Reasoning on Uncertainty in Smart Environments

Alencar Machado[1,2], Vinícius Maran[3], Iara Augustin[2], João Carlos Lima[2],
Leandro Krug Wives[4] and José Palazzo Moreira de Oliveira[4]

[1]*Colégio Politécnico, Universidade Federal de Santa Maria (UFSM), Santa Maria, Brazil*
[2]*Programa de Pós-Graduação em Informática, Universidade Federal de Santa Maria (UFSM), Santa Maria, Brazil*
[3]*Coordenadoria Acadêmica, Universidade Federal de Santa Maria (UFSM), Santa Maria, Brazil*
[4]*PPGC Instituto de Informática, Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, Brazil*

Keywords: Semantic Web, Uncertainty, Smart Environments, Probabilistic Ontologies.

Abstract: Currently, There Is a Convergence of Systems for Smart Environments and Uncertainty Reasoning. Context Models Are Being Proposed to Support the Detection of Situations in These Environments. However, Reasoning to Detect Situations Taking into Account Uncertainty Presents a Great Challenge. This Paper Describes a Contextual Model based on Semantic Web Technologies That Can Deal with Uncertainty. This Framework May Be Used to Detect Unwanted Situations with a Certain Grade of Contextual Uncertainty. the Model Was Evaluated in a Scenario Reasoning over Uncertain Data to Predict Unwanted or Perhaps Dangerous Situations.

1 INTRODUCTION

Homes are becoming Intelligent Environments, able to assist people who live into it. These systems are planned to act according to the user profile, and with complex physical environment, where objects are added, updated or changed of location. The user profile also changes over time. For instance, users may suffer from different diseases during their lives, and these can affect the interaction with the objects of their residence. Cognitive problems, such as forgetfulness, can sometimes put the user into an unwanted or perhaps dangerous situation; for instance, forgetting the stove on after preparing a meal.

Systems for Ambient Assisted Living need to interpret the context in which the user lives to be able to act in advance. Using ontologies to represent the contextual model is the most complete and expressive way to support reasoning about the user context for intelligent systems (Strang and Linnhoff-Popien, 2004) (Bettini et al., 2010). The ontologies have some constructors that give support the reasoning about the model domain. Some studies show contextual models for different domains (Sixsmith et al., 2009) (Tazari et al., 2010). These efforts search to model the user's context and support systems for decision making on situations of interest. Some works try to support systems in reasoning about uncertainty (Coronato, 2012) (Coronato and De Pietro 2013) (Rasch et al., 2011) (Forkan et al., 2015). These works present contextual models using ontologies that are implemented in Ontology Web Language – Description Logic (OWL-DL).

The main limitation for reasoning about uncertainty in OWL-DL is the conceptual foundation of DL, a subset of First Order Logic (FOL). FOL defines sentences, axioms, that are always true logical statements about the domain to which they are representing (Costa et al., 2011) (Laskey, 2008). However, for systems that seek to detect situations and make decisions in real environments, where data may have an error rate, low but present, it becomes imperative to support detection of situations and decision-making on uncertain data. Also the information obtained from the sensors must be interpreted and this interpretation normally is surrounded with uncertainty.

To manage these challenges, some research efforts try to manipulate information modelled in ontologies and submit this information to prediction algorithms using fuzzy logic, neural networks, or Bayesian networks (Coronato and De Pietro 2013). These strategies have purely statistical results as the

Machado, A., Maran, V., Augustin, I., Lima, J., Wives, L. and Oliveira, J.
Reasoning on Uncertainty in Smart Environments.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 2*, pages 240-250
ISBN: 978-989-758-187-8

prediction is made without the semantic information contained in the ontological representation. Consequently, it is necessary that the ontological contextual models represent the context uncertainty.

Also, it is necessary to have a native support for reasoning about uncertainty within the context model. This approach gives more possibility to the detection of situations and decision making made with incomplete information.

This paper presents a context model using for reasoning over uncertainty in smart environments. More specifically, we employed a Multi-Entity Bayesian Network (MEBN) for the developing of semantic model to support the prediction of unwanted situations in smart environments. The model is evaluated through empirical experiments using a case study where an unwanted situation is simulated. In the case study is presented the use of the context model which supports reasoning about uncertainty. It is shown as the model supports the automatic generation of Bayesian Networks in a specific situation depending on the available evidence within the ontology.

This paper is organized as follows. Section 2 discusses the background and related work. Section 3 presents the model for developing the context of AAL with reasoning with uncertainty in smart environments. Section 4 describes the developed case study. Finally, Section 5 discusses our conclusions and future work.

2 BACKGROUND AND RELATED WORK

The user interaction with devices in a real-world environment is uncertain by nature. This communication is influenced by several factors, such as the knowledge that the user has about using a device, the cognitive ability, and the humour of the user, among others.

There are big challenges for the design of AAL systems. Currently, the context aware area is consolidating systems for smart environments. Context can be understood as "*the environment in which the system operates*" (Ye et al., 2011). Dey and Abowd (1999) characterize *context* as the situation of an entity in an environment. The concept of *situation* is utilized to characterize the state of the user environment.

Ye, Stevenson and Dobson (2011) define *situation* as the abstraction of the events occurring in the real world that are derived from the context and hypotheses about how the observed context relates to factors of interest. Systems that manipulate the context detecting situations of interest through events generated by user's interaction must handle uncertainty to assist correctly users in their living environment.

2.1 Reasoning on Uncertainty

Reasoning for detection of situation and decision making with uncertain data comprises some shortcomings related to modeling with regard to the modeler conception of the world. The modeling process may have misconceptions and possibly will be addressed through consistency checks of the ontologies. Hence, this fact must be taken into consideration and can be treated through probabilistic reasoning.

Probabilistic reasoning might support the processing of uncertainty. It is possible to make predictions of future situations taking uncertainty into account. Many works that focuses on the prediction phase has been published and they present algorithms to identify the future with an acceptable accuracy (Paganello and Giuli, 2011).

Ontologies provide a range of features that search for represent the environmental context in a broad and expressive form. Context modeling using ontologies is currently done through the use of the Ontology Web Language (OWL). Among its variations (Lite, DL and Full), the most widely used by the possibility of decidability is the Description Logics (DL). The interpretation of a theory determines the definition of each constant, predicate and symbolic function in relation to the area. Each symbolic constant denotes a specific entity; each predicate denotes a group containing entities that the predicate holds, and each symbolic function is a function defined in the domain. The logical sequences of a set of axioms consist of sentences that are always true in all interpretations, also called true sentences. Due to sentences be always interpreted as logical assertions, DL is not suitable for areas where there are uncertainties in relations among the concepts (Laskey, 2008).

One way to model uncertainty is the use of probability, and a suitable model for existing domains is the Bayesian Networks (BN). These are Directed Acyclic Graphs representing a distribution function of joint probabilities of variables in a domain of interest. Each Bayesian network consists of nodes (random variables) and edges connecting these nodes. These links represent the influence from one node (ancestor) in relation to another node (descendent) generating a directed arc. Each node has a Conditional Probability Table (CPT) to calculate the influence of a parent node "*x*" in relation to its influenced node "*y*", and the joint probability distribution is measured by the influence of every parent in relation to a leaf node (Friedman et al., 1997).

For Semantic Web applications, BNs have the potential to provide a powerful, compact structure for probabilistic inference mechanisms. However, BNs have some key limitations. The first is that the number of variables must be known in advance (i.e. number of nodes is fixed). However, many domains require reasoning about numbers and types of related entities, where the relationships among entities cannot be specified in advance or are uncertain. The second limitation is that the language used to specify BNs is not powerfull enough to express significant problems with repeated structure. The third one is that a BN is a directed acyclic graph, and hence no native support for recursion is provided (Costa et al., 2011).

2.2 Probabilistic Ontologies

The area of probabilistic ontologies appeared with the objective of using the expressive power of the First-Order Logic and the treatment of uncertainty supported by Bayesian Networks. In general, there are two approaches to the generation of probabilistic ontologies. The first consists in representing uncertainty by probability values described as annotations, such as (Yang and Calmet, 2005). However, annotate ontology with numerical probability is not enough, and some information is lost with the lack of representation capturing the structural constraints and dependencies between nodes (Laskey, 2008). The second alternative consists of using a First-Order Probabilistic Language, which combines aspects of probabilistic representation with first-order logic (Howard and Stumptner, 2014).

According to Costa (2005), a probabilistic ontology must be able to properly represent: the (i) types of entities that exist in the field; the (ii) properties of these entities; the (iii) relationships between entities; the (iv) processes and events happening with these entities; the (v) statistical regularities that characterize the domain; the (vi) inconclusive knowledge, ambiguous, incomplete, unreliable related to domain entities; and (vii) uncertainty over all previous forms of knowledge. It should be noted that the term entity refers to any concept that can be described and reasoned in an application domain. Probabilistic ontologies are used to describe comprehensively the knowledge of a domain and associated uncertainty, structured and shareable, preferably in a format that can be read and processed by the computer (Fenz, 2012).

Howard & Stumpter (2014) compare those languages (First-Order Probabilistic Languages) in relation to (i) aspects for handling uncertainty, (ii) structural support related to types of inheritance (iii) types of fields on which the language may represent and (iv) reasoning techniques on a group of entities and relationships. An overview of this comparison is presented in Table 1.

In Table 1, attribute means uncertainty about the attributes of entities and relationships. Numeric indicates the uncertainty of numerical data entities in a domain. Reference means uncertainty about the relationship between domain entities. Existence means uncertainty about the existence (or not) of entities and their relationships in the area. Types mean when an entity of interest is identified, but it can be one or more of possible subtypes. Analyzing data present in Table 1, the languages Probabilistic Relational Models (PRMs), Object-Oriented Probabilistic Relational Modeling Language (OPRML) e Multi-Entity Bayesian Networks (MEBNs) are compared, and in this paper we choose to use the MEBNs language to represent uncertainty.

Table 1: Languages for Uncertainty Representation.

		PRMs	MEBNs	OPRML
Uncertainty	Attribute	X	X	X
	Numeric	X	X	X
	Reference	X	X	X
	Identity	X	X	
	Existence	X	X	X
	Type		X	
Inheritance	Simple	X		X
	Multiple		X	
Domain	Static	X	X	X
	Recursive	X	X	X
	Dynamic		X	X

Adapted from Horward & Stumpter (2014).

To make possible the use of MEBN in Semantic Web an OWL extension was created through an upper ontology called Probabilistic Ontology Web Language (PR-OWL). That extension expresses a probability distribution on interpretations of any first-order theory. PR-OWL was built to be interoperable with non-probabilistic ontologies. However, the probabilistic definitions of an ontology have to form a theory about the fragments of the complete or partial valid world (Carvalho et al., 2013).

In Figure 1, the concepts of the upper ontology are presented, where ellipses represent general classes while arrows represent the main relations between these classes. A probabilistic ontology must have at least an individual's *MTheory* class, which is formed by a group of *MFrags* that collectively form a valid *MTheory*. In the next section, we discuss the Multi-Entity Bayesian Network theory (Carvalho et al., 2013).

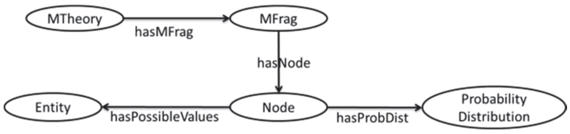

Figure 1: PR-OWL Adapted from Costa (2005).

2.3 Multi-Entity Bayesian Network

Multi-Entity Bayesian Network (MEBN) represents the world as composed of entities that have attributes and which are associated with other entities. Knowledge about the attributes of the entities and their relationships are represented as a collection of MEBN fragments (*MFrags*) organized in MEBN theories (*MTheories*). An *MFrag* represents a distribution of conditional probability for instances of random variables about their parents (parent node) in the same *MFrag*. An MEBN theory is a set of *MFrags* that collectively satisfy consistency constraints, ensuring the existence of a single joint probability distribution on instances of the random variables represented in each *MFrag*. MEBN integrates the semantics of the standard theoretical model of first-order logic with random variables, as formalized in Bayesian Networks (Howard; Stumptner, 2014).

Such as in a BN, one *MFrag* contains nodes that represent random variables arranged in a directed graph whose edges represent relations of direct dependence. An isolated *MFrag* can be compared with a standard BN with known values to their root nodes and local distributions of its nodes that are not root. A node in an *MFrag* may have a list of arguments in parentheses. These arguments are placeholders for the authorities in the field (Costa, 2005).

According to Laskey (2008), an *MFrag* consists of three types of nodes: (i) the residing nodes (object property in OWL-DL) have local distributions that define how their probabilities depend on the values of their parents in the graph. In a complete MEBN theory, each resident node has exactly one *MFrag* where their local distribution is set; (ii) the input and context nodes can influence the distribution of the resident nodes, but their distributions are set in their own *MFrags*. Finally, (iii) context nodes represent conditions that must be satisfied for the influences, and local distributions of an *MFrag* can be applied. These conditions are boolean values, which may have true, false or absurd values.

An MEBN does not specify a standard for Conditional Probability Table (CPT). However, as in a default BN, a CPT summarizes statistical regularities that characterize a domain. These regularities are captured and coded in a knowledge base using a combination of expert opinions and/or learning from observation. For more information about MEBN and PR-OWL, we suggest reading the following works (Laskey, 2008) (Costa, 2005) (Carvalho et al., 2013).

Most projects focused on Ambient Assisted Living or, more specifically in Smart Homes (SH) are interested in proposing models to provide services. A reactive requirement drives the execution of services, always after a fact has occurred. Therefore, they do not show proactive behaviour to adapt to the user environment. Related works are the ones shown in Table 2: Soprano (Sixsmith et al., 2009), Persona (Tazari et al., 2010), SM4ALL (Rasch et al., 2011), Uranus (Coronato, 2012), CoCaMAAL (Forkan et al., 2015).

Table 2: Related Work.

	Per	Sop	SM4	Ura	CoC
AAL	X	X	X	X	X
SH	X	X	X		
Sensor	X	X	X	X	X
Event	X	X			X
Situation	X	X		X	X
Action	X	X	X		X
Activity					X
Uncertainty					

These works show a semantic context model, but it is observed that these models have a reduced expressiveness. Those who address the uncertainty show a hybrid model to generate probability using techniques without using Semantic Web technologies. Therefore, it is important to develop a model to support reasoning about uncertainty in the AAL domain because none of the related work addresses these characteristics, fully supported by Semantic Web technologies, to sensitive systems on the Situation-Awareness in Smart Environments.

3 DEALING WITH UNCERTAINTY IN SMART ENVIRONMENTS

Systems for smart environments need to know the user environment, and where necessary, implement assistance actions. Currently, the main source for real-time data collection is data obtained from sensors. They collect raw data without semantic characterization and with error rates adding uncertainty in the collected data. Therefore, it becomes necessary to consider this data associated with data from other entities such as people, rooms' characteristics, and electrical networks state, among others. It is possible to group data to generate useful information from a higher level, i.e., detecting of an environmental or user situation, as cold or emergency. In this paper, the vision of the environment perceived by the system is obtained from the data captured by the sensors; these inputs are aggregated with environmental contextual entities generating useful information. The actions triggered by the system for the environment are realized by Web services, which are associated with objects like smart phones, televisions, microwaves ovens, and others in the living environment. In Figure 2, is presented the information and decision flow.

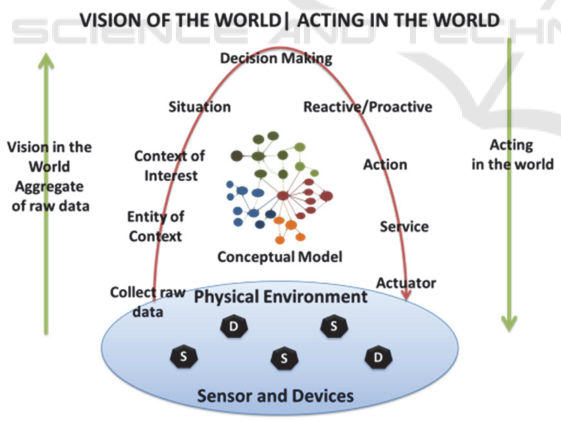

Figure 2: Adapted from Machado et. al, 2013.

The system starts collecting raw data from sensors and aggregates the information associated with those entities to generate higher-level information that is used to characterize the state (*situation*) of the environment. With this characterization, the system can do a decision to act in the environment using the capabilities (*services*) provided by the available web services automating the environment according to the situations of interest and user preferences.

3.1 Ontology Network for AAL

Other works on situation-awareness for smart environments are directed to modeling concepts that relate to the situation itself, paying little or no attention to the modeling of other concepts of an intelligent environment. In this article, we try to approximate the home automation model (essentially describes semantic relations between the structure of the physical environment) with the user model and higher-level information such as events, situations and actions.

The ontology network for AAL was described using Web Ontology Language using the Protégé software. The probabilistic fragments that make up the Reference Model for Systems to Predictive Situations-awareness in AAL was developed using UnBBayes software. Intra networks relations were established with OWL-DL *owl*:*subclass* and *owl:import* resource and other relationships were implemented using DOOR ontology (Allocca; D'Aquin, 2009).

A simplified form of each modelled network is presented in Figure 3; however, for easy viewing some entities and their relationships were omitted. This structure may be modified to incorporate new concepts, allowing the inclusion of new entities in different domains. The objective is to construct a model that describes an automated home environment, fully controlled by a middleware to support a home care environment.

Analysing Figure 3, the user performs *Actions* (*Human Action*). These result in *External Events* collected by the system. Events *start* and *finish* the *Current Situation* involving the user in the current time. The same events influence a *Predictive situation* that may involve the *User* in the future.

Using the information about *Current* and *Predictive Situation*, the system can select *Automated Actions* to handle situations of interest. For example, if is necessary to handle the situation including the interaction with a *User*, the system should choose an *Automated Action Type Regarding Person*.

This action will be performed by a functionality provided by a device of the type *Interaction with Person*. This functionality must be sensitive to the disability presented by the *User*. The *Automated Actions* produce *Internal Events* and analysing these, the system can detect if the *Current* and *Future Situation* change or will change in relation to a *User*.

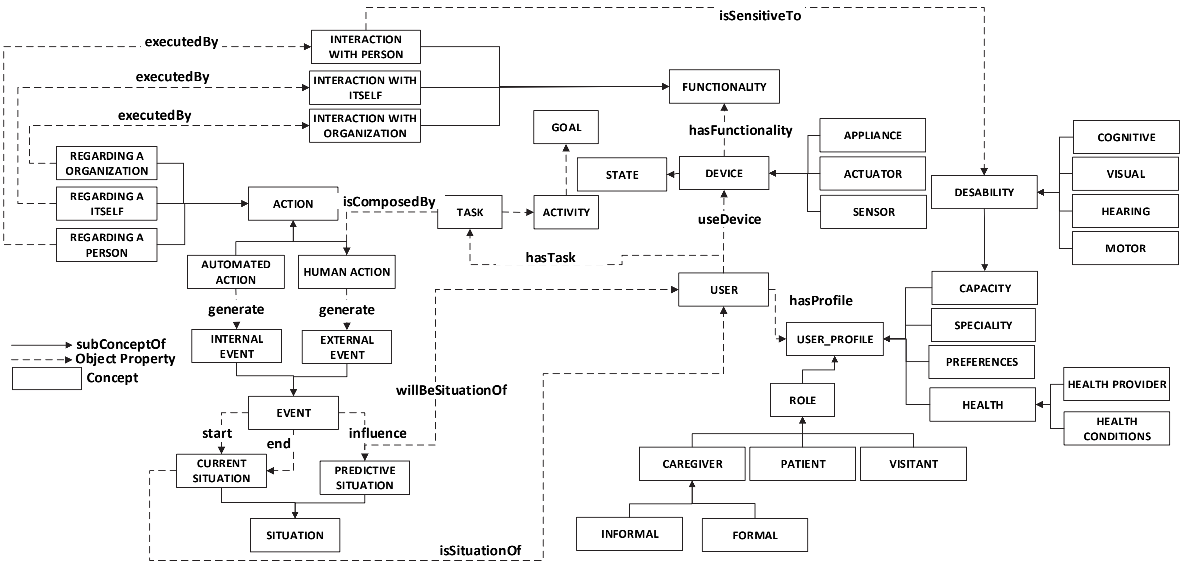

Figure 3: Ontology Network for AAL Environments.

3.2 Reference Model for Predictive Situations in Smart Environments

Represent uncertainty in ontologies is not the same as to build a probabilistic system. In this particular case, the *probabilistic part* refers to the semantic relationships modelled using the PR-OWL settings that collectively form a MEBN theory. There is no need for all the relations of an ontology to be probabilistic, however the parts modelled with PR-OWL extension should form a valid MEBN theory. The semantic relationships that extend PR-OWL resources are shown in Figure 4.

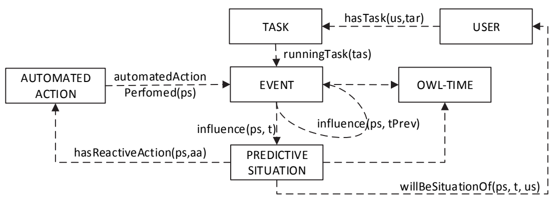

Figure 4: Probabilistic Ontology for proactive domain.

In this figure the probabilistic ontology for predictive situations is represented graphically. In this probabilistic ontology, the semantic relationship contains parameters derived from entity instances of ontology. The definition of the semantic relationship with parameters is inherited from the PR-OWL model. In this case, the instances of entities are random variables that will feed the reference model to *Predictive Situation* and form a valid theory. In this probabilistic ontology, events are affected when the user is involved by some *Task* running (*runningTask*(tar)) and/or *Automated Actions* taken in relation to Predictive situation (*AutomatedActionPerformed* (*ps*)).

The recursion for the model is promoted by the Temporal Entity defined in OWL-TIME ontology (Hobbs et al., 2004). This concept can be used as a discrete concept, so representing subsequent steps orderable of *"T0"* to *"Tn"* rather than continuous scales. Therefore, an event can influence a predictive situation at an earlier time by the relationship *influence(ps, tPrev)* and influence a predictive situation at time *t*. The *willBeSituationOf(ps, t, us)* gives the probability of a Predictive situation *ps* on a time *t* involving user *us*. Therefore, it can perform prediction of a situation over time.

So the question that the probability part of the ontology must answer is: "*what is the probability of a situation at a specific time, involving the User in his living environment?*"

In the Figure 4 is show the MBEN theory of reference for predictive situation-aware systems. The systems using this model can answer the question by resident node *willBeSituationOf(ps, t, us)*. This reference model is a repeatable structure (template) that generates Bayesian Networks according to user situation. The Specific Situation Bayesian Networks are generated according to the semantic relationships that are linked to the user's instance within the ontology. Therefore, at runtime, when exists a need to generate a Bayesian network, the structure of the network is dynamically generated according to the reference model and the user context that are represented in the ontology. The model is composed of fragments, described below, and the random variables related to the context nodes are: "*tar*" to task, "*us*" to User, "*aa*" for Automated Action, "*ev*" for Event, "*ps*" for Predictive Situation and "*tPrev*" and "*t*" for orderable Discrete Times (instances of OWL-TIME), the resident nodes will be described as *<resident node name> = {states}*.

The *Task* MFrag present the probability of a specific task to be in execution in the environment, and this fact is represented by runningTask*(tas) = {True, False}* resident node. The Automated Action *MFrag* presents the probability of an automated action to be executed in relation to a predictive situation, and this fact is represented by *automatedActionPerformed(ps) = {Automated Action}* resident node. The TaskUser *MFrag* describes, through the resident node *hasTask(us,tas)={True, False}*, the probability of a user be involved by a task. The Predictive Situation *MFrag* generates the local probability distribution of a particular predictive situation involving a user at a specific time by the resident node *willBeSituationOf(ps, t, us) = {True, False}*. Such distribution is calculated through of the influence of the resident node *influence (ps, t)={events}*, whereas the states of resident node *influence* must be instances of events. So this way is possible to calculate the probability of a particular event happens when being influenced by task and automated action. This context node applies the probability distribution that an Event influences a predictive situation in time *t*, when affected after by the input nodes *runningTask(tas)*, *automatedActionPerformed(ps)* and its own distribution in a previous time *influence(ps, tPrev)*.

This reference model is a repeatable structure that instantiates specific Bayesian Network according to existing evidence. The reasoning process on the Reference Model for Predictive situations is intended to the generation (queries) of Specific Situation Bayesian Network (SSBN). Determining the values (probability) of a set of queries the systems can query the probability of a situation will be situation of a User at a specific time in the future. Thus, the system has the possibility to choose what proactive actions to be triggered in relation to the probability values generated by virtue of a query generated of the *Specific Situation Bayesian Network.*

4 CASE STUDY

A scenario is the complete description of a contextualized user’s routine. This case study is based in a scenario that demonstrates how the probabilistic model developed in this article is applied in smart environments. To analyze the model, it was used a *Pervasive Application* and a prototype of Situation as a Service (SIaaS) middleware described in Machado et, al. 2014.

Thus, the scenario described is typical of an Ambient Assisted Living environment, consisting of an unwanted situation. Different aspects of user’s interaction with the object’s residence are identified to generate events that can determine the beginning, the end or influence the rise of unwanted situations. These features result in the execution of proactive actions to handle this situation. The scenario is intended to generate data for the detection of unwanted situations and actions performed to deal with these situations, and the events generated (effects of these actions).

Imagine John’s, a 75 years old citizen who has some aging associated diseases such as diabetes, hypertension and lightweight dementia. John's residence consists of a living room and kitchen, among other space. This type of patient tends to forget in what activity is immersed, it is common to start an activity and forget that he was doing another, for example, cooking and watching television, or even confuse the time of day and go to sleep. Therefore, the family of John buys a Pervasive Application to assist John in daily activities. The focus of this scenario is the activity of cooking, where John interacts with the stove device (task use stove). According Blasco et al. (2014) older people are a group with highest vulnerability to accidents, especially in their homes. The vast majority of domestic accidents are related to activities in the kitchen: kitchen utensils, cutlery and appliances are the most dangerous utensils. As a result of these accidents, older people lose confidence in their abilities, lowering their self-esteem and, consequently, in many cases, deciding to live in a nursing home.

Therefore, to avoid such features, in John's kitchen was installed a smart stove with the following features: (i) identify which user used / are using the stove, by detection of digital; only after this step the stove frees up resources, e.g. turning on the stove; and (ii) which was the last time (time) in which a person neared the stove, or even (iii) the functionality to automatically turn off the stove.

Therefore, imagine that John is watching television in the living room, but moves to the kitchen and decide to cook. John organizes the preparations, he turns on and put a pot on the stove and, after it, he listen news on TV that is of his interest. So, he comes out of the kitchen and back into the living room, sitting on sofa in front of the television. An application that seeks to assist users affected by the state of Senile in their daily activities should interfere in everyday life as little as possible.

Therefore, in this scenario, the application has interest in being notified if John forgot the stove turned on, so putting him and his family in an unwanted situation as a dangerous situation in cooking when using the stove. So, the application can make *decision for trigger proactive actions and to manipulate the unwanted situations.* Table 3 presents the actions and events produced by the scenario.

Table 3: Actions of appPervCook.

Actions
ac1: Warn User; **ac2:** Notify Caregiver that stove is turned on; **ac3:** Turn off the stove automatically
Events
ev1: User forgot stove on; **ev2:** Stove off; **ev2.1**: Sensor detected presence near to the stove

The ev1 event starts the unwanted situation, and the ev2 and ev2.1 events finish the situation. The actions can be used with a reactive or a proactive way. To handle this situation, a reactive behaviour of the application could be making decision to turn off the stove automatically or alert the caregiver to perform this action or wait more time. For a proactive behaviour, the application could have the ability to predict whether John, when turn on the stove, he will forget this on. This characteristic is promoted by the *Reference Model for Predictive situation-aware system*. The local probabilistic distribution for each MEBN fragment must be generated with the help of an expert in Senile (in a real scenario), or by machine learning. This model can be started with historical data of people affected by this disease, or even after John turned on the stove a few times. Thus, making it possible to identify a behavioral pattern and identify the probability of John forget the stove on while using it. In the fictitious scenario, if John turn on the stove, there is possibility of forget it on. In Table 4, is presented the distribution for each fragment MEBN.

The resident node *runningTask(tas)*, describes the probability that there is a running task in the environment. For this node, the article used values that do not affect other nodes residents (50% true and 50% false). The node *hasTask(us, tas)* establishes that 92% of the time there running task the user is involved. The resident node *automatedActionPerformed(ps)* shows that directly warn John (ac1) was executed 15% of the time. Notify the caregiver (ac2) by 50% and automatically turn off the stove (ac3) by 35%.

Table 4: Local probability distribution for resident nodes of appPervCook.

Resident: **automatedActionPerformed(ps)** [ac1=**0.15**, ac2=**0.5**, ac3=**0.35**]	*Resident:* **runningTask(tas)** [true=**0.5**, false=**0.5**]
Resident: **willBeSituationOf *(ps,t)*** If any **ps** has (*influence* = ev1) [true=**0.99**, false=**0.01**] else If any **ps** has (*influence* = ev2) [true=**0.01**, false=**0.99**] If any **ps** has (*influence* =ev21) [true=**0.45**,false=**0.55**]else[true=**0.5**,false=**0.5**]	*Resident:* **hasTask (us,tas)** [true=**0.92**, false=**0.08**]
Resident: **influence (ps,t)** If any **ps** has (*automatedActionPerformed* = ac1) If any **tas** has (*runningTask* = false) [ev1=**0.05**, ev2=**0.35**, ev21=**0.6**] else [ev1=**0.75**, ev2=**0.05**, ev21=**0.2**]]else If any **ps** has (*automatedActionPerformed* = ac2) If any **tas** has (*runningTask* = false) [ev1=**0.02**, ev2=**0.38**, ev21=**0.6**] else [ev1=**0.45**, ev2=**0.05**, ev21=**0.5**]]else If any **ps** has (*automatedActionPerformed* = ac3) If any **tas** has (*runningTask* = false) [ev1=**0**, ev2=**1**, ev21=**0**] else [ev1=**0**, ev2=**1**, ev21=**0**]]else [ev1=**0.34**, ev2=**0.33** ev21=**0.33**]	

The local distribution to the resident node *influence(ps, t)* describes how *Automatically Executed Actions* and *Running Task* influenced the establishment (arise) of the Events “forgot stove turned on (ev1)", "Stove turned off (v2)" and " Sensor detected presence near to the stove (ev2.1) ".

Therefore, if: (a) John is warned (ac1) and the task is not running, John forgets the stove turned in 5% of cases, do not forget 35% and there is presence near to the stove 60%. If the task is running, John forgets stove turned in 75%, does not forget 5% and there is presence near to the stove 20%; (b) Caregiver notified (ac2) and there is no running task, John forgets the stove turned on 2% of time, do not forget 38% and there is presence near to the stove 60%; If running task, he forgets in 45% of time, do not forget to 5% and presence by 50%; (c) Stove turned off automatically (ac3), in 100% of cases, the stove turned off (v2) and, for a default distribution, are distributed on average 33.33% for all events.

The resident node *influence(ps, t)* applies your local distribution probability into *willBeSituationOf(ps, t, us)* node at time t, so if John forgot the stove (ev1), then 99% cases the dangerous situation is valid and only 1% false. If John turned off the stove (v2), then there is 99% chance of dangerous situations does not exist and, if it was detected presence near stove (ev2.1), then there is a dangerous situation in 45% of time. Thus demonstrating that even if someone neared the stove (checking or not the cooking activity), there is a probability of danger.

Using the structure of the reference model defined in Section 3.1 and local probability distributions given in Table 4, the system can generates the Specific Situation Bayesian Network for the scenario as following. Figure 5 presents the SSBN not recursive (only T0 time). It was adding the evidences that John is using the smart stove and the dangerous situations will be situation of John at T0 on 46,39% of chance.

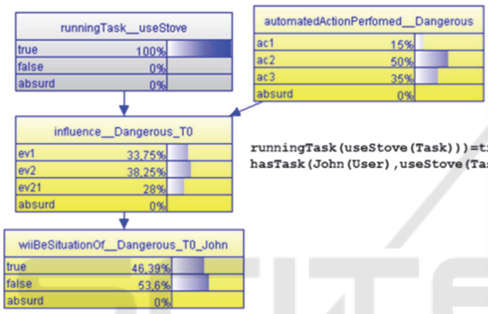

Figure 5: SSBN at T0 and John using Stove.

In Figure 6, it was added the T1 time (ontology instance). Therefore, it was possible to determinate that John will be involved by Dangerous Situation at T1 with 46,4%.

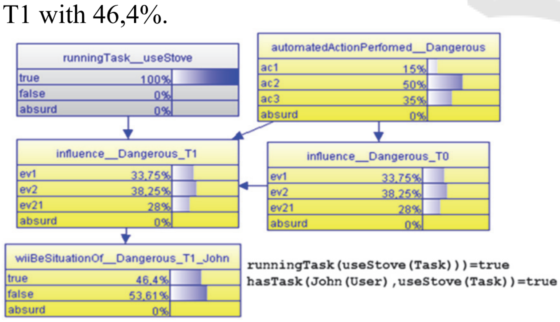

Figure 6: SSBN from T0 to T1 and John using Stove.

The difference between the last net and it was 0.01%, but if it is relevant, depends on the interpretation of an expert on the detected situation.

In Figure 7 is presented that there is an evidence that John, at T0, forgot the stove turned off (ev1), so in T1 there is a probability of 72,5% of chance that a Dangerous Situation will be the situation of John in T1.

In Figure 8 it was used more evidences. Thus, the network was used to answer the following query: *"What is the probability that John to be in a dangerous situation in T2?"*. In the Figure, is presented the SSBN with the evidences (axioms when applied in SSBN result in gray nodes) provided by the scenario. Therefore, the current time is T1, John is involved by tasks watching tv and use stove, these tasks comprise the cooking activity.

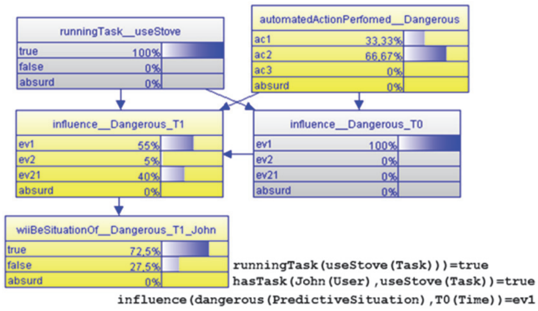

Figure 7: SSBN from T0 to T1, John using Stove and Ev1 detected at T0.

The local probability distribution of the Table 4 describe that John is involved by some task, it is a positive influence to dangerous situation. There is evidence that John at T0 (previous time) forgot the stove on (ev1 = 100%) and the current time (T1) already was detected presence of some people near the stove (ev21 = 100%). With these evidences, the system has the intention to know if in T2 John will be involved by a Dangerous Situation. According to figure 8, there is 69,8% of chance that John is in Dangerous Situation in T2.

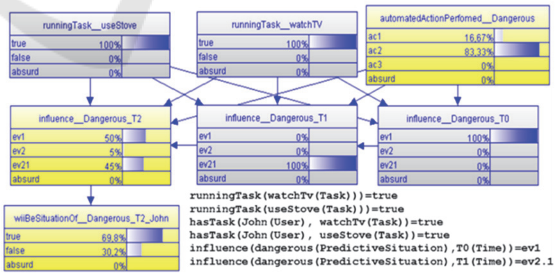

Figure 8: SSBN with the evidences.

5 CONCLUSIONS

Systems for intelligent environments may become proactive when technologies support the reasoning over uncertainty in runtime. Multi-Entity Bayesian Network made these characteristics possible through of the generation of Specific Situation Bayesian

Network. This paper presented the Reference Model for Predictive Situation-Awareness for give dynamic support in relation to reasoning over uncertainty for detection unwanted situation with Multi-Entity Bayesian Network Theory. This model provides essential support for the prediction of situations in real environments. The reference model enables the Bayesian network structures as well as the probability values for predictions are generated at runtime. Therefore, the SSBN are generates on the moment that the user is living in their residence. Ordinary Bayesian Network are not dynamic this way, because need of an expert to model their structures.

The contributions of this paper include the use of Semantic Web Technologies for reasoning about uncertainty, as well as the reference model for predicting unwanted situations. Further work is the identification of a top ontology to increase the coverage context model and using the reference model for different scenarios.

REFERENCES

Allocca, C, D'Aquin, M. 2009. Door: Towards a formalization of ontology relations. In. International Conference on Knowledge and Ontology Development, 2009. *Proceedings of the International Conference on Knowleadge and Ontology Development*. Madera: [s.n.], p. 13-20.

Bettini, C., Brdiczkab, O., Henricksen, K., Iindulzkad, J., Nicklase, D., Ranganathanf, A., Rriboni, D. 2010. *A survey of Context Modelling and Reasoning Techniques. Pervasive and Mobile Computing*, 6(2), p. 161-180.

Blasco, R., Marco, Á., Casas, R., Cirujano, D., Picking, R. 2014. *A smart kitchen for ambient assisted living. Sensors*, v. 14, n. 1, p. 1629-1653.

Carvalho, R.; Laskey, K.; Costa, P. C. 2013. PR-OWL 2.0–bridging the gap to OWL semantics. *In: Uncertainty Reasoning for the Semantic Web II.* Springer Berlin Heidelberg, p. 1-18.

Coronato, A. 2012. Uranus: *A middleware architecture for dependable AAL and vital signs monitoring applications*. Sensors, v. 12, n. 3, p. 3145-3161.

Coronato, A., De Pietro, G. 2013. Situation awareness in applications of ambient assisted living for cognitive impaired people. *Mobile Networks and Applications*, v. 18, n. 3, p. 444-453.

Costa, P. C. G., Carvalho, R. N., Laskey, K. B., Park, C. 2011. Evaluating uncertainty representation and reasoning in HLF systems. In: Information Fusion (FUSION), 2011 *Proceedings of the 14th International Conference on. IEEE.* p. 1-8.

Costa, P. C. 2005. Bayesian semantics for the Semantic Web. George Mason University Departament of Systems Engineering and Operations Research, George Mason University: Fairfax, VA, USA. p. 312.

Dey A, Abowd G. 1999. The context toolkit: Aiding the development of context-enabled applications. In: Proc. of the *SIGCHI conference on Human factors in compu-ting systems, Pittsburgh, Pennsylvania*, US, pp. 434–441.

Fenz, S., 2012. An ontology-based approach for constructing Bayesian networks. *Data & Knowledge Engineering*, v. 73, p. 73-88.

Forkan, A. R. M., Khalil, I., Tari, Z., Foufou, S., Bouras, A., 2015. A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living. *Pattern Recognition, v. 48, n. 3*, p. 628-641.

Friedman, N., Geiger, D., Goldszmidt, M., 1997. Bayesian network classifiers. *Machine learning*, v. 29, n. 2-3, p. 131-163.

Hobbs, J. R., Pan, F., 2004. An ontology of time for the semantic web.ACM Transactions on Asian Language Information Processing (TALIP), v. 3, n. 1, p. 66-85.

Howard, C; Stumptner, M., 2014. A Survey of Directed Entity-Relation--Based First-Order Probabilistic Languages. *ACM Computing Surveys (CSUR), v. 47, n. 1*, p. 4.

Laskey, K., 2008. MEBN: A language for first-order Bayesian knowledge bases. *Artificial intelligence*, v. 172, n. 2, p. 140-178.

Machado, A., Pernas, A. M., Augustin, I., Thom, L. H., Krug, L., Palazzo, J., Oliveira, M. De, 2013. Situation awareness as a Key for Proactive Actions in Ambient Assisted Living. In: *Proc. of the 15th International Conference on Enterprise Information*, p. 418–426.

Machado, A., Lichtnow, D., Pernas, A. M., Wives, L. K., de Oliveira, J. P. M. 2014. A Reactive and Proactive Approach for Ambient Intelligence. *International Conference on Enterprise Information System*, p. 501-512.

Paganelli, F., Giuli, D., 2011. An ontology-based system for context-aware and configurable services to support home-based continuous care. *In. Journal of the IEEE Transaction Information Technology Biomedical*, *v. 15, n. 2*, p. 324-333.

Rasch, K., Li, F., Sehic, S., Ayani, R., Dustdar, S., 2011. Context driven personalized service discovery in pervasive environments. World Wide Web, v. 14, n. 4, p. 295-319, springer, Netherlands.

Sixsmith, A., Meuller, S., Lull, F., Klein, M., Bierhoff, I., Delaney, S., Savage, R., 2009. SOPRANO: An Ambient Assisted Living System for Supporting Older People at Home. In: *Lecture Notes in Computer Science.* Springer Berlin: Heidelberg, vol. 5597, p. 233–236.

Strang, T., Linnhoff-Popien, C., 2004. A Context Modeling Survey. In: Workshop on Advanced Context Modeling, Reasoning and Management, UbiComp 2004 - *The Sixth International Conference on Ubiquitous Computing*, Nottingham, England.

Tazari, M. R., Furfari, F., Ramos, J. P. L., Ferro, E, 2010. The PERSONA service platform for AAL spaces.

In: *Handbook of Ambient Intelligence and Smart Environments*. Springer US. p. 1171-1199.

Yang, Y., Calmet, J., 2005. Ontobayes: An ontology-driven uncertainty model. In: Computational Intelligence for Modelling, Control and Automation, 2005 and *International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on. IEEE*, p. 457-463.

Ye J, Stevenson G, Dobson S., 2011. A top-level ontology for smart environments. *In: Pervasive and Mobile Computing. vol. 7, no. 3*, p. 359-378.