the entities' Semi-Markov chains independently from subscriptions.

The *Create/update Alignment or Enhanced Alignment Rules* capability looks for rules in ONHistories Dataset of the size of the Antecedence attribute of the subscription minus a preconfigured percentage of its value. This percentage has the function of obtaining rules with more time than the Antecedence attribute, so that ONQuery Module be able to make prediction with the proper antecedence time. The contexts sequences with that size are compared to the ones already saved in ONRules Dataset. If they are identical or very similar to contexts patterns in ONRules Dataset, the weight attached to that sequence is strengthened. Otherwise, a new rule is added to the dataset.

Furthermore, the weights of all context patterns in ONRules that are different to the newly observed context pattern are decreased. The increasing and decreasing of weights for typical context patterns enables to track the gradual changes in patterns that account for a changing context evolution process. Old behavior patterns literary fade out in importance, whereas new context patterns are added to ONRules Dataset. In order to control memory consumption, a simple process that removes low-weighted entries from ONRules can be added to the model.

The ability of the model to adapt to a changing environment is called learning frequency (SIGG et al., 2011). It describes how frequently the ORACON screens ONHistories in order to find new rules. With a higher learning frequency, the load processing of ONLearning Agent consequently increases. However, a high learning frequency will more often update ONRules Dataset. A rule dataset that is more frequently updated will in general describe the currently observed and typical context patterns more accurately.

6.6.2 ONRanker Agent

ONRanker Agent is responsible for choosing the best algorithm among the four supported by the model. ORACON algorithms ranking occurs based on the data that is available. For instance, to generate rules for the Alignment method, it is necessary to have subscriptions. Other examples are the Collaboration and Enhanced Alignment approaches, which require information about correlated contexts and histories. Therefore, if we do not have data of subscriptions neither correlated contexts or histories, ORACON applies the only method that does not requires any data to make predictions, which is the Semi-Markov chain.

In cases where the needed data is provided, the model compares the algorithms' accuracies. ORACON does not compare the accuracies of Alignment against Enhanced Alignment, because KONIG et al. (2011) analytically and experimentally proved that Enhanced Alignment performs equals or better than the standard Alignment approach. In Table 19, we present which approaches are compared as the information available changes.

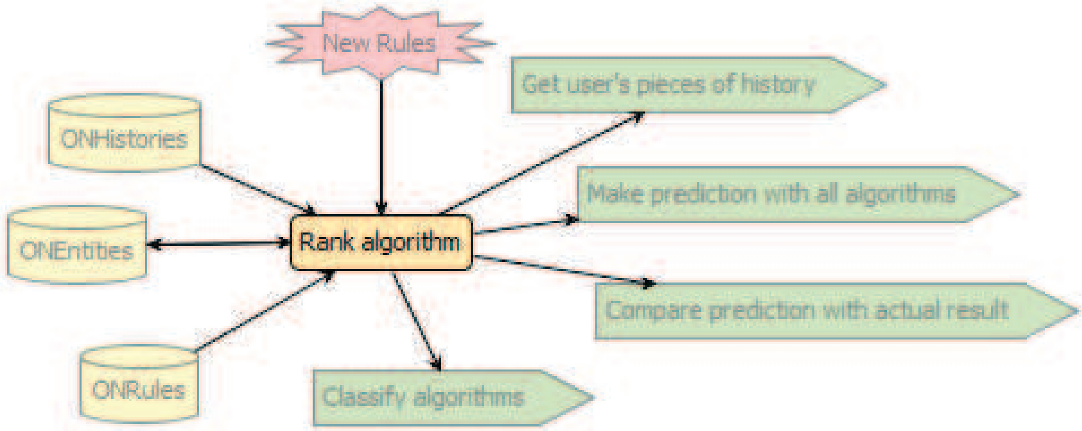
Table 19: Algorithms that are compared according to the information available

Data/Algorithms	**Alignment**	**Enhanced**	**Markov**	**Collaboration**
No subscription			Yes	
Subscription	Yes		Yes	
Subscription and correlated contexts		Yes	Yes	
Subscription and correlated histories	Yes		Yes	Yes
Subscription and correlated histories and contexts		Yes	Yes	Yes

Source: Made by the author

The agent, as presented in Figure 29, is composed of one capacity, one perception, and four actions. Furthermore, it communicates with ONEntities, ONRules, and ONHistories datasets.

Figure 29: ONRanker Agent capability



Source: Made by the author

It also uses the PredictionAlgorithms Library in order to test the different algorithms. The *Rank Algorithm* capacity classifies the best algorithm for a subscription. To rank the best algorithms, the *New Rules* perception detects rules change in ONRules Dataset and starts the capability.

The capability queries the ONHistories Dataset to obtain a pre-configured number of time series (through the *Get user's pieces of history* action). The greater the number of pieces of histories, the more accurate the classifications of the algorithms. However, as we increase the amount of series, the processing time also raises. Therefore, the proper configuration of this attribute will depend on the hardware available and on the amount of entities using the model.

The objective of those pieces of histories is to simulate currently observed contexts of the entity. Nonetheless, in practice, they do not necessarily correspond to the last observed contexts. ONRanker splits each one of the obtained series into two parts, one related to the beginning and other to the end. After that, ONRanker applies all algorithms to make predictions using each one of the beginnings of the series separately (represented by the *Make prediction with all algorithms* action).

Each algorithm returns a different result. The outcome from each algorithm is compared with the end of the series, which corresponds to what really happened in the entity behavior (*Compare prediction with actual result*). The algorithm that best approximates, in average, to what really happened in the real observed contexts is ranked as the best one for the subscription. The information of the best algorithm is placed in the *BestRankedAlgorithm* attribute of the subscription in ONEntities Dataset (*Classify algorithms* action).

The comparison between the real result and the predicted one is made using either the BIAS formula or the alignment algorithm. We use BIAS to compute the precision for low level contexts. And for high level abstractions, we employ alignment. The BIAS calculation is performed by subtracting the real value from the predicted one. This makes sense for low level context where the data follow a defined scale, e.g. GPS, temperature, battery power. In those cases, to subtract the predicted temperature of 35°C from the one really observed of -14°C, would really show how inaccurate the prediction was.

Nevertheless, for high level abstraction, where the numerical identifiers attributed for the context timeslots do not follow a specific scale, that subtraction does not make much sense. In other words, it would not describe the algorithm's precision to subtract the predicted timeslot of identifier 23 from the one that really happened of identifier 43, because those identifiers were randomly chosen. Therefore, for high level contexts, we employ the alignment algorithm. This method tries to align the prediction with the real result. The higher the alignment return, the better the prediction. Both the BIAS and the alignment results are normalized to vary from 0 to 1. Zero means the worse accuracy, and 1 represents the highest accuracy, i.e. that the predicted and the real contexts are equals.

The time series randomly obtained by ONRanker for the tests need to satisfy two conditions. The first one is that they need to be equal, in time, to the *Antecedence* attribute of the subscription. This constraint is due the fact that the rules generated follow the size of the *Antecedence* parameter. The second condition is that the time series correspond as much as possible to the end of the entity's contexts history. This prerequisite is related to the fact that the rules are constantly updated and describe the currently behavior of the entity. Thus, to use time series associated to the beginning of the history can lead to imprecise results.

6.7 Technologies used in the Prototype Development

The model prototype was developed in the Java programming language, using the MySQL relational database. Figure 30 shows the technologies and APIs employed. The Prediction Algorithms Library was implemented through Java classes. For the standard and the enhanced Alignment methods, we used the local alignment Smith-Waterman implementation, available in the NeoBio API[1]. The Semi-Markov approach was completely developed by us. And the Collaborative algorithm was implemented based on the JAMA API[2], which was explored in the Single Value Decomposition calculation.

The ORACON agents were implemented as Java Threads, running periodically in an interval randomly chosen of 30 seconds. The services of ONView Layer were implemented through Webservices, using the JAX-WS API[3]. The modules of the ONController Layer were developed through Java classes. The datasets of the ONModel Layer were modeled in a MySQL database, which is managed using the Hibernate API[4].

The database schema is presented in Figure 31. Note that each dataset was converted into

[1] NeoBio: a JAVA API for sequence alignment algorithms

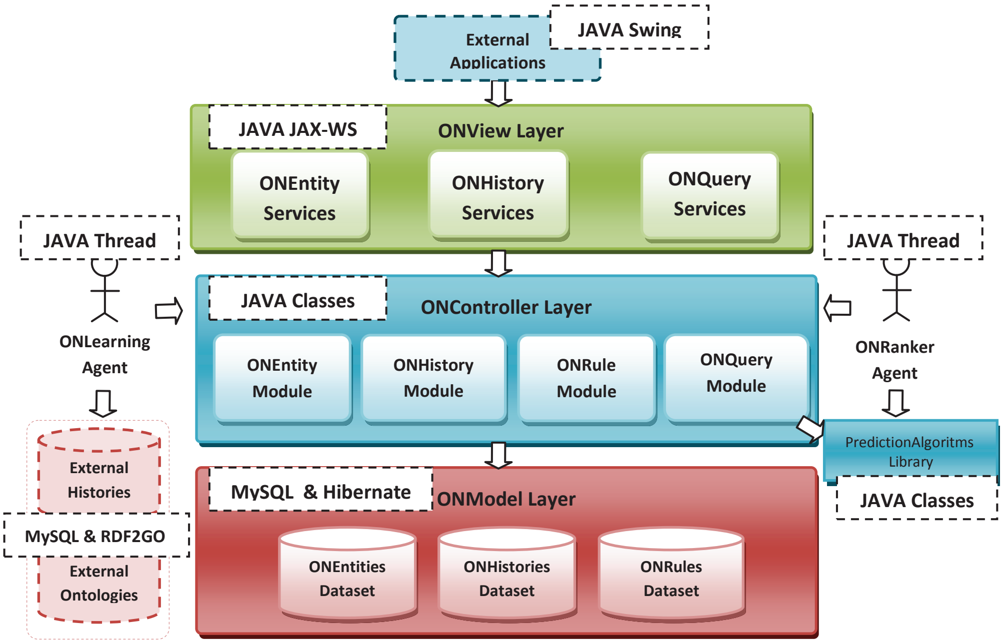
[2] JAMA: A Java Matrix Package, accessed on February 2013, available at: http://math.nist.gov/javanumerics/jama/

[3] Java API for XML Web Services (JAX-WS), accessed on February, 2013, available at: http://jax-ws.java.net

[4] Java API for Relational Persistence, accessed on February, 2013, available at: http://www.hibernate.org/

a set of tables of the database. For example, ONEntities dataset is composed of eleven tables, which are surrounded by a dashed line. ONHistories dataset was mapped to five tables, which are represented inside the dotted line. And ONRules dataset was implemented through six tables, which are shown inside the continuous line.

Figure 30: Technologies used in the model prototype



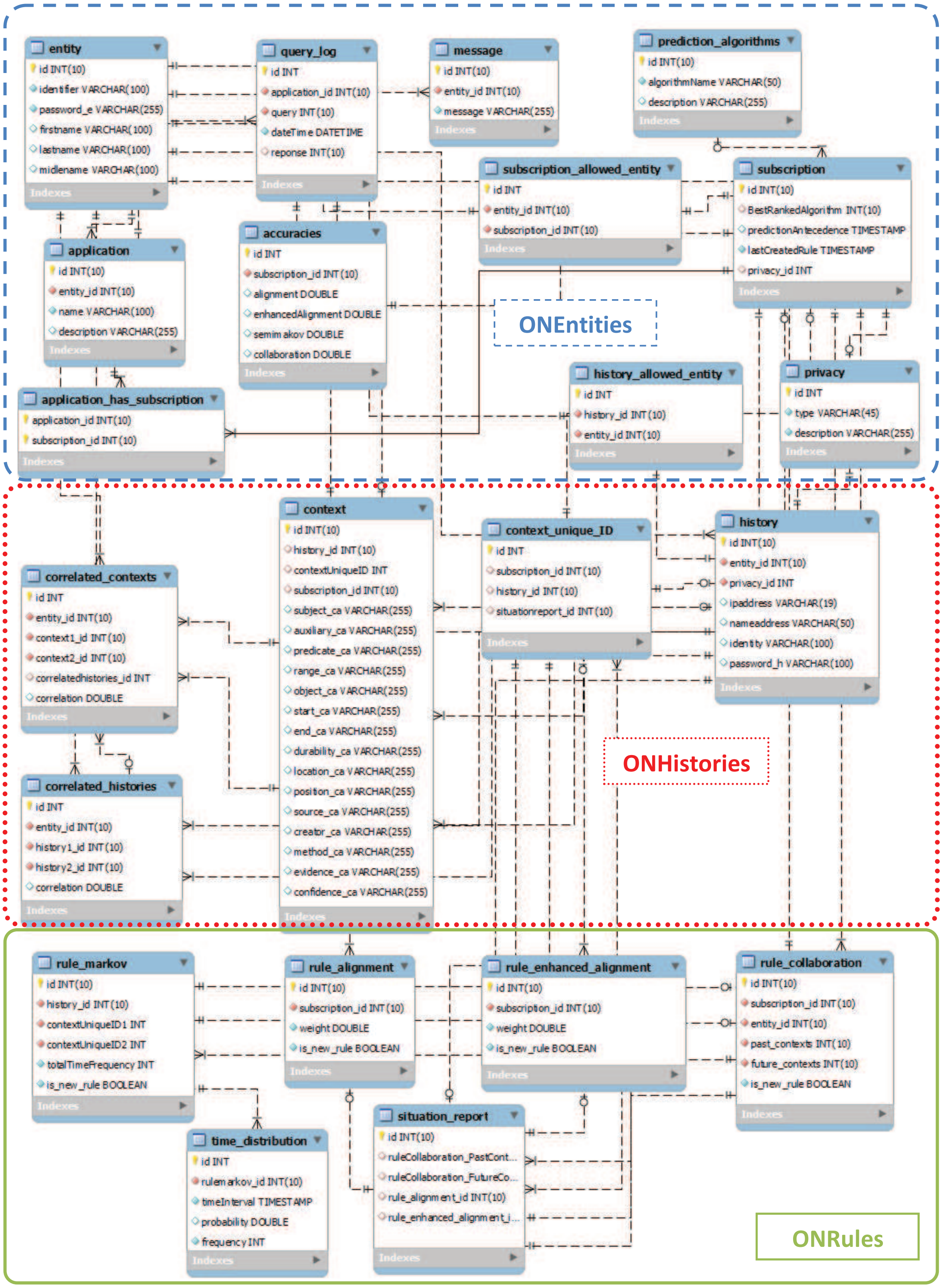
Source: Made by the author

Furthermore, we also created a generic External History, composed of a MySQL database and a service. The service can receive and answer queries in the SituationQL format. And the database contains a table with one column for each attribute of the SituationML format. Thus, all we would need to turn that generic history into a specific one, containing entities' data, is to convert entities' histories to the created database. For this purpose, we implemented a modular converser in Java. This converser can open a text file and save its information in the database. All the user needs to do is to specify, in a HashMap object, the mapping of the file attributes correspond to the SituationML properties.

Nonetheless, we have to remember that the SituationML is decoupled from semantics. And to add semantics to it, we need to use ontologies, such as, GUMO or UbisWorld. Therefore, the user also has to inform, in the HashMap object, the mapping among the SituationML attributes and the ontologies classes. To easy the obtainment of the URIs of the ontologies classes, we converted them to Java classes through RDF2GO API[5]. Thus, the user can use Java classes, in the modular converser, to specify the relationship among the SituationML properties and the ontologies classes.

[5]RDF2GO API, accessed on February, 2013, available at: http://semanticweb.org/wiki/RDF2Go

Figure 31: Modeling of the ORACON Datasets



Source: Made by the author

Moreover, we also implemented a generic External Application, using the Swing API. This application enables to register users, applications, and subscription. It also allows to submit queries and to receive messages. Figure 32 shows screens of the External Application.

7 EVALUATION EXPERIMENTS AND APPLICATION SCENARIOS

In this chapter, we present the experiments conducted to assess ORACON and scenarios in which the model could be used. We performed two tests to evaluate the model. The first one aimed to show all model functionalities. And the second experiment focused specifically on the adaptive feature. Both of the tests were conducted in a local machine using the developed prototype. This chapter is divided into three sections. Section 7.1 deals with experiment one. Section 7.2 describes test two. And Section 7.3 presents the application scenarios.

7.1 Experiment 1: the ORACON functionalities evaluation

For this experiment, we used the freely available location database[6] captured by CHENG et al. (2011). It contains 22 millions checkins across 220,000 users in the location sharing services Foursquare and Twitter. The data is organized in two different files in the comma-separated values format. One of them, named "users_data", contains information about the users' social networks, such as, the number of followers and friends. And the other file, called "checkin_data", has location data, such as, latitude, longitude, and places.

For this test, we only explored the "checkin_data" archive, because we were interested in the users' histories of positions. We used that file to create two external histories. One of them for the user with identifier number 110631929 containing 1459 records, and the other for the user 71305328 having 1254 registers. As the users do not have names in the archive, we gave them the fictitious names Eduardo and João.

The histories were implemented based on the generic External History described in Section 6.7, which is composed of a service and a database table with the columns correspondent to the SituationML attributes. To save the file's information in the database table, we used the modular converser presented in Section 6.7. We specified in the converser the mapping between the archive attributes and the SituationML properties.

Nevertheless, as SituationML is decoupled from semantics, we also needed to establish a link among the SituationML properties and the classes of the UbisWorld ontology. The mapping is described in Table 20. The *Subject* attribute of SituationML corresponds to the *UserID* property in the data file. The *Auxiliary* and *Predicate* attributes of SituationML were mapped to the *HasLocation* and *Location* classes of the UbisWorld ontology. And the *Object* characteristic refers to the *PlaceID* aspect of the file.

Moreover, the *Location, Start,* and *Owner* were linked respectively to the *LatitudeLongitude, CreatedAt,* and *UserID* attributes of the file. After we specified the relationship between the location data file format and the SituationML in the HashMap object of the modular converser, we ran it to fulfill the database table with information. Thus, we generated two different

[6] Location Database, accessed on February, 2013, available at: http://infolab.tamu.edu/static/users/zhiyuan/icwsm_2011.zip

External Histories, for the users Eduardo e João, containing their histories of locations.

Note that we did not use all attributes of the SituationML to describe the users' positions. SituationML is a compressive markup language, containing many different properties, which can be used to detail a variety of information, such as, confidence, privacy, and retention time, among others. However, for this experiment, we used only the attributes shown in Table 20 due two reasons. One of them is because we considered that they were enough to describe users' locations. And the second reason is because we did not have additional data available in the location database explored.

Table 20: Mapping between the SituationML and the location data file

SituationML attributes	**location data file properties**
Subject	UserID
Auxiliary	UbisWorld:HasLocation
Predicate	UbisWorld:Location
Object	PlaceID
Location	Latitude:Longitude
Start	CreatedAt
Owner	UserID

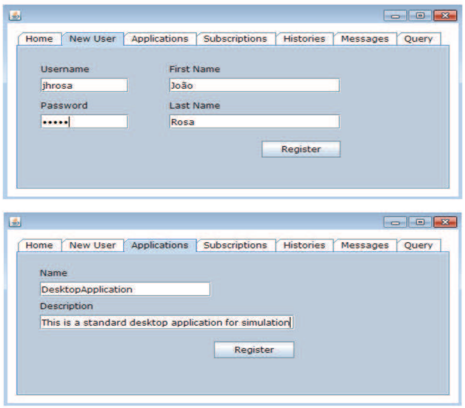
Source: Made by the author

After the histories were created, we started to play the roles of Eduardo and João, as they both were using ORACON to obtain predictions. Thus, Eduardo and João registered themselves, their applications, and their External Histories in the model through the External Application developed, see Figure 32a. Both registered the same application, which was the desktop application implemented. The external histories information they entered was: IP address - 127.0.0.1, port number - 8181, identifier - the users' names, and password - 1234. The histories had that IP address because the experiment was conducted in a local machine.

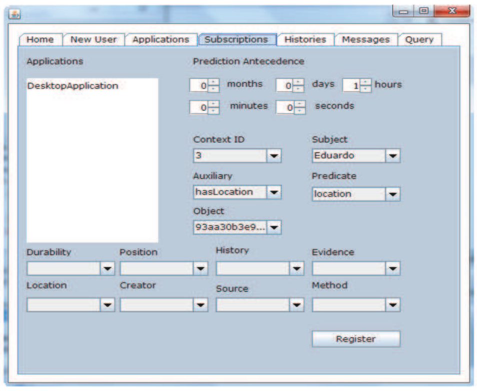
Subsequently, João subscribed for predictions regarding the position of Eduardo, which is shown in Figure 32b. As the subscription corresponded to other user's context, the model had to ask the user's permission. Therefore, ORACON sent a message to Eduardo requesting authorization for João to query for predictions about his context, see Figure 32c. Eduardo received the message and allowed João to obtain predictions. As you may have noticed, the model dealt with the **privacy** of Eduardo by asking him either he allowed or not Joao to have his predictions.

After awhile, ONLearning Agent detected new contexts in the users' histories, through the *New contexts in External Histories* perception, and started the *Copy External History to ON-Histories* capability. This capability set a SituationQuery, as presented in Figure 33a, and sent it to the External History. The history then processed it and returned a SituationReport, shown in Figure 33b. This report contained the users' contexts in the format of SituationML. Thus, the capability concluded its execution by storing that SituationReport in ONHistories Dataset.

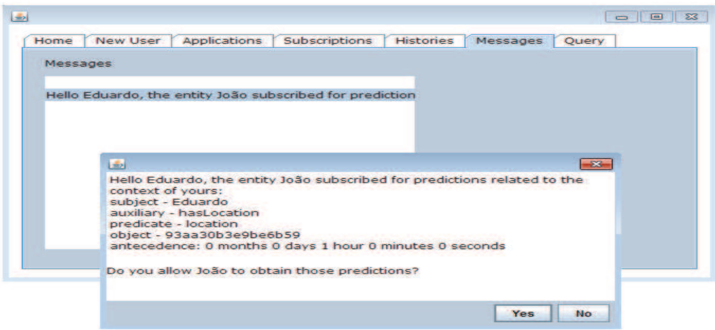
Figure 32: Registers in the External Application



(a) Entities' Register



(b) João's subscription



(c) Privacy Message

Source: Made by the author

In this process, we can see in operation the **context formal representation** used in the model. Moreover, other important fact to highlight is that the External History only returned Eduardo's statements that were set as public or had João in the allowed list. Even though Eduardo had allowed João to query for his context in ORACON, the context representation we chose also allows him to limit individually the statements he wants to share. In other words, the chosen format SituationML enables the user to control the **privacy** of their histories. This can be done through the privacy group of attributes of the SituationML standard.

Subsequent, the *New contexts in ONHistories* perception realized that context storing and triggered the *Convert Contexts to IDs* capability, which converted the contexts to numerical identifiers and saved them in ONHistories Dataset. Thus, the *New Unique Identifiers* perception started the rules creation for the Semi-Markov and Alignment algorithms by triggering the

Figure 33: SituationQuery and SituationML


```xml
<SituationRequest>
    <query>
        <owner> Eduardo </owner>
        <identity> 110631929:1234 </identity>
        <fromTime> 0 </fromTime>
    </query>
</SituationRequest>
```


(a) Query sent to Eduardo's External History


```xml
<SituationReport>
    <SituationStatement>
        <subject> Eduardo </subject>
        <auxiliary> http://ubisworld.com/hasLocation </auxiliary>
        <predicate> http://ubisworld.com/location </auxiliary>
        <object> 93aa30b3e9be6b59 </object>
        <location> -6.165.915:106.728.730 </location>
        <start> 19/12/2010 08:17 </start>
        <owner> Eduardo </owner>
    </SituationStatement>

    <SituationStatement> ... </SituationStatement>
    ...
</SituationReport>
```


(b) Response from Eduardo's External History

Source: Made by the author

capabilities *Create/update Alignment or Enhanced Alignment Rule* and *Create/Update Semi-Markov Chain*. As ONLearning Agent runs periodically, whenever new context are inserted in the user's history, they are copied to ONHistories dataset, converted to identifiers, and used to update the rules. That process is what we call **learning** mechanism in ORACON.

As soon as the rules were created, the *New Rules* perception initiated the *Classify algorithms* capability. That capability divided the sequence of contexts numerical identifiers of Eduardo into 10 pieces of the size of the subscription antecedence attribute plus a percentage value of 10 randomly chosen. After that, it split each one of the pieces into two parts, one related to the beginning and other to the end. The *Make prediction with all algorithms* action applied all algorithms using each one of the beginnings of the series separately.

Each algorithm returned a different prediction result. The *Compare prediction with actual result* action compared those outcomes with the end of the series, which corresponded to what really happened in the user's behavior. The comparison was performed using the Alignment algorithm as described in Section 6.6.2. In the results, the Alignment method obtained an average accuracy of 0.63 and the Semi-Markov approach of 0.72. Therefore, the *Classify algorithms* capability ranked the Semi-Markov chain as the most suitable method to make prediction for that subscription.

Thus, whenever João queries ORACON for predictions related to that context with that antecedence configuration, the model will use Semi-Markov method. However, it is important to highlight that as we vary the antecedence of that same subscription, the best classified algorithm may change. To analyze that, we ranked the methods for the same subscription with other antecedence configurations. We tested with the values 2, 3 and 4 hours. For all those values, the Alignment approach was chosen as most suitable, returning accuracies of 0.54, 0.56, and 0.49 against the 0.47, 0.40, and 0.33 of the Semi-Markov approach.

As you may have noticed, ORACON ranked the best method to make predictions for João's subscription. We also showed that as we changed the antecedence, the best algorithm also varied. ORACON constantly analyzes history changes or register of new subscriptions and classifies the best method for subscriptions. Thus, the model **adapts** itself in order to apply the best algorithm to the situation. Although in this experiment, the predictions were made using high-level contexts, which are the position identifiers, ORACON supports the lowest to the highest **context abstraction level**. Actually, the model does not make any distinction between low or high context levels. ORACON makes predictions based on the External Histories data, independently from their abstraction levels

7.2 Experiment 2: assessment of the adaptive feature

As argued in Section 6.3.2, entities that wish that ORACON chooses the most suitable method for their predictions, need to inform their interested contexts through subscriptions. That ensures that ORACON will analyze and rank the Alignment and Semi-Markov algorithms. Nevertheless, if the entities want to include the other two methods supported, they need to provide data about correlated contexts and histories. In cases where that information is available, all approaches are compared.

In this section, we describe an environment containing all necessary data for ONRanker Agent to choose one of the predictions methods. The experiment focuses specifically on evaluating the adaptive aspect. We used contexts histories generated through the Siafu tool MARTIN; NURMI (2006). Siafu was designed as a means to generate data for the evaluation and the comparison of different machine learning methods in mobile context-aware settings. In addition, it provides many simulation scenarios[7], including the day-to-day living on an office, a university campus, and even a small town. For this test, we used the Leimen scenario, which simulates people's life on the small town from Germany called Leimen.

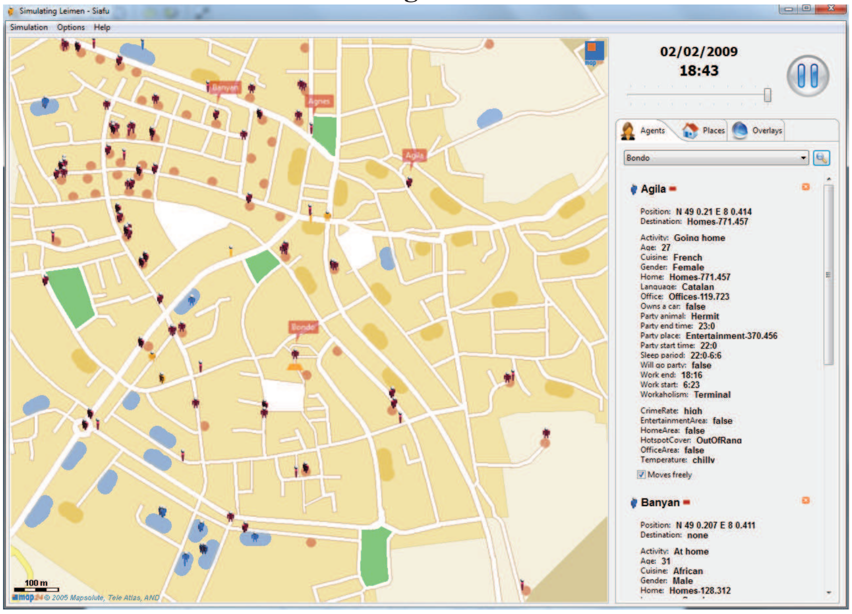
We chose that environment because it contains many users doing different activities. For example, in the morning people wake up and go to their offices (sometimes by car), then either go home, or go party, returning home early sometimes. Next day, they go work again. For each place the users visit, Siafu generates information about crime level, temperature, activity, and wireless coverage. That data compose the users' contexts together with their positions and time. Siafu generates an output file in the comma-separated values format.

We simulated the behavior of 30 users for a period of 30 days. Figure 34 shows the simulation in operation. Then, we obtained users with correlated contexts and histories. We computed that data separately using Excel. We obtained three users, named Agila, Banyan, and Bondo, with the position and the crime level correlated. We used the Pearson's correlation coefficient, which is a measure of linear dependence between two variables X and Y, giving a value between +1 and -1 (RODGERS; NICEWANDER, 1988). Agila obtained correlation of 0.97, Banyan of -0.91, and Bondo of 0.92, indicating that the variables are strongly correlated.

Notwithstanding, we needed to obtain other users having histories correlated with the his-

[7] Siafu - An Open Source Context Simulator, accessed on February 2013, available at: http://siafusimulator.org

Figure 34: Leimen Simulation in Siafu



Source: Made by the author

tories of Agila, Banyan, and Bondo. That way, we would have three users with all information necessary for the model to rank the prediction algorithms. Therefore, to ease the correlation calculation among the histories, we used the *Convert Contexts to IDs* capability of ORACON, because it can convert the contexts to numerical identifiers. However, before doing that, we had to transform the user's data generated by Siafu to the SituationML format, so that we could apply the capability.

Thus, we created an External History for each user of the simulation. To generate the histories, we used the generic External History and the converser implemented, which are described in Section 6.7. To use the converser, we specified the mapping between the attributes of the simulation output file and the SituationML properties. Nevertheless, as SituationML is decoupled from semantics, we also needed to establish a link among the SituationML attributes and the classes of the UbisWorld ontology. The mapping is presented in Table 21.

As we can see, the table has four rows. The first one shows how we represent the user's activities through SituationML. And the three other lines present how we describe the characteristics of the position. We specified the user's activities by mapping the *Subject* property of SimulationML to *EntityID* of the simulation, the *Auxiliary* to *HasDone* of UbisWorld, the *Predicate* to *ActionalActivities* of UbisWorld, and the *Object* to *Activity* of the simulation.

The characteristics of the position that we described were: temperature, hotspot coverage, and crime rate. To detail those aspects, we used the *Subject* configured to *Position* attribute of the simulation, the *Auxiliary* to *Has* of UbisWorld, the *Predicate* to *Temperature, CrimeRate, and Hotspot* of UbisWorld, and the *Object* to *Temperature, CrimeRate, HotspotCover* of the simulation.

After the mapping was concluded, we ran the converser, thus generating an External History

Table 21: Mapping between the SituationML and the the Leimen simulation output file

Subject	Auxiliary	Predicate	Object	Position	Start	Owner
entityID	UbisWorld:HasDone	UbisWorld:ActionalElements	Activity	position	time	entityID
position	UbisWorld:Has	UbisWorld:Temperature	Temperature	position	time	entityID
position	UbisWorld:Has	UbisWorld:CrimeRate	CrimeRate	position	time	entityID
position	UbisWorld:Has	UbisWorld:HotspotCover	HotspotCover	position	time	entityID

Source: Made by the author

for each user in the simulation. Then, we converted the users' contexts to numerical identifiers by employing the *Convert Contexts to IDs* capability of ONLearning Agent. That gave us sequences of numbers representing the histories. Thus, we used those sequences to compute the correlation of all histories with the sequences of Agila, Banyan, and Bondo.

We used the Pearson's correlation coefficient (RODGERS; NICEWANDER, 1988) and obtained three users with histories correlated to the Agila, Banyan, and Bondo. For Agila's history we found Ailan's history with 0.76 correlation coefficient. For the Banyan, we identified Agila with correlation of -0.87. And for Bondo, we found Banyan with correlation of -0.74. All users had moderate correlation, which we considered acceptable for the test.

Table 22: Algorithms' accuracies for the users' subscriptions

Subscriptions/Accuracies	Semi-Markov	Enhanced Alignment	Collaboration
Agila, antecedence of 15 min	0.82	0.61	0.54
Agila, antecedence of 45 min	0.53	0.58	0.63
Agila, antecedence of 2 h	0.35	0.52	0.45
Banyan, antecedence of 45 min	0.46	0.65	0.70
Banyan, antecedence of 2 hours	0.41	0.71	0.76
Banyan, antecedence of 3 hours	0.37	0.58	0.68
Bondo, antecedence of 45 min	0.50	0.63	0.56
Bondo, antecedence of 4 hours	0.26	0.57	0.50
Bondo, antecedence of 5 hours	0.14	0.46	0.54

Source: Made by the author

Now that we had all correlation information that we needed, we registered the users in the model, using the External Application developed. After that, we registered their applications, subscriptions, and correlated histories and contexts. We created three subscriptions for each one of the three users. For Agila, we created subscriptions for the home location with antecedences 15 minutes, 45 minutes, and 2 hours. For Banyan we used office location with antecedences 45 minutes, 2 hours, and 3 hours. And for Bondo we subscribed for nightclub location with antecedences 45 minutes, 3 hours, and 4 hours.

Thus, ORACON had all data needed to choose the best prediction algorithm for the contexts. Then, the model generated rules for all algorithms and classified the best method for each subscription. Table 22 shows the algorithms' accuracies for the users' subscriptions. For Agila, the Semi-Markov approach had best accuracy for the antecedence of 15 minutes. However, for the antecedences of 45 minutes and 2 hours, the Collaboration and Enhanced Alignment methods had respectively best precisions.

For Banyan, the Collaboration algorithm was classified as most suitable for all antecedences. A fact that may have influenced on the Collaboration supremacy was the high correlation of Banyan and Agila histories. Furthermore, it is interesting to highlight that the Enhanced Alignment won second place, beating Semi-Markov approach in all case. That might have happened because the prediction horizons set for Banyan were relatively high. And as already described in other studies, Semi-Markov usually performs better for low prediction horizons (SIGG, 2008). This can be noted in Agila's subscription with antecedence of 15 minutes.

For Bondo, Semi-Markov lost for the other two algorithms for all antecedences configurations. In this case, the antecedences were again relatively high. For 45 minutes and 4 hours, the Enhanced Alignment had best accuracy. Nonetheless, for 5 hours, the Collaboration algorithm beat Enhanced Alignment. Analyzing those results, we can note that there was not a method most suitable for all situations. On the contrary, as we varied the prediction horizons of the users' subscriptions, the most accurate algorithms also changed.

ORACON tested all methods for the subscriptions and chose the most accurate for each situation. Thus, whenever the users submit a query related to the same configuration of their subscriptions, the model will use the best classified algorithm. Moreover, it is important to emphasize that the ORACON constantly analyzes history changes or register of new subscriptions. And when it detects one these events, it reclassifies the best method for subscriptions. Thus, the model adapts itself in order to apply the best algorithm to the situation.

7.3 Application Scenarios for ORACON

As you may have noted, both of the experiments were based on predicting users' locations. The main reason for that was that we only found location database to perform the tests. However, in this section, we describe some scenarios where ORACON could be applied. ORACON would be useful in a diverse range of areas. For example, it could be employed to make predictions regarding u-health (ELGAZZAR et al., 2012), u-city (SHIN, 2009), u-commerce (FRANCO et al., 2011), u-learning (BARBOSA et al., 2011), and context-aware competences management (ROSA et al., 2011).

In the u-commerce field, ORACON could be used to make predictions about business opportunities. For instance, consider an u-commerce model, such as MUCS (FRANCO et al., 2011). MUCS is able to find deals possibilities among people in the same context. Nevertheless, if the users are not in the same context at the same time, the model is not capable of discovering the opportunities. Thus, we could employ ORACON to overcome this limitation. For example, using ORACON, people could query certain contexts to find out if they will have chance of closing some deal in it.

That same idea could be explored in the u-learning and competence management areas. In these fields, we have models, such as, LOCAL (BARBOSA et al., 2011) and DECOM (ROSA et al., 2011). LOCAL helps users to find learning possibilities, by forming interest groups and suggesting learning objects according to the users' contexts. And DECOM assists people to advance in their competences by recommending interactions with other people who are in the same context and have higher competence level. As we can see, these models have the same drawback that MUCS does, i.e. they cannot find learning opportunities if the users are not in the same context at the same time. Therefore, ORACON could also be applied to them. Thus, the users would be able to query contexts to find if they have any chance of finding learning opportunities.

Moreover, ORACON could also be explored in fields, such as, u-health and u-city. In u-health we can imagine an application that monitors the users' daily habities, such as, eating, sleeping, and exercising, among others. Such application could use ORACON to predict problems related to high stress level, high blood pressure, and allergic attack. In the u-city area, we could use the model to predict assaults, car accident, or energy and water consumption.

8 FINAL CONSIDERATIONS AND FUTURE WORK

In this thesis, we described and compared four related works. The works were analyzed according to five aspects considered important for contexts prediction models, which are: (1) Adaptive Approach; (2) Context Formal Representation; (3) Privacy; (4) Low and high context levels; and (5) Learning Capability. Now, the ORACON model is analyzed considering these characteristics. Table 23 presents ORACON together with the studied related works.

Table 23: Comparison of ORACON with the related works

	Adaptive Approach	Context Formal Representation	Privacy	Low and high context levels	Learning Capability	Data Obtainment
Mayrhofer's model	No	No	No	No	Yes	Yes
Sigg's architecture	No	No	No	Yes	Yes	Yes
Structured architecture	Manual	No	No	Yes	Yes	Yes
PreCon model	No	No	No	Yes	Yes	Yes
ORACON	Automatic	Yes	Yes	Yes	Yes	No

Source: Made by the author

Different from all studied works, ORACON can **adapt** itself in order to apply the best algorithm to the situation. To do so, the model constantly tests all algorithms supported and compares their results to identify which one is the best to make a particular prediction. This approach is different from the Structured Contexts Prediction architecture (MEINERS; ZAPLATA; LAMERSDORF, 2010), which is manual, that is; the designer needs to choose at design time the most suitable algorithms for predictions. ORACON adaptive approach decides automatically and in run time the best method to make a particular prediction.

Furthermore, ORACON is context-aware and empowers the implementation of ubiquitous computing concepts. The model supports common aspects of ubicomp, such as, context formal representation and privacy. To support **context representation**, we studied the context modeling and ubiquitous user modeling areas and identified the most complete work, which was the study of HECKMANN (2005).

That work was chosen due to its ontological approach and the decoupling from syntax and semantic, as detailed in Chapter 4. Thus, we employed the languages and ontologies of HECKMANN (2005) in ORACON, so that applications interested in taking advantage of the model need to support those standards. Moreover, we also defined standards to query ORACON for predictions and to answer those queries, which are based on the SituationML, and described in Section 6.3.3.

Other characteristic regarding **privacy** discussed in this work is related to which applications and entities will have access to the predictions made for a specific entity. ORACON support this aspect by enabling entities to specify which entities can obtain predictions regarding them. In addition, through the standards chosen, we ensure that only public data of the entities' histories are used by the Collaboration algorithm (VOIGTMANN; LAU; DAVID, 2011) to make prediction for one entity based on many entities' histories.

ORACON also supports contexts prediction from the **lowest to the highest context abstraction level**. In fact, the model does not differ between low and high context abstraction degrees. ORACON makes predictions based on the External Histories data, independently from their abstraction levels. The **learning mechanism** is supported through the ONLearning Agent, which constantly monitors the entities' histories and updates the algorithms' rules, which serve as basis to make predictions.

Moreover, a new comparison item was included in the comparison table, which is: **data obtainment**. This item describes if the prediction model supports the obtainment of the entities' histories. As we can see, ORACON does not approach that characteristic. We decided not to approach that aspect because it is too complex and outside the scope of this proposal. In addition, there are many works specifically built for this task, such as, SILVA et al. (2010); ASHLEY (2008); DOHERTY et al. (2011). Therefore, we decided to focus our efforts on the prediction itself rather than on monitoring entities and collecting their information. We understand that task should be done by an external and specialized model, such as, the approaches of SILVA et al. (2010); ASHLEY (2008).

Contexts prediction has been receiving considerable attention in the last years (MEINERS; ZAPLATA; LAMERSDORF, 2010; FOLL; HERRMANN; ROTHERMEL, 2011; KONIG et al., 2011; VOIGTMANN; LAU; DAVID, 2011). Furthermore, it important to highlight that this area seems to be the next logical step in context-aware computing, which, until a few years ago, had been concerned more with the present and the past temporal dimensions. Therefore, the area has a lot to improve, as well as has ORACON. The model consists of an initial proposal, which can receive many enhancements in form of future works. For instance, a new module, responsible for calculating the correlation among context sources and contexts histories, could be added to ORACON. Nevertheless, this module would need to have a heuristic to overcome the high complexity problem of this task.

Furthermore, it would be extremely important to analyze the model's performance, mainly of the agents, in real and simulated environments. There are two principal aspects that can have great impact on the performance. The first is the number of the pieces of histories that ONRanker agent uses to compute the algorithms' accuracies. As detailed in Section 6.6.2, the greater the number of pieces of histories, the more accurate the classifications of the algorithms. However, as we increase the amount of pieces, the processing time also raises. Therefore, it would be relevant to analyze how the agent's time processing increases according to the amount of pieces of histories.

The second characteristic that could be tested is the learning frequency of ONLearning agent. As discussed in Section 6.6.1, the higher the learning frequency, the more often the ONRules Dataset will be updated. And a rule dataset that is more frequently updated will in general result in more accurate prediction. However, that makes the load processing of ONLearning Agent to rise. Thus, it would also be important to analyze how the learning frequency impacts on ONLearning agent performance.

REFERENCES

ALFRED, K. Generic user modeling systems. **User Modelling and User-Adapted Interaction Journal**, [S.l.], v. 11, p. 49–63, 2001.

ANDERSON, J. **Cognitive psychology and its implications**. [S.l.]: 3 edn. Spektrum, 2001.

ARGILAGA, M.; JONSSON, G. Detection of Real-Time Patterns in Sports: interactions in football. In: MEETING OF COMPLEX SYSTEMS AND SPORT & 4TH INTERNATIONAL CONFERENCE OF COMPUTER SCIENCE IN SPORT - COM & COM. INEFC, BARCELONA, 1., 2003. **Anais. . .** [S.l.: s.n.], 2003. p. 14 – 17.

ASHLEY, S. **Who Controls the Past Controls the Future - Life Annotation in Principle and Practice**. [S.l.]: University of Southampton, School of Electronics and Computer Science, Doctoral Thesis., 2008.

BAIER, C.; KATOEN, J.-P. **Principles of Model Checking (Representation and Mind Series)**. [S.l.]: The MIT Press, 2008.

BALDAUF M., D. S. . R. F. A survey on context-aware systems. In: INTERNATIONAL JOURNAL OF AD HOC AND UBIQUITOUS COMPUTING., 2007. **Anais. . .** [S.l.: s.n.], 2007. p. 263–277.

BARBOSA J. L. V., H. R. M. R. S. A. . B. D. N. F. Content Distribution in Context-Aware Environments. In: XIII BRAZILIAN SYMPOSIUM ON MULTIMEDIA AND THE WEB (WEBMEDIA), 2007. **Anais. . .** [S.l.: s.n.], 2007. p. 1–8.

BARBOSA, J. L. V.; HAHN, R. M.; BARBOSA, D. N. F.; SACCOL, A. I. d. C. Z. A ubiquitous learning model focused on learner interaction. **Int. J. Learn. Technol.**, Inderscience Publishers, Geneva, SWITZERLAND, v. 6, n. 1, p. 62–83, May 2011.

BARBU, V.; LIMNIOS, N. **Semi-Markov Chains and Hidden Semi-Markov Models toward Applications**: their use in reliability and dna analysis. 1. ed. [S.l.]: Springer Publishing Company, Incorporated, 2008.

BAUR, D.; SEIFFERT, F.; SEDLMAIR, M.; BORING, S. The Streams of Our Lives: visualizing listening histories in context. **Visualization and Computer Graphics, IEEE Transactions on**, [S.l.], v. 16, n. 6, p. 1119 –1128, nov.-dec. 2010.

BELIMPASAKIS, P.; ROIMELA, K.; YOU, Y. Experience Explorer: a life-logging platform based on mobile context collection. In: THIRD INTERNATIONAL CONFERENCE ON NEXT GENERATION MOBILE APPLICATIONS, SERVICES AND TECHNOLOGIES, 2009., 2009, Washington, DC, USA. **Proceedings. . .** IEEE Computer Society, 2009. p. 77–82. (NGMAST ’09).

BETTINI, C.; BRDICZKA, O.; HENRICKSEN, K.; INDULSKA, J.; NICKLAS, D.; RANGANATHAN, A.; RIBONI, D. A survey of context modelling and reasoning techniques. **Pervasive Mob. Comput.**, Amsterdam, The Netherlands, The Netherlands, v. 6, n. 2, p. 161–180, Apr. 2010.

BROCKWELL, P.; DAVIS, R. **Introduction to Time Series and Forecasting**. [S.l.]: Springer, 2002. (Springer Texts in Statistics).

BROWN, P.; BOVEY, J.; CHEN, X. Context-aware applications: from the laboratory to the marketplace. **Personal Communications, IEEE**, [S.l.], v. 4, n. 5, p. 58 –64, oct 1997.

BUSH, V.; WANG, J. As we may think. **Atlantic Monthly**, [S.l.], v. 176, p. 101–108, 1945.

CARMAGNOLA, F.; CENA, F. User identification for cross-system personalisation. **Inf. Sci.**, New York, NY, USA, v. 179, n. 1-2, p. 16–32, Jan. 2009.

CHENG, Z.; CAVERLEE, J.; LEE, K.; SUI, D. Z. Exploring millions of footprints in location sharing services. In: 2011, 2011. **Anais. . .** [S.l.: s.n.], 2011.

CIARAMELLA, A.; CIMINO, M.; LAZZERINI, B.; MARCELLONI, F. Using context history to personalize a resource recommender via a genetic algorithm. In: INTELLIGENT SYSTEMS DESIGN AND APPLICATIONS (ISDA), 2010 10TH INTERNATIONAL CONFERENCE ON., 2010. **Anais. . .** [S.l.: s.n.], 2010. p. 965 –970.

COEN, M.; PHILLIPS, B.; WARSHAWSKY, N.; WEISMAN, L.; PETERS, S.; FININ, P. Meeting the Computational Needs of Intelligent Environments: the metaglue system. In: IN PROCEEDINGS OF MANSE'99, 1999. **Anais. . .** Springer-Verlag, 1999. p. 201–212.

COSTA, C. A. da; YAMIN, A. C.; GEYER, C. F. R. Toward a General Software Infrastructure for Ubiquitous Computing. **IEEE Pervasive Computing**, Piscataway, NJ, USA, v. 7, n. 1, p. 64–73, Jan. 2008.

DAVISON, B. D.; HIRSH, H. Predicting Sequences of User Actions. In: 1998. **Anais. . .** AAAI Press, 1998. p. 5–12.

DEY, A. K.; ABOWD, G. D.; SALBER, D. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. **Hum.-Comput. Interact.**, Hillsdale, NJ, USA, v. 16, n. 2, p. 97–166, Dec. 2001.

DEY A. K. HIGHTOWER J., L. E. D. N. Location-Based Services. **IEEE Pervasive Computing**, [S.l.], v. 9, n. 11, 2010.

DIAZ, A.; MERINO, P.; RIVAS, F. Mobile Application Profiling for Connected Mobile Devices. **Pervasive Computing, IEEE**, [S.l.], v. 9, n. 1, p. 54 –61, jan.-march 2010.

DOHERTY, A. R.; CAPRANI, N.; CONAIRE, C.; KALNIKAITE, V.; GURRIN, C.; SMEATON, A. F.; O'CONNOR, N. E. Passively recognising human activities through lifelogging. **Comput. Hum. Behav.**, Amsterdam, The Netherlands, The Netherlands, v. 27, n. 5, p. 1948–1958, Sept. 2011.

DRIVER, C.; CLARKE, S. An application framework for mobile, context-aware trails. **Pervasive Mob. Comput.**, Amsterdam, The Netherlands, The Netherlands, v. 4, n. 5, p. 719–736, Oct. 2008.

ELGAZZAR, K.; ABOELFOTOH, M.; MARTIN, P.; HASSANEIN, H. S. Ubiquitous Health Monitoring Using Mobile Web Services. **Procedia Computer Science**, [S.l.], v. 10, n. 0, p. 332 – 339, 2012.

FOLL, S.; HERRMANN, K.; ROTHERMEL, K. PreCon - Expressive Context Prediction Using Stochastic Model Checking. In: HSU, C.-H.; YANG, L.; MA, J.; ZHU, C. (Ed.). **Ubiquitous Intelligence and Computing**. [S.l.]: Springer Berlin / Heidelberg, 2011. p. 350–364. (Lecture Notes in Computer Science, v. 6905). 10.1007/978-3-642-23641-9_29.

FRANCO, L. K.; ROSA, J. H.; BARBOSA, J. L. V.; COSTA, C. A.; YAMIN, A. C. MUCS: a model for ubiquitous commerce support. **Electron. Commer. Rec. Appl.**, Amsterdam, The Netherlands, The Netherlands, v. 10, n. 2, p. 237–246, Mar. 2011.

GEMMELL, J.; BELL, G.; LUEDER, R.; DRUCKER, S.; WONG, C. MyLifeBits: fulfilling the memex vision. In: ACM INTERNATIONAL CONFERENCE ON MULTIMEDIA, 2002, New York, NY, USA. **Proceedings...** ACM, 2002. p. 235–238. (MULTIMEDIA '02).

GYORBIRO, N.; FABIAN, A.; HOMANYI, G. An activity recognition system for mobile phones. **Mob. Netw. Appl.**, Hingham, MA, USA, v. 14, n. 1, p. 82–91, Feb. 2009.

HECKMANN, D. Integrating Privacy Aspects into Ubiquitous Computing: a basic user interface for personalization. In: ARTIFICIAL INTELLIGENCE IN MOBILE SYSTEMS (AIMS 2003), 2003, Seattle, USA. **Anais...** [S.l.: s.n.], 2003. p. 106–110.

HECKMANN, D. **Ubiquitous User Modeling**. [S.l.]: Akademische Verlagsgesellschaft, 2005. (Dissertationen zur künstlichen Intelligenz - DISKI).

HECKMANN, D.; KRUEGER, A. A user modeling markup language (userML) for ubiquitous computing. In: USER MODELING, 9., 2003, Berlin, Heidelberg. **Proceedings...** Springer-Verlag, 2003. p. 393–397. (UM'03).

HIGHTOWER, J.; BORRIELLO, G. Location systems for ubiquitous computing. **Computer**, [S.l.], v. 34, n. 8, p. 57 –66, aug 2001.

HIGHTOWER, J.; LAMARCA, A.; SMITH, I. Practical Lessons from Place Lab. **Pervasive Computing, IEEE**, [S.l.], v. 5, n. 3, p. 32 –39, july-sept. 2006.

HOAREAU, C.; SATOH, I. Modeling and Processing Information for Context-Aware Computing: a survey. **New Generation Computing**, [S.l.], v. 27, p. 177–196, 2009. 10.1007/s00354-009-0060-5.

HONG, J.; SUH, E.-H.; KIM, J.; KIM, S. Context-aware system for proactive personalized service based on context history. **Expert Syst. Appl.**, Tarrytown, NY, USA, v. 36, n. 4, p. 7448–7457, May 2009.

JAMESON. Systems That Adapt to Their Users: an integrative perspective. **Department of Computer Science, Saarland University**, [S.l.], 2001.

JURSA R., L. B. R. K. Advanced wind power predicting with Artificial intelligence methods. In: ARTIFICIAL INTELLIGENCE IN ENERGY SYSTEMS AND POWER, 2006. **Anais...** AIESP 2006, 2006.

KALATZIS, N.; ROUSSAKI, I.; LIAMPOTIS, N.; STRIMPAKOU, M.; PILS, C. User-centric inference based on history of context data in pervasive environments. In: SERVICES INTEGRATION IN PERVASIVE ENVIRONMENTS, 3., 2008, New York, NY, USA. **Proceedings...** ACM, 2008. p. 25–30. (SIPE '08).

KEOGH, E. J.; PAZZANI, M. J. **An Enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback**. 1998.

KLYNE, G.; REYNOLDS, F.; WOODROW, C.; OHTO, H.; HJELM, J.; BUTLER, M. H.; TRAN, L. **Composite Capability/Preference Profiles (CC/PP)**: structure and vocabularies. [S.l.]: World Wide Web Consortium (W3C), 2005. Online verfügbar unter `http://www.w3.org/TR/CCPP-struct-vocab`, abgerufen am 18. Oktober 2005.

KOLDA, T. G.; BADER, B. W. Tensor Decompositions and Applications. **SIAM REVIEW**, [S.l.], v. 51, n. 3, p. 455–500, 2009.

KONIG, I.; VOIGTMANN, C.; KLEIN, B. N.; DAVID, K. Enhancing alignment based context prediction by using multiple context sources: experiment and analysis. In: MODELING AND USING CONTEXT, 7., 2011, Berlin, Heidelberg. **Proceedings...** Springer-Verlag, 2011. p. 159–172. (CONTEXT'11).

KRSUL, I. **Authorship analysis**: identifying the author of a program. [S.l.]: Department of Computer Sciences, Purdue University, 1994.

LANGHEINRICH, M. Privacy in Ubiquitous Computing. In: UBIQUITOUS COMPUTING, 2009. **Anais...** CRC Press, 2009. p. 95–160.

LEE, M. L.; DEY, A. K. Lifelogging memory appliance for people with episodic memory impairment. In: UBIQUITOUS COMPUTING, 10., 2008, New York, NY, USA. **Proceedings...** ACM, 2008. p. 44–53. (UbiComp '08).

LEE, S.; PARK, S.; LEE, S.-g. A Study on Issues in Context-Aware Systems Based on a Survey and Service Scenarios. In: ACIS INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, ARTIFICIAL INTELLIGENCES, NETWORKING AND PARALLEL/DISTRIBUTED COMPUTING, 2009., 2009, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2009. p. 8–13. (SNPD '09).

LEICHTENSTERN, K.; LUCA, A. D.; RUKZIO, E. Analysis of Built-in Mobile Phone Sensors for Supporting Interactions with the Real World. In: PERMID'05, 2005. **Anais...** [S.l.: s.n.], 2005. p. 31–34.

LEWIS, M.; NINO, C.; ROSA, J.; BARBOSA, J.; BARBOSA, D. A management model of learning objects in a Ubiquitous Learning environment. In: PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS (PERCOM WORKSHOPS), 2010 8TH IEEE INTERNATIONAL CONFERENCE ON, 2010. **Anais...** [S.l.: s.n.], 2010. p. 256 –261.

LéVY, P. Processus semi-markoviens. In: IN PROC. OF INTERNATIONAL CONGRESS OF MATHEMATICS, 1954. **Anais...** [S.l.: s.n.], 1954.

MANIKANDAN, D.; MADHUSUDHANAN, J.; PRASANNA, V.; AMRITH, V.; BRITTO, M. Smart banking environment based on context history. In: RECENT TRENDS IN INFORMATION TECHNOLOGY (ICRTIT), 2011 INTERNATIONAL CONFERENCE ON, 2011. **Anais...** [S.l.: s.n.], 2011. p. 450 –455.

MANTORO, T.; MUATAZ, Z.; AYU, M. Mobile user location prediction: genetic algorithm-based approach. In: INDUSTRIAL ELECTRONICS APPLICATIONS (ISIEA), 2010 IEEE SYMPOSIUM ON, 2010. **Anais...** [S.l.: s.n.], 2010. p. 345 –349.

MANTYJARVI, J.; TUTKIMUSKESKUS, V. teknillinen. **Sensor-based Context Recognition for Mobile Applications**. [S.l.]: VTT, 2003. (VTT publications).

MARTIN, M.; NURMI, P. A Generic Large Scale Simulator for Ubiquitous Computing. In: MOBILE AND UBIQUITOUS SYSTEMS - WORKSHOPS, 2006. 3RD ANNUAL INTERNATIONAL CONFERENCE ON, 2006. **Anais...** [S.l.: s.n.], 2006. p. 1 –3.

MAYRHOFER, R. An Architecture for Context Prediction. In: FERSCHA, A.; HÖRTNER, H.; KOTSIS, G. (Ed.). **Advances in Pervasive Computing**. [S.l.]: Austrian Computer Society (OCG), 2004. v. 176, p. 65–72. part of the Second International Conference on Pervasive Computing (PERVASIVE 2004).

MAYRHOFER, R. Context Prediction based on Context Histories: expected benefits , issues and current state-of-the-art. In: IN PROCEEDINGS OF EXPLOITING CONTEXT HISTORIES IN SMART ENVIRONMENTS (ECHISE)'05., 2005. **Anais...** [S.l.: s.n.], 2005.

MEHTA, B.; NIEDEREE, C.; STEWART, A.; DEGEMMIS, M.; LOPS, P.; SEMERARO, G. Ontologically-Enriched unified user modeling for cross-system personalization. In: USER MODELING, 10., 2005, Berlin, Heidelberg. **Proceedings...** Springer-Verlag, 2005. p. 119–123. (UM'05).

MEINERS, M.; ZAPLATA, S.; LAMERSDORF, W. Structured context prediction: a generic approach. In: IFIP WG 6.1 INTERNATIONAL CONFERENCE ON DISTRIBUTED APPLICATIONS AND INTEROPERABLE SYSTEMS, 10., 2010, Berlin, Heidelberg. **Proceedings...** Springer-Verlag, 2010. p. 84–97. (DAIS'10).

MULVENNA BROWN P., B. W. L. M. R. O. R. G. S. J. S. D. Context-awareness: some compelling applications. In: CH12000 WORKSHOP ON THE WHAT, WHO, WHERE, WHEN, WHY AND HOW OF CONTEXT-AWARENESS., 2000. **Proceedings...** [S.l.: s.n.], 2000.

MULVENNA, M.; NUGENT, C.; GU, X.; SHAPCOTT, M.; WALLACE, J.; MARTIN, S. Using context prediction for self-management in ubiquitous computing environments. In: CONSUMER COMMUNICATIONS AND NETWORKING CONFERENCE, 2006. CCNC 2006. 3RD IEEE, 2006. **Anais...** [S.l.: s.n.], 2006. v. 1, p. 600 – 604.

NIEDEREE, C.; STEWART, A.; MEHTA, B.; HEMMJE, M. A Multi-Dimensional, Unified User Model for Cross-System Personalization. In: Procceedings of the AVI 2004 Workshop On Environments For Personalized Information Access, 2004. **Anais...** [S.l.: s.n.], 2004.

PADGHAM, L.; WINIKOFF, M. **Developing Intelligent Agent Systems A practical guide**. [S.l.]: Michael Wooldridge, Liverpool University, UK, 2004.

PATTERSON, D. J.; LIAO, L.; FOX, D.; KAUTZ, H. Inferring High-Level Behavior from Low-Level Sensors. In: UBIQUITOUS COMPUTING, 2003. **Anais...** [S.l.: s.n.], 2003. p. 73–89.

RODGERS, J. L.; NICEWANDER, A. W. Thirteen Ways to Look at the Correlation Coefficient. **The American Statistician**, [S.l.], v. 42, n. 1, p. 59–66, 1988.

ROSA, J.; CAMPES, C.; BARBOSA, J.; COSTA, C.; KICH, M. COMPETENCE MANAGEMENT IN TRAIL-AWARE ENVIRONMENTS. In: IADIS INTERNATIONAL CONFERENCE WWW/INTERNET 2011, 2011. **Anais...** [S.l.: s.n.], 2011. p. 145–152.

SALBER, D.; DEY, A. K.; ABOWD, G. D. **Ubiquitous Computing**: defining an hci research - agenda for an emerging interaction paradigm. [S.l.: s.n.], 1998.

SATYANARAYANAN, M.; BAHL, P.; CACERES, R.; DAVIES, N. The Case for VM-Based Cloudlets in Mobile Computing. **Pervasive Computing, IEEE**, [S.l.], v. 8, n. 4, p. 14 –23, oct.-dec. 2009.

SELLEN, A. J.; WHITTAKER, S. Beyond total capture: a constructive critique of lifelogging. **Commun. ACM**, New York, NY, USA, v. 53, n. 5, p. 70–77, May 2010.

SHIN, D.-H. Ubiquitous city: urban technologies, urban infrastructure and urban informatics. **J. Inf. Sci.**, Thousand Oaks, CA, USA, v. 35, n. 5, p. 515–526, Oct. 2009.

SIGG, S. **Development of a Novel Context Prediction Algorithm and Analysis of Context Prediction Schemes**. [S.l.]: Kassel University Press, 2008.

SIGG, S.; GORDON, D.; ZENGEN, G.; BEIGL, M.; HASELOFF, S.; DAVID, K. Investigation of Context Prediction Accuracy for Different Context Abstraction Levels. **Mobile Computing, IEEE Transactions on**, [S.l.], v. PP, n. 99, p. 1, 2011.

SIGG, S.; HASELOFF, S.; DAVID, K. An Alignment Approach for Context Prediction Tasks in UbiComp Environments. **Pervasive Computing, IEEE**, [S.l.], v. 9, n. 4, p. 90 –97, october-december 2010.

SILVA, J.; ROSA, J.; BARBOSA, J.; BARBOSA, D.; PALAZZO, L. Content distribution in trail-aware environments. **Journal of the Brazilian Computer Society**, [S.l.], v. 16, p. 163–176, 2010. 10.1007/s13173-010-0015-1.

SLANEY, J. K.; THIéBAUX, S. Blocks World revisited. **Artif. Intell.**, [S.l.], p. 119–153, 2001.

SMITH, W. Regenerative stochastic processes. In: R. SOC. LOND. SER. A MATH. PHYS. ENG., 1955. **Proceedings...** [S.l.: s.n.], 1955. p. 6–31.

SONG, I.; HAAM, D.; KIM, H.; KIM, M. H. OntLMS: an ontology-based lifelog management system. In: WEB CONFERENCE (APWEB), 2010 12TH INTERNATIONAL ASIA-PACIFIC, 2010. **Anais...** [S.l.: s.n.], 2010. p. 341 –343.

TAKACS, L. Some investigations concerning recurrent stochastic processes of a certain type. In: MAGYAR TUD. AKAD. MAT. KUTATO INT. KZL., 1954. **Anais...** [S.l.: s.n.], 1954. p. 115–128.

VAUGHAN-NICHOLS, S. Will Mobile Computing's Future Be Location, Location, Location? **Computer**, [S.l.], v. 42, n. 2, p. 14 –17, feb. 2009.

Viviani, M.; Bennani, N.; Egyed-Zsigmond, E. A Survey on User Modeling in Multi-Application Environments. In: THE THIRD INTERNATIONAL CONFERENCE ON ADVANCES IN HUMAN-ORIENTED AND PERSONALIZED MECHANISMS, TECHNOLOGIES, AND SERVICES CENTRIC'10, 2010. **Anais...** IEEE, 2010. p. 111–116.

VOIGTMANN, C.; LAU, S.; DAVID, K. An approach to Collaborative Context Prediction. In: PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS (PERCOM WORKSHOPS), 2011 IEEE INTERNATIONAL CONFERENCE ON, 2011. **Anais...** [S.l.: s.n.], 2011. p. 438 –443.

WANT, R.; SCHILIT, B. N.; ADAMS, N. I.; GOLD, R.; PETERSEN, K.; GOLDBERG, D.; ELLIS, J. R.; WEISER, M. An Overview of the ParcTab Ubiquitous Computing Experiment. **IEEE Personal Communications**, [S.l.], v. 2, p. 28–43, 1995.

WEISER, M. The Computer for the Twenty-First Century. **Scientific American**, [S.l.], v. 265(3), p. 94–104, 1991.

WEISER, M. Some computer science issues in ubiquitous computing. **Commun. ACM**, New York, NY, USA, v. 36, n. 7, p. 75–84, July 1993.

WEISS, G. M.; HIRSH, H. **Learning to Predict Rare Events in Categorical Time-Series Data**. [S.l.]: In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, CA, 1998.