Computing and Informatics, Vol. 37, 2018, 1411–1442, doi: 10.4149/cai_2018_6_1411

USING PROBABILISTIC TEMPORAL LOGIC PCTL AND MODEL CHECKING FOR CONTEXT PREDICTION

Darine Ameyed

Synchromedia Laboratory, Quebec University, École de Technologie Supérieure
Montreal, Canada
e-mail: darine.ameyed.1@ens.etsmtl.ca

Moeiz Miraoui

Higher Institute of Applied Science and Technology of Gafsa, University of Gafsa
Gafsa, Tunisia
e-mail: moeizmiraoui@gmail.com

Atef Zaguia

College of Computers and Information Technology, Taif University, Hawiyah
Taif, Kingdom of Saudi Arabia
e-mail: zaguia.atef@tu.edu.sa

Fehmi Jaafar

Faculty of Management of Concordia University of Edmonton
Canada
e-mail: fehmi.jaafar@concordia.ab.ca

Chakib Tadj

MMS Laboratory, Quebec University, École de Technologie Supérieure
Montreal, Canada
e-mail: chakib.tadj@etsmtl.ca

Abstract. Context prediction is a promoting research topic with a lot of challenges and opportunities. Indeed, with the constant evolution of context-aware systems, context prediction remains a complex task due to the lack of formal approach. In this paper, we propose a new approach to enhance context prediction using a probabilistic temporal logic and model checking. The probabilistic temporal logic PCTL is used to provide an efficient expressivity and a reasoning based on temporal logic in order to fit with the dynamic and non-deterministic nature of the system's environment. Whereas, the probabilistic model checking is used for automatically verifying that a probabilistic system satisfies a property with a given likelihood. Our new approach allows a formal expressivity of a multidimensional context prediction. Tested on real data our model was able to achieve 78 % of the future activities prediction accuracy.

Keywords: Context prediction, logic, PCTL, pervasive system, context-aware system, stochastic, transition model

Mathematics Subject Classification 2010: 68T01, 68T30, 68T37, 68U35

1 INTRODUCTION

Prediction is a research topic in different fields: meteorology, economy, trends of prices and stocks as well as in computer science and software engineering such as predicting failure in software [1]. Predictive mechanisms help to anticipate actions and to implement the appropriate preventive measures. Ubiquitous computing systems are no exception in this respect; they do actually follow this trend. To be more proactive, ubiquitous systems have to provide service adaptation, according to the dynamic evolution of their context, in order to offer an adequate service fitting the user's needs.

One significant challenge, in particular, is to proactively assess the user's needs in the real world without requiring explicit input. Furthermore, a ubiquitous system must provide the user with services well adapted to the overall context. Indeed, services will be triggered dynamically and without an explicit user intervention in a proactive way. Making use of the context in applications is a current area of research known as "context-awareness" [2, 7]. A sensitive-context application must perceive the context of the users and their environment and adapt its behaviour accordingly. Most of the work on service adaptation in context-awareness is focused on the current context.

In ubiquitous computing several studies and research have been conducted too, under the prediction topic [2, 3, 4, 5]. These works aim to introduce new prediction techniques to increase the dynamic nature and the proactivity of those pervasive systems.

1.1 Problem and Motivation

Predict the future context allows the pervasive system to choose the most effective strategies to achieve its goals and to provide an active and fast adaptation to future situations.

However, the existing approaches face key issues that need to be addressed:

1. provide a multi-dimensional context prediction,
2. support a temporal constraint and identify the expected time of context variations,
3. improve expressiveness and provide a clear semantics.

Current approaches in context prediction only deduce one-dimensional information for the future context (e.g. future location). As a consequence, their expressiveness and effectiveness are limited. Even more so, if the system is unable to recognize the expected time of such context changes and the underlying behavior.

Moreover, these approaches face a common challenge: the lack of formal and general approaches for dealing with context prediction and more specifically, allowing proactivity and service anticipation using context prediction. They assert the lack of a common development framework for context prediction as well as formal representation for the context and a formal approach for the prediction.

Over the past few years, a more general research trend emerged, focusing on context prediction such as the work described in [5, 6], which discussed directions for research on this issue. They pointed out that the work in this area is mostly limited to location information, and a challenge they face is:

1. to consider more general context information,
2. to be able to support a temporal constraint and
3. to provide a logic-based expressive prediction with a clear semantic and formalism.

1.2 Proposition

Pervasive proactive systems need, therefore, the ability to reason with time dependencies and even more complex than that: spatiotemporal dimensions and the overall context. To be able to recognise a future contextual information (e.g., where is the location of the user X in the next 5 minutes?) and to provide an answer and anticipate a service associated with a future context must be possible (e.g., activity X can be executed on location Y in the next Z minute). A system that can include this kind of knowledge provides more flexibility and allows the ability to act in a more efficient manner.

In previous research work, we emphasized on context prediction context in pervasive context-aware systems. We proposed a new definition that supports prediction in the same multi-dimension reasoning [8]. In another step towards the goal of providing formal prediction approach to context modeling we proposed a logic-based model including a temporal constraint [9]. This paper is, therefore, another step to provide a new spatiotemporal expressive prediction based on a formal semantic of probabilistic temporal logic and stochastic transition model.

1.3 Contribution

The efficient deployment of a context-sensitive prediction, its dynamic and unpredictable evolution, is still limited due to a semantic gap between the data provided by the physical detection devices and the information needed to predict the future behavior of the system and its users. Our proposed approach exceeds the weaknesses identified in the literature [5, 10, 11] by providing: better context expressivity, more efficient prediction based on logical reasoning, stochastic, non-deterministic modeling and below a multidimensional approach, what fitting better the nature of ambient systems.

In this paper, we are formalizing a new approach to express context prediction in context-aware systems. We express context and the transition in a pervasive system with a formal semantic, using a probabilistic temporal logic PCTL (a probabilistic extension of temporal logic). We propose a probabilistic transition model to encode the system's behavior over the time. Combining PCTL with a stochastic model, we can trace, analyze and predict the future context. Thus, we propose to use the model checking verification to verify the future state properties with a quantitative result and return the future state that has the maximum probability.

1.4 Paper's Structure

The paper is organized as follows. First, we give an overview of the available prediction methods (Section 2) with a synthesis and an evaluation. After that, we present our approach (Section 3) starting with a presentation of temporal logic and an explanation of the choice of probabilistic temporal logic. We then present a model detailing each included component. And we finish this section by explaining the prediction process. Before concluding the paper, we present the evaluation of our approach (Sections 3.6, 3.7) and expected future work (Section 4).

2 RELATED WORK

In this section we give an overview of the available research within the context prediction topic, specifically including proactive adaptation for pervasive systems; we analyze, discuss those various works, and later we present an evaluation/synthesis according to a selected set of criteria. As we have discussed and analyzed the prediction research work in a previous survey [11], according to the technical prediction approaches, we tried in this overview to discuss other related work, mostly from recent research in chronological order.

Also, we circumscribed a survey to research proposing generic models to support context prediction. Hence, the chosen works should support generic context information: works specifically devoted to the location prediction were not considered relevant. As discussed in recent surveys [11, 12] the development of generic approaches is a challenge in this research area.

One of the first contributions in context prediction was proposed by Mayrhofer [13]. Mayrhofer proposed architecture and a framework for context prediction that are based on an unsupervised classification, attempting to find context clusters, previously unknown from the input data. These context clusters represented recurring patterns in the input data. This approach modeled the context as a finite sequence of states where a user or a device triggers the change of the current state from one state to another. This modeling helped to predict the next states of the context based on the current state. He suggested a five-step process, taking sets of observations, each recorded at a specific time, as input and providing as output the current context of the user as well as predicting the future states of the context. The proposed stages are sensor data acquisition, feature extraction, classification, labeling, and prediction.

Mayrhofer proposed a prediction module based on the sequence prediction technique. This technique is based on the prediction task of a theoretical computer sequence and can only be applied if the context is broken down into some form of event flow. The context prediction in this work is based only on high-level context, and the framework does not have any mechanism to support an adaptive strategy.

Like Mayrhofer, Sigg et al. [14, 6] provided a formal definition for the context prediction task relevant to the issues raised on the quality of the context and on how to handle the ambiguity of incomplete data. This method is also based on patterns of context the learning algorithm builds to enable the prediction module. The context prediction module is based on an alignment method that attempts to predict the most likely continuation of a time series starting from the suffix of the observed sequence.

Finally, Sigg et al. [6] also offer a continuous learning module to adapt to the change in the environment or user habits. It continuously monitors the recorded time series stored in context history and updates the relevant patterns.

However, we did not find in this work any specific implementation for this learning module. Only its constraints were given, including the interface specified by the context history and language description of the rules, representing patterns. Sigg does not describe any adaptive mechanism for prediction neither considers any specification for context information.

Meiners et al. [15] suggested a context prediction approach called SCP (Structured Context Prediction). This approach is based on two key principles. The first is making use of knowledge of the application domain that developers can integrate when designing the application. This knowledge is described as a prediction model that specifies how the predictions are to be executed and which configures the prediction system. The second principle sets out the application of several prediction methods, which are interchangeable. These methods are proposed to ensure the accuracy and effectiveness of predictions relevant to a given domain. They can be selected and combined by the application developers. According to [15] the prediction model assigns a method for each variable to predict its value. The method uses as input the values of other variables that are either already predicted by their own methods, or simply measured by sensors. Also, the authors proposed an architecture for a prediction system which can be used as a reusable component by context-aware applications.

In this work, the proposed Contexts Prediction architecture supports 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 supports only low-level context data and does not have a formal context representation.

Contextual spaces theory is an approach developed by Andrey Boytsov [3], to best define context-awareness and to deal with sensor problems that create uncertainty and incur a lack of reliability. This theory used spatial metaphors to represent the context as a multidimensional space. It was designed to make context-awareness clearer.

The theory of context space was initially submitted by Padovitz and Zaslavsky [16]. The authors attempted to provide a general model to help thinking about and to describe the context and develop context-aware applications. This work will be later the basis for several researches of Zaslavzky and Boytsov [4, 3, 17]. Boytsov and Zaslavsky presented the CALCHAS system, which offered context prediction and used an extension to the context space theory to provide proactive adaptation.

This approach addressed the context prediction problem in a general sense. In context spaces theory several methods were tested and used for reasoning about the context. The authors judged sequence technique as the most prospective prediction approach.

For adaptation mechanisms, algebraic operations on situations and some logic-based methods were developed for reasoning regarding situations [18].

This works had presented a general framework model, included an adaptation approach based on prediction but did not propose a new formal or a generic prediction method.

In her work on services prediction, Salma Najar offered a mechanism of discovery and prediction guided both by context and user intent [19]. She used semantic similarity techniques. The system is based on the implementation of a matching algorithm, which computes the matching degree between the intention and the current context of the user and the set of semantic services described accordingly. OWL-SIC (OWL-S Intentional & Contextual) is an extension of OWL-S (Web Ontology Language-Semantic, is an ontology, within the OWL-based framework of the Semantic).

The similarity approach required historical data, to select and recommend services that are not always available. In fact, it needs a first phase of a collection to get enough data which will be processed after that. The intentional approach provided by Najar [19] was a user-centered approach but can generate conflict: for instance a problem of interoperability between services. Indeed, two compatible intentions do not necessarily map to two technically compatible services. This work also proposed a conceptual framework focused on services prediction.

Joao et al. [5] proposed new framework including a prediction-algorithms library. They named the proposed model ORACON. The architecture of this model 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 proposed prediction of entities. An entity, in this sense, can be a living being, an object 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. This work focused more on the framework; it did not propose a specific prediction approach. There prediction algorithm library contains four prediction approaches: alignment, enhanced alignment, semi-Markov and collaboration [5]. This proposed model was an interesting work which can be enhanced with many extensions to improve the performance, increase the accuracy of classification and optimize the processing time.

Föll et al. [20] proposed a PreCon as a multi-dimensional context predicting method, composed of three parts: a stochastic model to represent context changes, an expressive temporal-logic query language using CSL (continuous stochastic language) and stochastic algorithms to predict the context. The model based on user behavior was presented as an SMC (Semi-Markov Chain).

This work was the unique formal work using the CSL as a query language of the system, and a Semi-Markov Chain. There is also another work that had tried to automate the recognition of activities using the LTL formalism with a model checking [21].

They concluded their work, noting that a probabilistic extension using a PCTL can increase the expressive power of the formal core.

We found this to be the most relevant work, and we based our approach on it, specifically in a model checking verification. We use PCTL formalism and include action in a model to get a more descriptive model.

2.1 Synthesis

Table 1 summarizes a comparison of the related works. As we can see the majority of works do not support formal representation of the context, low and high context level. They focused more on providing a framework including a predictive module, rather than on the prediction module itself. The essential part of a prediction model being the approach used in the prediction process itself.

Ubiquitous environments are highly dynamic, that is, applications can interact with a great number of different and unknown applications all the time [22, 23, 24]. Hence, it is essential to define a formal representation for the context, so that different systems can easily communicate. Thus, specifying a context representation is considered a key feature for model prediction. This is why we choose

	Adaptive strategy	**Context formal presentation**	**Low-high context level**	**Learning capability**	**Prediction technique**	**Framework proposed**
Mayrhofer 2004	no	no	no	yes	sequence prediction approach	yes
Sigg 2008–2010	no	no	yes	yes	trajectory prolongation	yes
Meiners 2010	yes	no	yes	yes	Bayesian	yes
Boytsov 2011	yes	yes	no	yes	sequence predictor the most perspective approach	yes
S. Föll 2014	no	no	yes	yes	temporal query prediction	no
S. Najar 2014	yes	no	no	yes	semantic similarity (discovering)	yes
H. Joa 2016	yes	yes	yes	yes	alignment semi-markov	yes

Table 1. Comparative overview of context prediction research work

a formal context representation based on a logic perspective [9]. Also, we build a model in a temporal logic formalism providing clear formal semantics by using a probabilistic temporal logic (PCTL), and we propose a new probabilistic-labeled transaction model Model-LPTM. One might also conclude that the prediction approaches supported by previous works compute the most probable future context, based on simple uni-dimensional context information. Existing systems do not allow a formal context prediction through temporal-semantics and multidimensional processes.

In this paper, we propose to investigate the application of probabilistic temporal logic as a powerful formal presentation for context prediction. It also proposes a formal prediction approach based on temporal logic in a multidimensional context space and on a new formalism that integrates probability and labeling; which provide a new probabilistic labeled transaction model thus helping effective context-aware prediction.

3 THE PROPOSED APPROACH

3.1 Temporal Logic in the Context Aware System

Time is a fascinating subject. We are moving through time continuously, and in order to survive and manage ourselves, we regularly have to make temporal-logic-based decisions. In daily lives, people are using time-dependent information, e.g. when to go to the dentist? When is a meeting to be held? With the rise of ubiquitous systems (which ideally aim to provide a smart user-focused service; like reminder services, assisted-living services and more), temporal analysis and reasoning appear best-suited to ensure the proper functioning for this kind of system. Temporal logic can also be used as a programming language. The basic paradigm is to review the past and then take action in the future. Abstractly we have an initial state and certain actions that can be performed in a given state if it satisfies a certain set of conditions. Performing an action on a state produces a new state.

We have defined a variant of TL (temporal logic) as a language for the specification of each situation and its related context. In general, TL has been developed and applied as a formalism for reasoning about the ordering and quantitative timing of events [25]. Several formulations have been proposed to satisfy the needs of different contexts. TL may be classified according to the underlying nature of time: linear temporal logic LTL and computational tree logic CTL.

LTL, CTL and CTL* can express qualitative properties of a system. Real systems such as a pervasive system, however, are quite often characterized by non-deterministic behavior and this is because of the human presence. In order to provide efficient services, to be user-centric and more realistic, those systems should be attuned to the unpredictable behavior of humans. Taking probabilities into account, in addition to non-deterministic behavior, would expand this aspect of the system allowing the quantification of unpredictable behavior, if the specification holds with an arbitrary probability value and within a given time limit.

We propose to use PCTL, which had the expressive power of probabilistic temporal logic (it introduces probability to extend CTL which is inadequate in dealing with a real-life system like a ubiquitous computing system) (Figure 1).

3.2 Probabilistic Temporal Logic Specification

Temporal logic extends the traditional modal logic to allow the description of when a formula is true. That is, rather than just "necessity" or "possibility", a formula may be true at the next point in time or at some other point in the future.

Branching time logic, such as Computation Tree Logic (CTL) [26], enables the choice of a path among multiple possible paths in a tree structure describing probable future events. So that, each choice has to mirror the possible set of behaviors starting from the current state. As opposed to linear-time temporal logic, for which, there is only one possible future path, we can express whether a property holds for all possible paths (A formula), or if there exists at least one path for which it is true

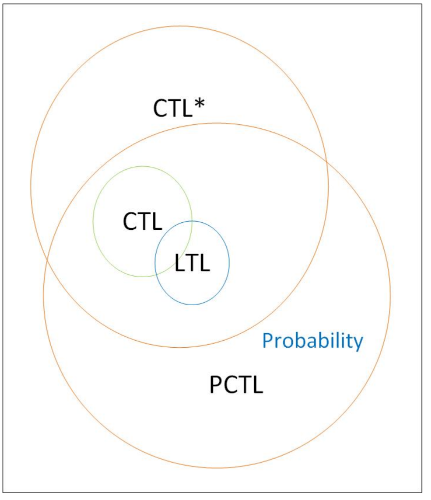

Figure 1. Expressivity CTL vs. LTL vs. CTL* vs. PCTL

(E formula). The values of these formulas are determined with a Kripke structure: a graph with a set of states, transitions between states, and labels indicating which propositions are true within the states.

We will use a probabilistic extension of CTL, Probabilistic Computation Tree Logic (PCTL) [27, 28] as it allows probabilistic state transitions, as well as explicit deadlines for when a formula must hold.

The proposed PCTL syntax is based on the syntax and semantics proposed in [27, 28]. For the sake of clarity, some specific notations, as well as the underlying probabilistic model, have been slightly modified from the original syntax presented by those papers, in order to adapt them to the work context.

In this section, we present the proposed model for the context prediction problem based on the real-world situation and the related features; which represent the contextual information of each situation (e.g. location, time, occupation, ambient information, sound, temperature, etc.).

Figure 2 summarizes the proposed approach. It is based on PCTL formalism, a probabilistic labeled transition model which will be detailed later (Subsection 3.3). The context prediction is based on model checking, which will return the future situation and its probability.

A. Formalism

a. Context

Definition 1. In order to specify this situation-context, let $s = (c_1, c_2, \dots, C_n) \in S$, s being an n-dimension vector of context information described by a preposition or a combination of prepositions, where each component c_i of s

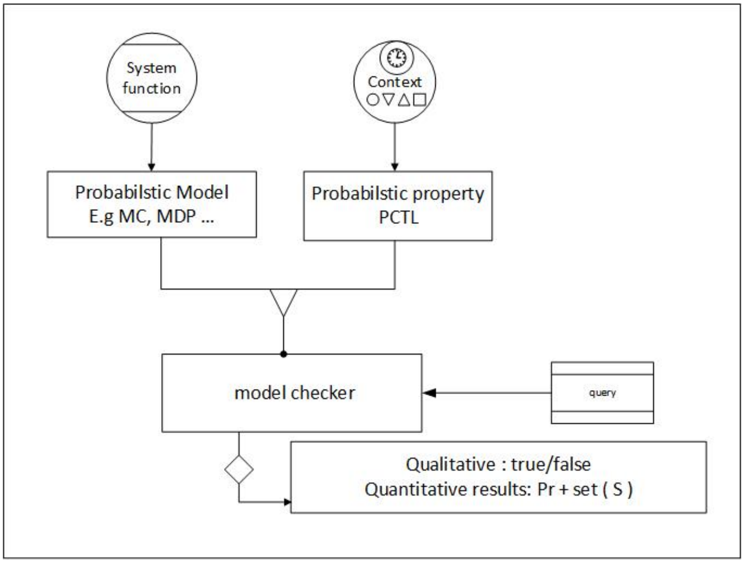

Figure 2. Overview of the proposed approach

is of a specific context type C_i (e.g. ⟨location⟩, ⟨occupation⟩, ...). A state s can be multidimensional and expresses composite contextual data describing the features of a specific situation; e.g., $s = (($meeting-room$)x($power-point$)$ $\times($occupation $= 5))$ designates the presentation situation on a meeting room.

For each new combination of context information $(c_1, \dots, c_n)$ that has not been observed before, is detected, a new state s will be inserted into the model and labeled with $(c_1, \dots, c_n)$. For more details about the context logic-based modeling we refer to previous related work [9].

b. Path and state

The prediction semantics is based on PCTL syntax. For this let $p \in [0, 1]$ be a probability, let $\mathbf{t} \in \mathbb{R}^+$ be a time-bound, and let (C_i, c_i) be a contextual value c_i of type C_i as defined earlier.

Definition 2. Path formulas express the properties and behaviour allocated to paths.

$$\varphi = X^{\leq t}\Phi|\Phi_1 \bigcup^{\leq t} \Phi_2.$$

State formulas express the properties and behaviour allocated to states

$$\Phi := tt|ff|(C_i, c_i)|(A, a)|\neg\Phi|\Phi_1 \vee \Phi_2|\Phi_1 \wedge \Phi_2|P \sim p(\varphi)$$

where $c_i \in 2^{AP}$ AP a set of atomic propositions describing situation context (e.g. location: ⟨meeting room⟩, light: ⟨$bright$⟩, occupation: ⟨3⟩, application-running: ⟨power-point⟩), $a \in A$ is a finite set of actions, and $\sim$ is a comparison operator $\sim \in \{\langle, \rangle, \leq, \geq\}$, and p is a probability threshold $p \in [0, 1]$. Path quantifiers as in PCTL are built from one of the temporal modalities: X (next) or U (until) (Table 2). t is a time constraint defining an upper bound on a time interval to describe the duration of a situation, the subsequent transition and when an action will be active.

Quantifiers over Paths	
$A\Phi$ – All	Φ has to hold on all paths starting from the current state.
$E\Phi$ – Exists	There exists at least one path starting from the current state where Φ holds.
Path-Specific Quantifiers	
$G\Phi$ – Globally	Φ has to hold on the entire subsequent path
$F\Phi$ – finally	Φ eventually has to hold
$X\Phi$ – Next	Φ has to hold at the next state
$\Phi U \psi$ – Until	Φ has to hold at least until at some position ψ holds. ψ will be verified in the future

Table 2. Paths quantifiers

Considering Φ a state formula expressed as a pair (C_i, c_i), which describes the type of context and the specific context value in this state (e.g.: location, meeting-room). We leverage these operators to analyze the future context behavior;

- F is the Eventually operator used to verify if a condition ϕ eventually has to hold in any state from s somewhere on a subsequent path in the model.
- G is the Globally operator, and it can be used to check if the condition ϕ holds in every state on all subsequent paths starting in s.
- X is the Next operator: it evaluates a condition ϕ on all immediate successor states to the current state s. It has to hold at the next state (this operator is sometimes noted N instead of X). Since we focus on immediate prediction, we will build a prediction model on this operator in this paper.
- U is the Until operator and expresses that Φ_2 will be verified in the future. And Φ_1 has to hold starting at the current state at least until at some further position Φ_2 holds.

The PCTL state formula $P \sim p(\varphi)$ asserts that, under all schedulers [28], the probability for the event expressed by the path formula φ meets the bound specified by $\sim p$. The probability bounds "$\sim p$" can be understood as quantitative counterparts to the CTL path quantifiers $\exists$ and $\forall$.

B. PCTL Semantic

Definition 3. Let $M = (S, AP, L)$ be a PCTL model, s is a state $\in M$, AP a set of an atomic preposition, L is a labeling function, and ϕ is a PCTL formula. The satisfaction relation is noted as $M,\, s \vDash \phi$.

Let s be a state, $s \in S$ we can define the satisfaction relation for state formulas as follows:

- $M,\, s$ true $\forall s \in S$,
- $M,\, s \vDash c_i \Leftrightarrow c_i \in L(s)$,
- $M,\, s \vDash \neg\phi \Leftrightarrow M,\, s \nvDash \phi$,
- $M,\, s \vDash \phi_1 \wedge \phi_2 \Leftrightarrow M, \vDash \phi_1$ and $s \vDash \phi_2$,
- $M,\, s \vDash \phi_1 \vee \phi_2 \Leftrightarrow M, \vDash \phi_1$ or $s \vDash \phi_2$,
- $M,\, s \vDash P \sim p(\varphi) \Leftrightarrow P\{\pi \in Paths(s) | M,\, \pi \vDash \varphi\} \sim p$.

The satisfaction relation for path formula is defined inductively as follows:

- $M,\, \pi \models X\Phi \Leftrightarrow \pi = s_0 \rightarrow^{a_0,t_0} s_1 \rightarrow^{a_1,t_1} \ldots s_n \rightarrow^{a_{n-1},t_{n-1}} s_n$ and $M,\, s_1 \models \Phi$,
- $M,\, \pi \models \Phi_1 U \Phi_2 \Leftrightarrow \pi = s_0 \rightarrow^{a_0,t_0} s_1 \rightarrow^{a_1,t_1} \ldots s_n \rightarrow^{a_{n-1},t_{n-1}} s_n$ and $\exists k.M$, $s_k \models \Phi_2$ and
- $\forall j < k.M,\, s_j \models \Phi_2$.

C. Labeled Probabilistic Transition Model: Model-LPTM

A pervasive system follows various behavioral patterns depending on user's behavior. Those patterns cannot be described in a deterministic way. Hence, our choice of a probabilistic non-deterministic model. In the following, we give a description of this model and the proposed approach to predicting the next situation using this formalism.

We represent an LPTM model as a transition system which combines probabilistic choice as in Markov chains with a non-deterministic choice. We define the model with a timed probabilistic transition based on models defined in [27, 28]. The model integrates time and action and will be presented as follows.

Definition 4. Let LPTM be a Kripke (S, A, P, L): a labeled transition probabilistic model defined as follows:

- S: a finite set of states where $s \in S$ and $s_{init} \in S$,
- Act: a finite set of actions where $a \in A$ and $A \subseteq Act$,
- L: $S \rightarrow 2^{AP}$ state labeling function assigning to each state one or several atomic prepositions $\in AP$,
- $P \subseteq S \times A \times \mathbb{R}^+ \times Dist(S)$ is the function assigning a probabilistic transition distribution, such that if $(s, \delta t, a, \rho) \in Dist(S)$ and $\delta t > 0$ after a span time $\triangle t$ in a situation s was spent and a is an active $\in A(s)$ then ρ is a point distribution.

As for probabilistic systems, we can introduce paths for timed probabilistic systems except that transitions are now labeled by a (duration, action, distribution) tuple. Each transition is labeled by a tuple $(\delta t, a, \rho) \in Dist(S)$, where:

- δt is the time span between s_i and s_j (Section 3.6),
- $P(s_i, s_j)$ is the probability assigned to the path transition between s_i and s_j (Section 3.4),
- $a \in A(s_i)$ is an action active between two states s_i and s_j (Section 3.5).

Our contribution using this model consists in considering every $s_i \in S$ as described by a set of context parameters $(c_i \in C_i)$ such that $L(s_i) = c_i$ and an action for a transition path with a temporal duration constraint δt.

To avoid transient states, we choose to integrate them as proposals in paths. Thus, the path describes a transient context as an accomplishment action or activity action (see Section 3.5). That can be part of the next state. This makes the modeling more context-aware and proactive.

Using this LPTM, we can formalize the behavior trace and context variation by an infinite state tree like in MDP. The context can be a composite context. The variation of one or several context's element introduces changes on the state. We can describe a pervasive environment according to the user's behaviour with action semantic (Section 3.4), and context variation, at each spatiotemporal interval, we have an active state describing a specific context $s_i \in S$. While the user (e.g.: walking, driving, be, ...) or the environment and the system environment (running process, etc.) act, the context changes and the LPTM moves to the new state $s_j \in S$ expressing the property of new context. This successor state s_j is visited with a probability $p(s_i, s_j)$. Before leaving the current state s_i, the context does not change and stay active for a limited duration of time δt spent in s_i. Example: model (Figure 3).

Explanation: To lead the next situation from the current situation i to the next one j we count:

- $a_{s,n}$ represents an action active for a given state (e.g., a_{01} describes the active action from S_0 to S_1),
- δt_{ij} represents the time span between s_i and s_j (e.g., δt_{01} describes transition duration from S_0 to S_1),
- P_{ij} refers to the transition probability from the situation s_i to the situation s_j such that $\Sigma_j P_{ij} = 1$.

a. Transition Probability

For each transition (S_i, S_j), the transition probability will be:

$$p(s_i, s_j) = P(X_{n+1} = s_j | X_n = s_i) \tag{1}$$

where X_n is the random variable that models the stochastic behavior at the current state and X_{n+1} model the stochastic behavior at the next state.

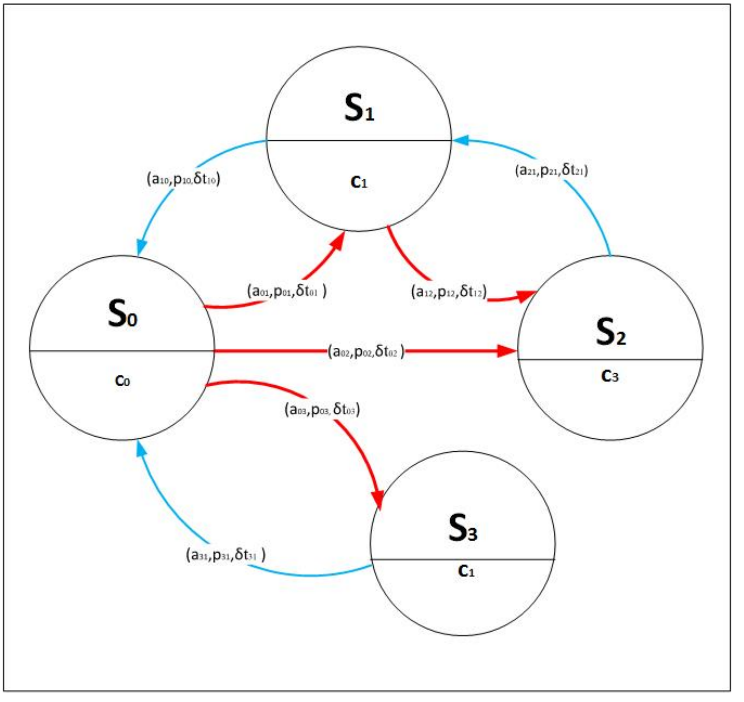

Figure 3. Transition model

Recall that the formulas are defined about a probabilistic structure, as described earlier. While the used structures consist of labeled states and path, they only imply that it is possible to transition from the state at the tail to the state at the head with some non-zero probability.

We express a model as a causal model. In this paper, we assume a dependent relation between current state and the next one. The probabilistic transition

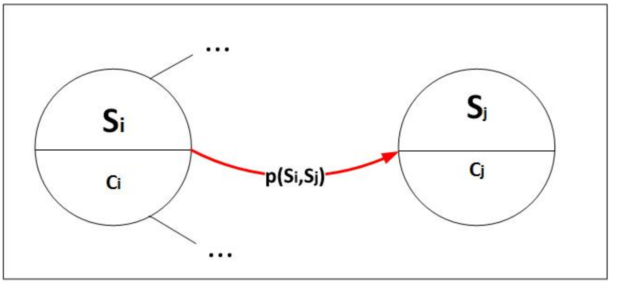

Figure 4. State transition probability

depends only on the current state s_i and s_j is independent of all previous state changes.

The transition probability and the set of prepositions describing contextual feature situations can be estimated and deduced from the history of past trace of state transitions and their linked contextual features.

As in statistic computation, let the transition weight be ω_{ij}, which defines the number of transitions observed from s_i to s_j. The transition probability is calculated as follows:

$$p(s_i, s_j) = P(X_{n+1} = s_j | X_n = s_i) = \frac{\omega_{s_i,s_j}}{\sum_{n \in S} \omega_{s_j,s_n}}. \quad (2)$$

The probability of transition between two states is the ratio of the number of observed state transitions from s_i to s_j to the number of all observed transitions from s_i.

Example: We have a current state s_0 that can lead to any of the immediate next states as in Figure 5 as a distributed probability.

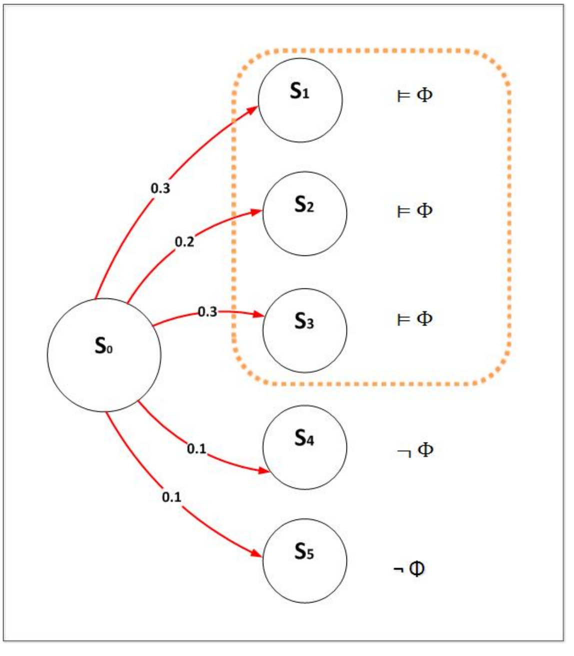

Figure 5. State transition

The probability without any constraint of time or action to lead to any next state when ϕ will be verified as a Next (optimal) in S_1, S_2 and as a Next (all) in S_3 be S_3 or S_1.

b. Action

Observing the system's and user's behavior, we also noted information describing actions which influence a service process and make a change in a situation in the feature context. Based on reference work discussing the linguistics of time and the semantic verbs and time [30, 31], these actions can use the aspectual verbs according to the categories in Table 3.

	Expressivity	Dynamic	Durative	Telic (bound)
Accomplishment	Describing Durative action Ending by a culmination point	Yes	Yes	Yes
Activity	describing durative action	Yes	Yes	No
State	Often durative	No	Yes (temporary state) No (permanent state)	Yes
Achievement	Change of state near punctual duration	Yes	No	Yes

Table 3. The four aspectual categories

In the proposed model we can use accomplishment and activity to describe a transition over a path and a state and achievement in a situation (node).

The computation over the proposed model we use the accomplishment-action on the path because we are reasoning in a dynamic system with a time-bound and we count the durative actions in a bound time during a transition. We can label a graph with state-action and achievement to clearly describe a scenario or an example.

In the proposed model, actions depend on transition and describe a transition over a special path. The set of actions available at $s \in S$ is denoted by $A(s)$. For each action $a \in A(s_i)$, the probabilities can be estimated as other observations from the history of past trace. We count the probability of transitioning from s_i to s_j under the action a, and we denote this probability by $\alpha_a^s i(s_j)$. We refer to [32] for more details about computation in mapping and learning steps.

Example: We have a set $A(s_0) = \{a_1, a_2\}$ and a transition and s_0 can lead to any of the immediate states as in Figure 6.

In this example, the probability next ϕ to occur with any action $a \in A(s_0)$ is

$$\frac{\sum_{s_j \in S \wedge s_j \models \phi} \alpha_a^{s_i}(s_j)}{\sum_{s_j \in S} P(s_j, s_i) \cdot \sum_{s_j \in S \wedge s_j} \alpha_a^{s_i}(s_j)} = 0.45,$$

the optimal next will be the path with a strategy probability ≥ 0.45 in this case that will be the transition (S_0, S_2) under the action a_2.

3.3 Space Time Duration

We will show how we can estimate the time span between s_i and the next s_j. The time was considered in the model as the constraint parameter for states as well as

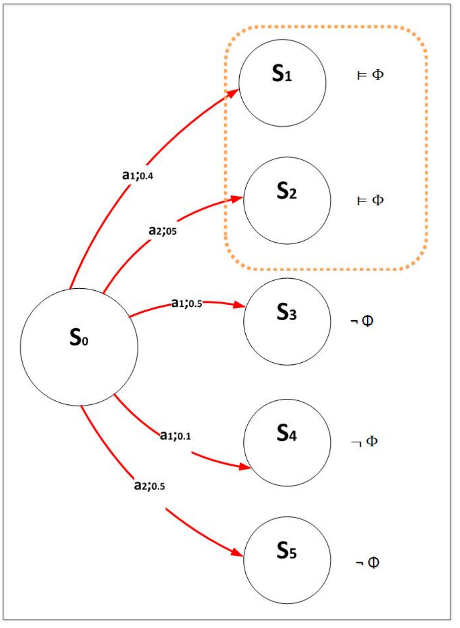

Figure 6. Action transition

transitions (path), as described in a previous contextual definition and model. Every situation has a time interval describing its start time and end time which can be useful as a learning data base [24, 8].

We express the time span as a probability function where μ and α are the mean and standard deviation values, calculated from the time span. In order to limit the computation, we consider in the current work only the observation falling with standard deviation

$$f(\delta t_{ij}, (\mu, \sigma)) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\delta t_{ij} - \mu}{\sigma}\right)^2}. \tag{3}$$

Figure 7 gives an example of transition time span: the typical time span falls in the following range.

We model the time span as a random variable D_n expressing the time spent between s_i and s_j. To figure out D_n, we observe the time periods spent between consecutive states transitions, and we associate an individual distribution to every transition between s_i and s_j. Formally the distribution can be presented as:

$$\Delta_{ij}(\delta t) = P(D_n = \delta t | X_{n+1} = S_j, X_n = S_i). \tag{4}$$

The cumulative distribution is given by $\triangle_{ij}$ which is given as $\int_0^b \triangle_{ij}(\delta t)\, \mathrm{d}\delta t$ and can be computed as the sum of probabilities associated with consecutive intervals up to a desired upper time bound b. The probability of a time span to

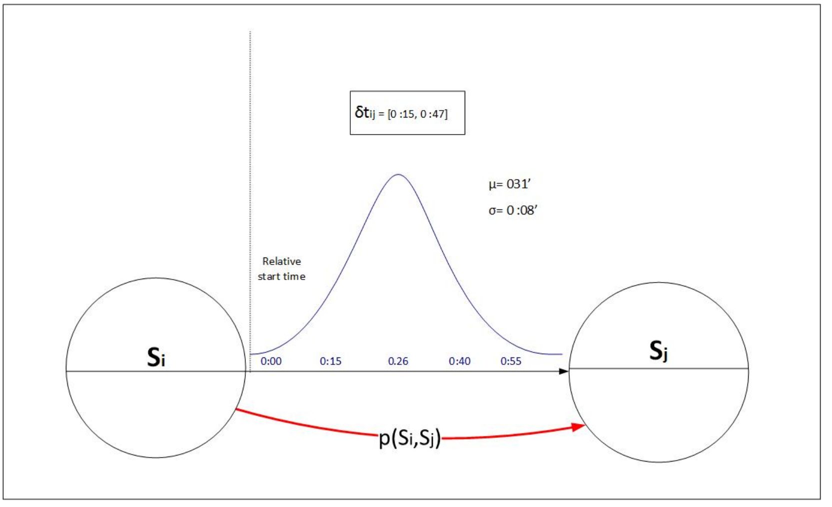

Figure 7. Transition span time

lie within the interval $[a, b]$ can be derived from the cumulative distribution as $\int_0^b \triangle_{ij}(\delta t)\,\mathrm{d}\delta t$.

3.4 Immediate-Context Prediction Processing

The state space can be traversed by going from one state to the next as allowed by transitions among states. The resulting series of visited states (path) models one possible spatiotemporal behavior of context. For context prediction we start at the state $s_i \in S$ occupied in the real world, and we evaluate the possible path starting at s_i and leading to the next state s_j. The state and path follow the PCTL semantic, as explained in Section 3.2.

In the proposed model we can evaluate a satisfaction relation for the path formula as follows:

$$X_{n+1} \leftarrow \operatorname{argmax}_{X_{n+1}} P(X_{n+1} = s_j | X_n = s_i, a \in A(s_i)). \tag{5}$$

The path formula φ is satisfied after $\triangle t$ unit of time elapsed in a situation s and under an action a if and only if the probability $P((s, a, \triangle t) \vDash \varphi)$ satisfies the threshold $\sim p$.

In our case, we need to be able to verify that a given state satisfies the context's state preposition $\phi = (C_i, c_i)$ (as described in Section 3.3). We also need to consider the temporal operator Next $P \sim p[X\phi]$ and define its probability computation.

Using a PCTL, we can investigate the reachability properties using the Next operator, evaluating a condition state formula ϕ, expressed over the contextual information (C_i, c_i), on all immediate successor states s_j of the current situation s_i.

Using this reasoning, the system is able to predict a variety of information about the context (e.g. the next location, the next activity, what time the user finishes work, at what time the next meeting starts, what is the optimal strategy to lead the next situation, etc.). The high threshold probabilities according to a special action describing a transition reduce the number of false prediction prepositions and make the prediction more efficient and more context-aware.

We can derive the prediction based on next operator in PCTL as explained in the next subsection using the verification algorithm in a model checking based on symbolic method [33, 34].

3.5 Computation for PCTL Next Operator

In this paper, we focus on immediate prediction. Thus, we will only use the next operator. In future work, we might extend the proposed approach with the two more temporal operators: (i) Until: $P \sim p[\phi_1 \cup \phi_2]$ and (ii) Bounded Until: $P \sim p[\phi_1 \cup^{\leq k} \phi_2]$ which can be useful for a long-term prediction.

The Next operator restricts the space of satisfaction property of path formula φ to the immediate successor the next state s_j of the current state s_i. We need to determine the Next (optimal) $\phi = P_{max=?}([X\phi])$ which is the maximum probability satisfying Next ϕ.

$$X_{n+1} \leftarrow \text{argmax}_{X_{n+1}} P(X_{n+1} = s_j | X_n = s_i). \tag{6}$$

Or the all Next (all) $\phi = P_{\bowtie\rho}([X\phi])$; here we can find all the policies that satisfy the next state with ϕ property, where:

$$\begin{aligned} & P(X_{n+1} = s_j | X_n = s_i, a \in A(s_i)) && (7)\\ & \qquad = P_{\langle a \rangle}\left(X_{\triangle t}^{\leq \delta t_{ij}}(\phi)\right) && (8)\\ & \qquad = \frac{\sum_{s_j \in S \wedge s_j \models \phi} P(s_j, s_i). \sum_{s_j \in S \wedge s_j \models \phi} \alpha_a^{s_i}(s_j). \int_{\triangle t}^{\triangle t + \delta t_{ij}} T_{ij}(\delta t)\, \mathrm{d}\delta t}{\sum_{s_j \in S} P(s_j, s_i). \sum_{s_j \in S \wedge s_j} \alpha_a^{s_i}(s_j). \int_{\triangle t}^{\infty} T_{ij}(\delta t)\, \mathrm{d}\delta t}. && (9) \end{aligned}$$

The optimization function $log(P(\phi|\lambda))$ is proposed to avoid data overflow in the computation of feed forward probability.

The prediction approach is based on the traces contained in the stochastic user model. The traces are used as a search space of possible context changes. Information about the recent sensed context changes (current state's context) is used to condition the prediction on what the optimal Next might be expected in the immediate future. Using a model based on statistical knowledge, the predictions in the proposed approach, work as a scanning process in a stochastic transition system to find the Next verifying the property expressed in the formula. A component diagram of the prediction model can be represented, as shown in Figure 8.

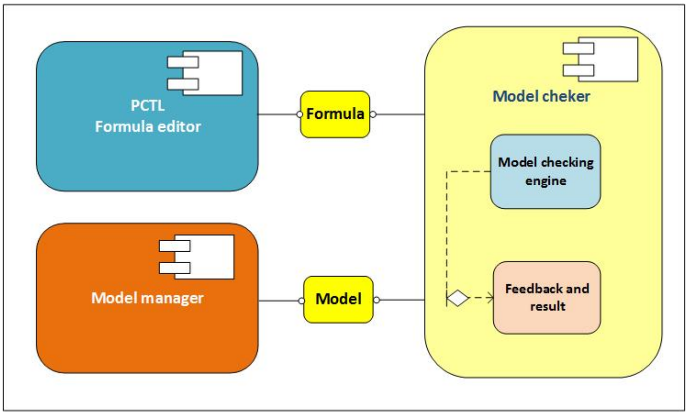

Figure 8. Component diagram of LPTM system

3.6 Use Case and Test

In this section, we present the experimental results for the proposed model. Before getting into the evaluation of the prediction model, we describe the data set we used [35, 36].

We use a real-world context traces from Domus smart home case study. The Domus smart home is one-bedroom apartment mounted inside the University of Sherbrooke. The apartment is equipped with different types of sensors. During the experiments, users have participated to evaluate the early morning routines, which correspond to the basic everyday tasks during the morning. The routine describes morning activities as follow: wake up, toileting, preparing breakfast, having breakfast and other activities. We use this study case to predict the Next activity. The activities we consider in the simulation are as follows: wake up, use toilet, preparing breakfast, having breakfast.

As a simulator tool, we use Petri nets, that means formal models of information flow which support timing specifications and a non-deterministic behavior for more details about tools we refer to [37]. We first model the prediction model as shown in Figure 9.

The model is composed mainly of:

- Generation: this module generates the current context and constraints as a random choice.
- Get related activity: the module gives the activity probability (Section 3.4).
- Get related activity Action: the module determines the action probability (Section 3.5).
- Get related activity time: the module defines the time span probability (Section 3.6).

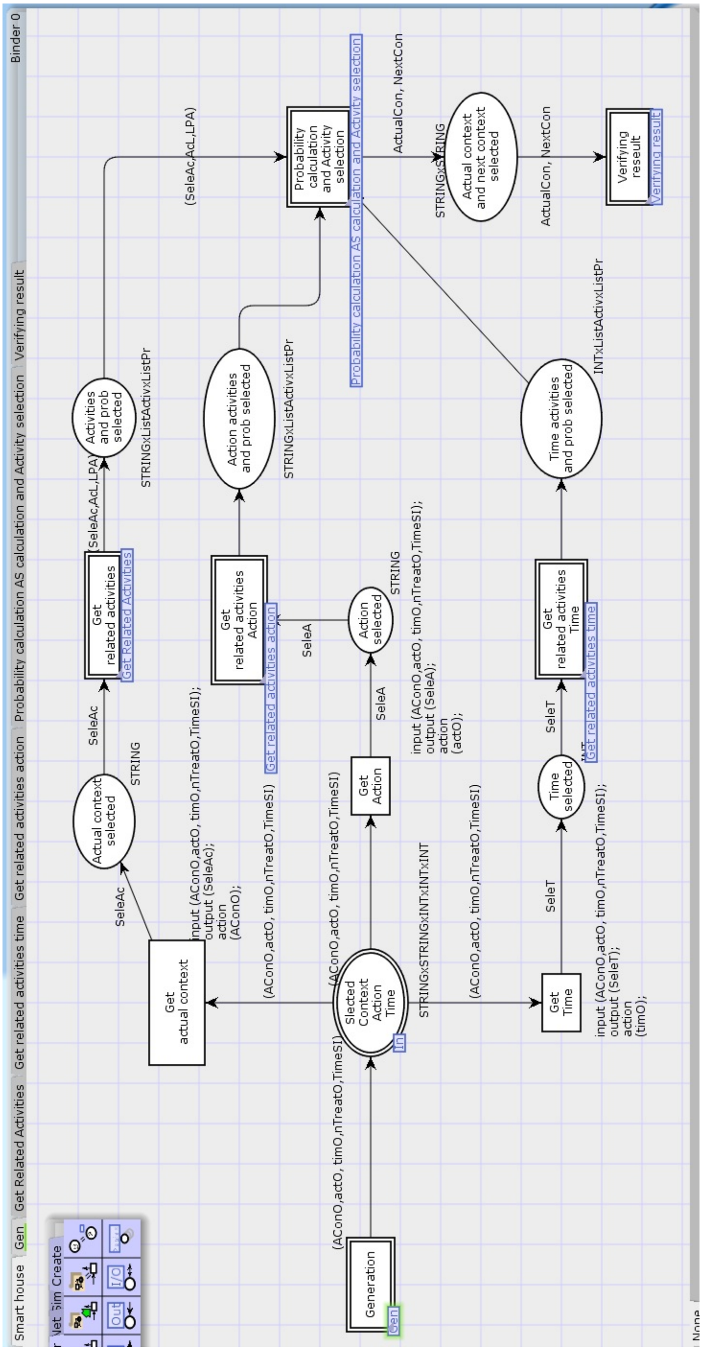

Figure 9. General view of prediction model

- Probability calculation: the module gives the probability of the most probable Next activity (Section 3.7).

The transition between different activities are learned based on the LPTM trace Model as shown in Figure 10.

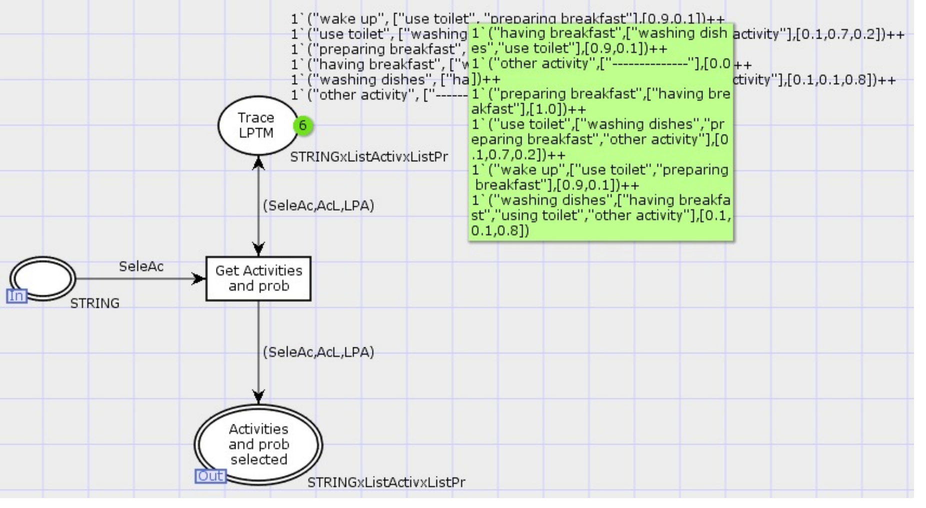

Figure 10. Activities transitions information

To recognize the next activity, we generate a random for a variety of activity (context value) time and action as shown in Figure 10. When an event is detected, this module generates automatically the actual context, the action and the transition time (Figure 10).

As we mentioned before, the Dumas data set that we used for actual context is limited to having breakfast, other activities, preparing breakfast, use toilet, wake up, washing dishes, for action is limited to (close door, open door) and for time is limited to $(5, 10, 15, 20, 25, 30, 35, 40, 45, 50)$ The outputs of this module are:

- The actual context used as input by the transition Get related activity to determine the activity probability.
- The action used as input by the transition Get related activity Action to determine the activity probability.
- The transition time used as input by the transition Get related activity time to determine the time span probability.

The transitions between different activities are learned based on the LPTM trace model, as shown in Figure 11. The input of this module is the actual activity selected randomly by the generator (Figure 10). According to this activity the transition "Get activities and Prob" selects the adequate activities and probabilities from the place Trace LPTM. As output of this module, we have three parameters: the actual activity presented by the variable SeleAC as string, the list of activities presented by the list ACL and the list of probabilities presented by the list LPA.

For instance, if the actual activity is "Wake up" (SeleAC) then the output of this module will be ("wake up", ["use toilet", "preparing breakfast"], $[0.9, 0.1]$) the different probabilities in the place "Trace LPTM" are computed from dataset DUMAS. After we get all probabilities, the transition "Probability calculation and Activity selection" (Figure 9) determines the next activity (Section 3.7). Then the role of the transition "verifying result" to test the result generated by the transition "probability calculation and Activity selection" The actual activity and the next activity are the input of this module. We compare the result obtained by the values in the place "DB-Real-Flow Evidence".

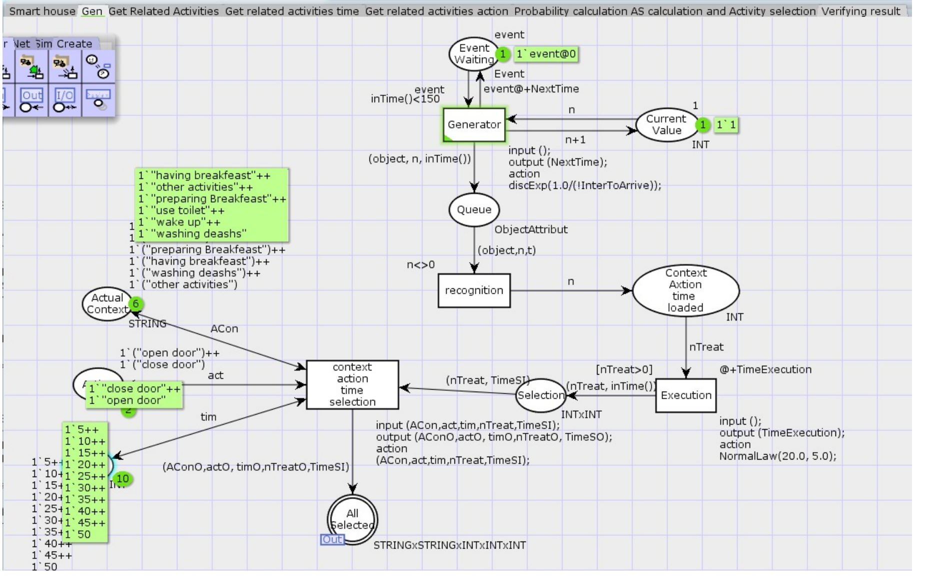

Figure 11. Action-time generators

Finally, after getting the Next activity identified, we evaluate the results based on real flow evidence as shown in Figure 12.

The diagram in Figure 13 resumes the prediction results for each activity. The average of the prediction model was 65 %, we also get 78 % in some activity, as shown in the following diagram.

3.7 Result Discussion

The accuracy criteria can usually be ranging from low/worst performance to high/best performance, depending on the capacity of the approach to be effective in

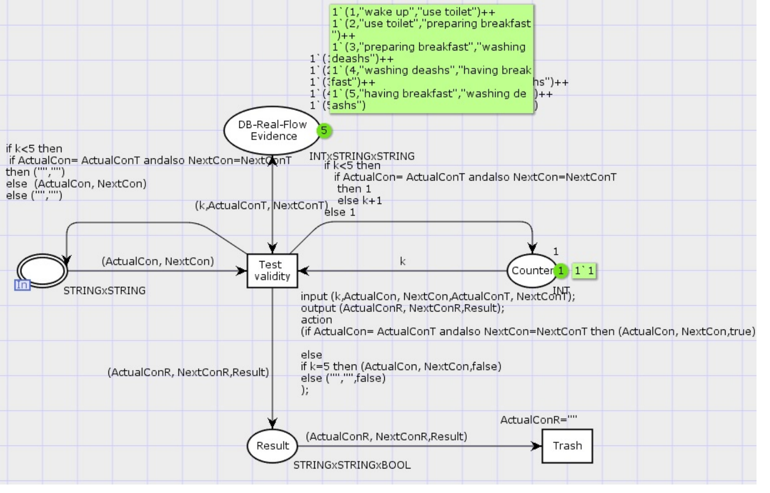

Figure 12. Verifying results Next activity

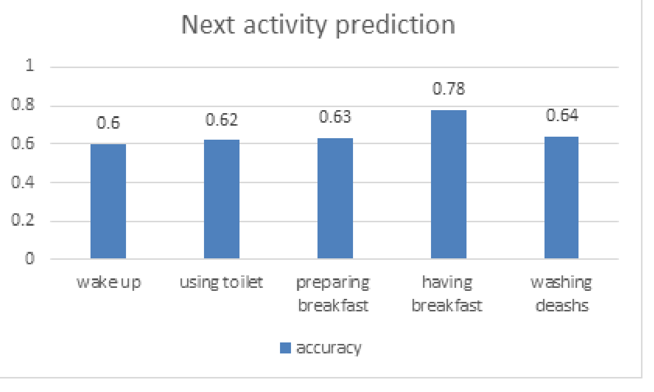

Figure 13. The activities prediction accuracy

a ubiquitous environment. Our model is in the high rang performance comparing to other context prediction model tested in real data. Using Lezi algorithm [38], the authors obtained prediction rate nearing 47 %. Using Markov and Bayes [39], the prediction accuracy achieved was 70 % to 80 %. In Najar's work [19], the system was based on the implementation of matching algorithm the prediction had a result that neared 60 %. Sigg et al. [40] have used ARMA in an analytical test, and we disregarded it for our work because is applicable only for a numerical data set. Da Rosa et al. [5] obtained an average accuracy of 60 % for the alignment method and 72 % for the Semi-Markov approach, and the model does not make a distinction between low or high context level. Föll et al. [41] used CSL and Semi-Markov-

Chain, they achieved 87 %, and they concluded their work noting using PCTL could increase the expressive power of the formal core. Which constitutes an essential and valuable contribution presented in our model including the semantic of action and span time duration as a probability function, to improve the expressivity and obtain better precision for the prediction.

Figure 14 summarizes the comparison of the existing approach and our proposed approach, regarding the different evaluation criteria [11].

Approach		Reasonning criteria						Data criteria		
		Accuracy	Tolerence to messing value	Evolvability	Speed of run time	Observability	Prior knowledge inference	Data loss in preprocessing	Numeric data	No numeric data
Sequence		*	*	*	*	□	✓	✓	X	✓
Markov		**	**	***	**	□	✓	✓	X	✓
Bayesian networks		***	***	****	****	■	✓	✓	✓	✓
Neuronal networks		***	*	**	**	□	X	X	✓	X
Branch method		**	***	***	**	■	✓	✓	X	✓
Trajectory prolongation	Interpolation	**	*	*	****	□	X	X	✓	X
	Approximation		*	*	**	□	✓	X	✓	X
Expert system		**	**	****	***	□	✓	✓	X	✓
Space theory		***	**	***	**	□	✓	✓	✓	✓
Data mining algorithm		***	**	**	**	□	✓	✓	✓	✓
Similarity and Semantic similarity		***	*	**	**	□	✓	✓	✓	✓
LPTM-Model		****	****	****	**	□	✓	X	✓	✓

Legend

Performance	Observability	Confirmation
**** Best performance * Worst performance	■ Black box □ White box	No X Yes ✓

Figure 14. Comparative analysis of approaches

4 CONCLUSION AND FUTURE WORK

The prediction of future context has become a central element in pervasive systems to provide proactive context-awareness adaptation. However, the effective deployment of a context-aware prediction is still limited due to a semantic gap between the data provided by the physical sensing devices and the necessary information to predict future behavior of the system and its users. In this paper, we have demonstrated how formal methods could be adapted to offer a formal ground to reduce this semantic gap and provide improved expressiveness via the PCTL logic. And therefore, verify reachability a next-state in the future. Introducing the constraints of time and action adds logic-based expressiveness and provides a clear tracing and learning model. Thus, increasing the effectiveness of probabilistic measures.

In this paper, we present a new formal approach using probabilistic temporal logic and model checking to provide an immediate prediction. The proposed approach allows a formal expressivity of prediction. This is useful in pervasive computing systems to deal with their inherently heterogeneous nature. The model offers a real-time ability to discover a future context on multidimensional space and can handle a general context in low or high level. Adopting a PCTL as formalism provides better expressivity to describe the nondeterministic nature of human behavior which can provide an efficient prediction and consequently offer adequate proactivity, fitting with the user's needs. In fact, PCTL can be used to specify properties of probabilistic timed automata adding the semantic of action in our model, Thus, we think it will be useful to specify properties of probabilistic timed labelled automata. Regarding the complexity of model checking with probabilistic timed labelled automata, we consider this in a separate future work after more research in this direction.

In future work, we will extend the current research to include the long-term prediction and possibly discuss a generic framework that can support the proposed prediction model to automating proactive adaptation based on predicted context. We will try to investigate more, the issue of semantic in action to be able to provide a more expressive model, inducing cognitive and linguistic support.

REFERENCES

[1] Salfner, F.—Lenk, M.—Malek, M.: A Survey of Online Failure Prediction Methods. ACM Computing Surveys (CSUR), Vol. 42, 2010, No. 3, Art. No. 10, doi: 10.1145/1670679.1670680.

[2] Boytsov, A.: Context Reasoning, Context Prediction and Proactive Adaptation in Pervasive Computing Systems. Thesis, Luleå Tekniska Universitet, 2011.

[3] Boytsov, A.—Zaslavsky, A.: Extending Context Spaces Theory by Proactive Adaptation. In: Balandin, S., Dunaytsev, R., Koucheryavy, Y. (Eds.): Smart Spaces and Next Generation Wired/Wireless Networking (ruSMART 2010, NEW2AN 2010). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6294, 2010, pp. 1–12, doi: 10.1007/978-3-642-14891-0_1.

[4] Boytsov, A.—Zaslavsky, A.—Synnes, K.: Extending Context Spaces Theory by Predicting Run-Time Context. In: Balandin, S., Moltchanov, D., Koucheryavy, Y. (Eds.): Smart Spaces and Next-Generation Wired/Wireless Networking (ruSMART 2009, NEW2AN 2009). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 5764, 2009, pp. 8–21, doi: 10.1007/978-3-642-04190-7_2.

[5] da Rosa, J. H.—Barbosa, J. L. V.—Ribeiro, G. D.: ORACON: An Adaptive Model for Context Prediction. Expert Systems with Applications, Vol. 45, 2016, pp. 56–70, doi: 10.1016/j.eswa.2015.09.016.

[6] Sigg, S.—Haseloff, S.—David, K.: An Alignment Approach for Context Prediction Tasks in UbiComp Environments. IEEE Pervasive Computing, Vol. 9, 2010, No. 4, pp. 90–97, doi: 10.1109/mprv.2010.23.

[7] Zaguia, A.—Tadj, C.—Ramdane-Cherif, A.: Context-Based Method Using Bayesian Network in Multimodal Fission System. International Journal of Computational Intelligence Systems, Vol. 8, 2015, No. 6, pp. 1076–1090, doi: 10.21307/ijssis-2017-824.

[8] Ameyed, D.—Miraoui, M.—Tadj, C.: A Spatiotemporal Context Definition for Service Adaptation Prediction in a Pervasive Computing Environment. arXiv preprint arXiv:1505.01071, 2015.

[9] Ameyed, D.—Miraoui, M.—Tadj, C.: Spatiotemporal Context Modelling in Pervasive Context-Aware Computing Environment: A Logic Perspective. International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 7, 2016, No. 4, pp. 407–414, doi: 10.14569/ijacsa.2016.070454.

[10] Barbosa, J. L. V.: Ubiquitous Computing: Applications and Research Opportunities. 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2015, doi: 10.1109/iccic.2015.7435625.

[11] Ameyed, D.—Miraoui, M.—Tadj, C.: A Survey of Prediction Approach in Pervasive Computing. International Journal of Scientific and Engineering Research, Vol. 6, 2015, No. 5, pp. 306–316.

[12] David, K.—Kusber, R.—Lau, S. L.—Sigg, S.—Ziebart, B.: 3rd Workshop on Recent Advances in Behavior Prediction and Pro-Active Pervasive Computing. Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '14 Adjunct), 2014, pp. 415–420, doi: 10.1145/2638728.2641675.

[13] Mayrhofer, R.: Context Prediction Based on Context Histories: Expected Benefits, Issues and Current State-of-the-Art. Cognitive Science Research Paper, University of Sussex CSRP, 2005, Vol. 577, p. 31.

[14] Sigg, S.—Haseloff, S.—David, K.: A Novel Approach to Context Prediction in Ubicomp Environments. 2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications, 2006, doi: 10.1109/pimrc.2006.254051.

[15] Meiners, M.—Zaplata, S.—Lamersdorf, W.: Structured Context Prediction: A Generic Approach. In: Eliassen, F., Kapitza, R. (Eds.): Distributed Applications and Interoperable Systems (DAIS 2010). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6115, 2010, pp. 84–97, doi: 10.1007/978-3-642-13645-0_7.

[16] Padovitz, A.—Loke, S. W.—Zaslavsky, A.: Towards a Theory of Context Spaces. In Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004, doi: 10.1109/percomw.2004.1276902.

[17] Boytsov, A.—Zaslavsky, A.: Context Prediction in Pervasive Computing Systems: Achievements and Challenges. In: Burstein, F., Brézillon, P., Zaslavsky, A. (Eds.): Supporting Real-Time Decision-Making. Springer, Boston, MA, Annals of Information Systems, Vol. 13, 2011, pp. 35–63, doi: 10.1007/978-1-4419-7406-8_3.

[18] Abowd, G. D.—Dey, A. K.—Brown, P. J.—Davies, N.—Smith, M.—Steggles, P.: Towards a Better Understanding of Context and Context-Awareness.

In: Gellersen, H. W. (Ed.): Handheld and Ubiquitous Computing (HUC 1999). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1707, 1999, pp. 304–307, doi: 10.1007/3-540-48157-5_29.

[19] Najar, S.—Kirsch Pinheiro, M.—Souveyet, C.: A New Approach for Service Discovery and Prediction on Pervasive Information System. Procedia Computer Science, Vol. 32, 2014, pp. 421–428, doi: 10.1016/j.procs.2014.05.443.

[20] Föll, S.—Herrmann, K.—Rothermel, K.: PreCon – Expressive Context Prediction Using Stochastic Model Checking. In: Hsu, C. H., Yang, L. T., Ma, J., Zhu, C. (Eds.): Ubiquitous Intelligence and Computing (UIC 2011). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6905, 2011, pp. 350–364, doi: 10.1007/978-3-642-23641-9_29.

[21] Magherini, T.—Fantechi, A.—Nugent, C. D.—Vicario, E.: Using Temporal Logic and Model Checking in Automated Recognition of Human Activities for Ambient-Assisted Living. IEEE Transactions on Human-Machine Systems, Vol. 43, 2013, No. 6, pp. 509–521, doi: 10.1109/tsmc.2013.2283661.

[22] Wagner, A.—Barbosa, J. L. V.—Barbosa, D. N. F.: A Model for Profile Management Applied to Ubiquitous Learning Environments. Expert Systems with Applications, Vol. 41, 2014, No. 4, Part 2, pp. 2023–2034, doi: 10.1016/j.eswa.2013.08.098.

[23] Miraoui, M.: Dynamic Adaptation in Ubiquitous Services: A Conceptual Architecture. In: Ramanathan, R., Raja, K. (Eds.): Handbook of Research on Architectural Trends in Service-Driven Computing, Chapter 7. IGI Global, 2014, pp. 160–180, doi: 10.4018/978-1-4666-6178-3.ch007.

[24] Bohn, J.—Coroamă, V.—Langheinrich, M.—Mattern, F.—Rohs, M.: Social, Economic, and Ethical Implications of Ambient Intelligence and Ubiquitous Computing. In: Weber, W., Rabaey, J. M., Aarts, E. (Eds.): Ambient Intelligence. Springer, Berlin, Heidelberg, 2005, pp. 5–29, doi: 10.1007/3-540-27139-2_2.

[25] Schnoebelen, P.: The Complexity of Temporal Logic Model Checking. Advances in Modal Logic, Vol. 4, 2002, pp. 393–436.

[26] Clarke, E.—Grumberg, O.—Peled, D.: A Model Checking. MIT Press, 1999.

[27] Kwiatkowska, M.—Norman, G.—Sproston, J.—Wang, F.: Symbolic Model Checking for Probabilistic Timed Automata. Information and Computation, Vol. 205, 2007, No. 7, pp. 1027–1077, doi: 10.1016/j.ic.2007.01.004.

[28] Baier, C.—de Alfaro, L.—Forejt, V.—Kwiatkowska, M.: Model Checking Probabilistic Systems. In: Clarke, E., Henzinger, T., Veith, H., Bloem, R. (Eds.): Handbook of Model Checking. Chapter 28. Springer, 2018, pp. 963–1000, doi: 10.1007/978-3-319-10575-8_28.

[29] Wolovick, N.—Johr, S.: A Characterization of Meaningful Schedulers for Continuous-Time Markov Decision Processes. In: Asarin, E., Bouyer, P. (Eds.): Formal Modeling and Analysis of Timed Systems (FORMATS 2006). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4202, 2006, pp. 352–367, doi: 10.1007/11867340_25.

[30] Vendler, Z.: Verbs and Times. The Philosophical Review, Vol. 66, 1957, No. 2, pp. 143–160, doi: 10.2307/2182371.

[31] Ratté, S.—Ratté, S.: Computational Event Structures – Part I. 1994.

[32] Lahijanian, M.—Andersson, S. B.—Belta, C.: Control of Markov Decision Processes from PCTL Specifications. Proceedings of the 2011 American Control Conference, 2011, IEEE, doi: 10.1109/acc.2011.5990952.

[33] Kwiatkowska, M.—Norman, G.—Parker, D.: Stochastic Model Checking. In: Bernardo, M., Hillston, J. (Eds.): Formal Methods for Performance Evaluation (SFM 2007). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4486, pp. 220–270, doi: 10.1007/978-3-540-72522-0_6.

[34] Kwiatkowska, M.—Norman, G.—Sproston, J.: PCTL Model Checking of Symbolic Probabilistic Systems. Technical report CSR-03-2, University of Birmingham, School of Computer Science, 2003.

[35] Kadouche, R.—Pigot, H.—Abdulrazaka, B.—Giroux, S.: Support Vector Machines for Inhabitant Identification in Smart Houses. In: Yu, Z., Liscano, R., Chen, G., Zhang, D., Zhou, X. (Eds.): Ubiquitous Intelligence and Computing (UIC 2010). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6406, 2010, pp. 83–95, doi: 10.1007/978-3-642-16355-5_9.

[36] Chikhaoui, B.—Wang, S.—Pigot, H.: A New Algorithm Based on Sequential Pattern Mining for Person Identification in Ubiquitous Environments. KDD Workshop on Knowledge Discovery from Sensor Data, 2010.

[37] Liu, Y.—Miao, H.-K.—Zeng, H.-W.—Ma, Y.—Liu, P.: Nondeterministic Probabilistic Petri Net – A New Method to Study Qualitative and Quantitative Behaviors of System. Journal of Computer Science and Technology, Vol. 28, 2013, No. 1, pp. 203–216, doi: 10.1007/s11390-013-1323-7.

[38] Gopalratnam, K.—Cook, D. J.: Active LeZi: An Incremental Parsing Algorithm for Sequential Prediction. International Journal on Artificial Intelligence Tools, Vol. 13, 2004, No. 4, pp. 917–929, doi: 10.1142/s0218213004001892.

[39] Kaowthumrong, K.—Lebsack, J.—Han, R.: Automated Selection of the Active Device in Interactive Multi-Device Smart Spaces. Workshop at UbiComp, 2002, Citeseer.

[40] Sigg, S.: Development of a Novel Context Prediction Algorithm and Analysis of Context Prediction Schemes. Kassel University Press GmbH, 2008.

[41] Föll, S.: State-Based Context Prediction in Mobile Systems. Dissertation, University of Stuttgart, 2014.

Darine Ameyed is currently Post-Doctoral Researcher in Synchromedia Laboratory in the École de Téchnologie Supérieure (ÉTS). She received her Ph.D. in engineering from the École de Téchnologie Supérieure (ÉTS), University of Quebec, Montreal, Canada, in 2016. She obtained her M.Sc. in digital art and technology from the University Rennes 2 (UR2), University of Upper Brittany, France, in 2010. She obtained her M.Sc. in multimedia engineering from the University Paris-Est Marne-la-Vallée (UPEM), University of Paris 11, France, in 2008. She received her B.Sc. in software and management from the Institut Supérieur de Gestion (ISG), University of Tunis, Tunisia, in 2005. Her research interests include pervasive and ubiquitous computing, context-aware systems, predictive modelling, activity recognition, human centred computing, cognitive IoT. She has published papers in national and international conferences and journals. She had also several years of industry experience in Canada and Africa in the areas of software engineering, mobile computing, ubiquitous computing, CIoT, and management.

Moeiz Miraoui received his Ph.D. (2009) in computer science from the École de Technologie Supérieure (ÉTS), University of Quebec, Montreal, Canada. He obtained his M.Sc. in computer science from the National Engineering School of Sfax, University of Sfax, Tunisia, in 2003. He received his B.Sc. Ing. in computer science from the National School for Computer Science, University of Tunis, Tunisia, in 1999. He began working as Assistant Professor in computer science at High Institute of Applied Science and Technology, University of Gafsa, Tunisia, in 2009. He has supervised several M.Sc. students and co-supervises Ph.D. projects. His research interests include pervasive and ubiquitous computing, context-aware systems and smart spaces. He has many published papers in national and international conferences and journals, and he is a reviewer and a technical committee member of several journals and conferences.

Atef Zaguia is currently Assistant Professor in College of Computers and Information Technology, Taif University, Kingdom of Saudi Arabia. He received his Ph.D. and M.Sc. degrees in computer science from École de Téchnologie Supérieure (ÉTS), University of Quebec, Montreal, Canada, and his Bachelor degree in computer engineering from Ottawa University. He spent one year at École de Téchnologie Supérieure (ÉTS), University of Quebec, Montreal, Canada, as a postdoctoral fellow. He was working on developing application for newborn cry-based diagnosis system with the integration of interaction context, financial by Bill and Melinda Gates Foundation. His research interests include a multimodal system, pervasive and ubiquitous computing and context-aware systems. He has published papers in national and international conferences and journals. He was in Program Committee for the Tenth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies UBICOMM 2016, Venice, Italy.

Fehmi Jaafar is Adjunct Professor in the Faculty of Management of Concordia University of Edmonton, Canada. Previously, he was postdoctoral fellow at Queen's University and Polytechnique Montreal, and Research Scientist in computer security and software engineering at the Research and Development Team of Ubitrak Inc. He received his Ph.D. from the Department of Computers Sciences and Operations Research of the Université de Montréal in Quebec, Canada, 2013. His research interests include the analysis of software quality and evolution, cyber security, and techniques and tools for mining software repositories. He also had several years of industry experience in Canada and Africa in the areas of software engineering, computer security, web and mobile computing, and business process management.

Chakib Tadj serves as Professor at École de Technologie Supérieure (ÉTS), University of Quebec, Montreal, Canada. He received his Ph.D. degree in signal and image processing from ENST Paris, Paris, France, in 1995. His main research interests include signal processing, speech recognition, pervasive computing and multimodal systems.