http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (8): 1130-1138, 2007
ISSN 1812-5638

A RDF and OWL-Based Temporal Context Reasoning Model for Smart Home

Hsien-Chou Liao and Chien-Chih Tu
Department of Computer Science and Information Engineering, Chaoyang University of Technology,
168 Jifong E. Rd., Wufeng Township Taichung County, 41349, Taiwan, Republic of Chaina

Abstract: In the ubiquitous computing environment, context reasoning is an important issue of context-awareness. It is used to deduce desired or higher-level context and then to provide suitable services automatically. The previous context-reasoning approaches are mainly non-temporal. The reasoning is according to the real-time contexts without time information. However, temporal contexts are very important information for context-awareness. Therefore, a model, called TempCRM (Temporal Context Reasoning Model), based on Resource Description Framework (RDF) and Web Ontology Language (OWL) is proposed in this paper. TempCRM is used for inferring the dangerous level of a smart home. In a home environment, a potential dangerous situation is caused by a series of temporal events. A temporal event is represented as a RDF-based temporal context. A smart home ontology is defined for the terms and relationships used in the temporal context. Then, a set of reasoning rules can be defined for inferring and computing the dangerous level. In the simulation study, a script with dangerous situations is designed to evaluate the dangerous level generated by TempCRM. The result illustrates that TempCRM is useful to alarm the inhabitant and thus prevent the occurrence of an incident from the temporal contexts.

Key words: Context-awareness, smart home, temporal context, first-order predicate logic, resource description framework, web ontology language

INTRODUCTION

Home is the most important place in people's daily left. Smart home is an important research area of ubiquitous computing. The integration of hardware and information technology into the home environment is mainly to achieve two important goals: entertainment and safety. For the safety goal, many kinds of detectors or sensors are used to alarm the inhabitant the occurrence of an incident, e.g., CO (Carbon Monoxide) or gas detector, infrared detector, biosignal monitor of older adults, etc., Such kind of detectors or sensors is useful for instant incident. However, most of the incidents are caused by the potential dangerous situations. For example, the inhabitant forgets to turn off the gas switch causes smoke emission and is then detected by the smoke detector. The detectors or sensors are unable to find out the existence of the potential dangerous that is caused by a series of temporal events. If such a potential dangerous situation can be aware before the occurrence of the incident, it will be effective to prevent the tragedies and increase the safety of a home.

The awareness of a specific situation is called context-awareness. It relies on the context reasoning to generate desired or higher-level context. For example, the activity in a meeting room can be deduced from the executed software and the number of peoples in the room. If the number of peoples is more than three and PowerPoint is executing, a presentation is the most likely activity in the room. In the previous research, Chen *et al.* (2004b) defined an architecture for context reasoning, called CoBrA (Context Broker Architecture). CoBrA can provide an appropriate service or information for a user according to his contexts. Wang *et al.* (2004b) proposed a semantic space infrastructure. The context reasoning is based on the semantic Web technology. A context-aware service, called SituAwarePhone, was also implemented to illustrate the feasibility of the infrastructure. It can switch the mobile phone into silent or normal mode automatically according the user's activity. Man *et al.* (2005) proposed a reasoning method for the smart meeting room. This method and another agent-based framework proposed by Chen and Finin (2004a) utilize ontology in the definition of context information and reasoning rules. However, it is still not flexible enough for the reasoning of complex situation. Therefore, Ranganathan and Campbell (2003) utilized first-order predicate logic for defining complex reasoning rules. Wang *et al.* (2004a) incorporated ontology and rule-based for deducing high-level context in a complex environment.

Corresponding Author: Hsien-Chou Liao, Department of Computer Science and Information Engineering, Chaoyang University of Technology, 168 Jifong E. Rd., Wufeng Township Taichung County, 41349, Taiwan, Republic of Chaina Tel: +886-4-23323000/4211 Fax: +886-4-23742375

The above approaches are non-temporal. That is, the context-reasoning can deduce a high-level context on the specific time and deliver suitable information or service for satisfying user's demand automatically. However, they are unable to infer from a series of temporal contents. Some temporal representation and reasoning approaches were also proposed. For example, Nogueira *et al.* (2004) built a framework of contextual logic programming language (CxLP) for a temporal database. Zhu *et al.* (2004) extended a context interchange framework (COIN) for overcoming the context heterogeneity problem that is complicated when the context is changed over time. Moldovan *et al.* (2005), proposed a model for converting the temporal event in a query of natural language into a logic representation in order to increase the performance of a question answering system. Although the temporal context can be represented in the above approaches, they still cannot be used for detecting and inferring the potential dangerous situations from a series of temporal events.

In order to infer the dangerous level of a smart home from the temporal contexts, a temporal context reasoning model (TempCRM) is defined in this paper. In TempCRM, the semantic Web is used to represent the temporal contexts. The Resource Description Framework (RDF) and Web ontology language (OWL) are two important specifications. RDF is originally designed as a metadata model and is used to model information here. OWL is used to represent terms and their relationships of a specific domain, i.e., ontology. A smart home ontology is defined in TempCRM. In addition, the first-order predicate logic to represent the temporal reasoning rules for potential dangerous situations caused by the usage of devices. The usage behaviors are different in different homes. For example, the usage interval of a gas stove is different for various homes. A probability density function is then defined to model the usage behavior of a device. The function is then used in the reasoning rules to generate a proper dangerous level matched with the usage behavior. Finally, an event script is defined for the simulation study of TempCRM. The result illustrates that TempCRM can generate expected dangerous level from temporal contexts and thus increase the safety in the home environment.

MATERIALS AND METHODS

In the previous research of context-awareness, CoBrA and Gaia are two well-known frameworks (Ranganathan *et al*., 2004; Ranganathan and Campbell, 2002; Roman and Campbell, 2000; Roman *et al*., 2002). CoBrA is based on the OWL and RDF specification and

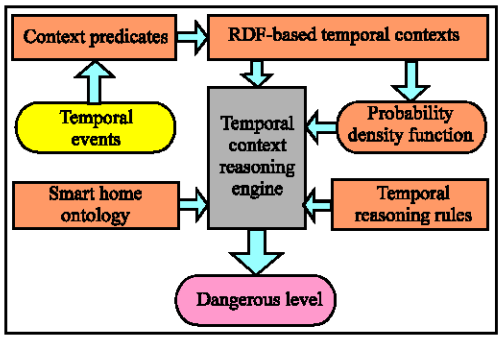

Fig. 1: The representation of TempCRM model

Gaia is based on the first-order predicate logic. The OWL and RDF is powerful for context representation and reasoning. However, the first-order predicate logic is a simple and widely used. It is also easily used in the rule-based reasoning engine. Therefore, OWL and RDF are incorporated with the first-order predicate logic for temporal context reasoning in TempCRM (Fig. 1).

TempCRM consists of five main parts: context predicates, RDF-based temporal contexts, probability density function, smart home ontology and temporal reasoning rules. Any inhabitant activities, movements and the interactions with devices are deemed as temporal events. A temporal event is represented as a context predicate. A smart home ontology is defined for the standard terms and their relationships used in the home environment. A context predicate is then represented as RDF-based temporal contexts by incorporating the smart home ontology. According to the history temporal contexts, the usage behavior of a device is modeled by a probability density function. For a potential dangerous situation, a corresponding temporal reasoning rule is defined for detecting and computing its dangerous level. When the rules are executed by the reasoning engine, a final dangerous level is generated. The reasoning engine is executed under two conditions. One is executed once a new temporal context has been generated. It is similar to a general rule-based reasoning engine. The other is executed periodically. The dangerous level is usually increased as the elapsed time. The periodic execution can refresh the dangerous level. These parts are presented in details in the following subsections.

Context predicate: In TempCRM, the first-order predicate is used to represent the temporal events (Ranganathan and Campbell, 2003). The first-order predicate is a simple and uniform format. When a temporal event is occurred, it is represented as a predicate of a

context type with four tuples, including subject, verb, object and time (SVOT). Its format is shown below.

ContextType(<subject>, <verb>, <object>, <time>)

The format is the expansion of the SVO (subject-verb-object) format in the previous research (Ranganathan and Campbell, 2003) by appending the time tuple. ContextType is the type of a temporal event, such as location, temperature, device and so on. <subject> is the thing or person giving the context. <verb> is the action or relation to the object. <object> is the value of the context. <time> is the occurrence time of the event. Some examples are listed in the following:

Location(john, entering, kitchen, 12:38:41).
Location(tv_set#1, in, living_room, 18:03:15).
Device(gas_switch#2, is, on, 08:13:50).
Door(entry_door#1, is, open, 02:25:17).
Temperature(room#415, =, 98F, 13:45:20).

It is easily to realize the context in the above examples. The values of subject, verb and object are determined by the context type. For example, if the context type is Location, the subject could be either a person or device. Verb could be in, entering, leaving and so on. Object must be an explicit place name.

The time tuple represents the occurrence of the event. However, one important characteristic of the context predicates is the context persistence. It means a context predicate is persistence until the occurrence of a new predicate with the same type, subject and verb. For example, two context predicates are generated on different time as shown below.

Location(john, entering, kitchen, 12:38:41).
Location(john, entering, living_room, 14:05:07).

The context type, subject and verb of two predicates are the same. The first predicate represents the location of John is in the kitchen from the time 12:38:41. John's location is persistent in the kitchen until the generation of the second context predicate. Therefore, the query of John's location from 12:38:41 to 14:05:06 will return kitchen.

In addition, Boolean operators can be used to represent more complex context predicate, including conjunction ($\wedge$), disjunction ($\vee$) and negation ($\neg$). A variable can also be defined to query desired context predicates. It is quantified by two kinds of quantifiers: existential ($\exists$) and universal ($\forall$). The existential quantifier means there exists a and the universal quantifier means for all or for every. They are popular used in the first-order predicate.

Smart home ontology: In TempCRM, an ontology is defined in OWL specification for the terms and their relationships in the home environment, called smart home ontology. OWL is a tree structure for the classification of terms. The relationship between two terms can be easily inferred via the tree structure.

The ontology consists of an upper-level context ontology and extended context ontologies (Wang *et al*., 2004a). The upper-level ontology provides a set of basic concepts, i.e., entities. A connection is established between two entities via an attribute (owl:DatatypeProperty) or relationship (owl:ObjectProperty). Besides, the relationship between an entity and its sub-entity of extended context is established via an attribute (owl:subClassOf). Therefore, the relationship between an entity of the upper-level context and an entity of an extended context can be established in such way.

In TempCRM, four basic entities are defined as the upper-level context ontology for the smart home. It contains computing entity, location, person and activity as shown on the left side of Fig. 2. There are two extended contexts on the right side. The entity Room of the upper-level context is extended to a set of sub-entities of smart home domain, including Living Room, Bath Room, Bed Room and so on.

The smart home ontology represents all the entities, attributes and relationships for representing RDF-based temporal context predicates. The ontology also provides legal representations for the predicates. For example, a Person entity can only have name, sex, age, height and weight attributes.

RDF-based temporal context: RDF is a specification formed by a set of nodes and labeled links between nodes. A node represents a resource, including the entity of smart home ontology, or the tuple value of a context predicate. A link represents attributes of a resource.

Basically, RDF can be used to represent the non-temporal resources in the home environment. In TempCRM, RDF is used to represent the temporal contexts called RDF-based temporal context. A context predicate defined in the previous subsection represents the context of a temporal event. It can be transformed into RDF format based on the smart home ontology. For example, the RDF representation of the following context predicate (Fig. 3).

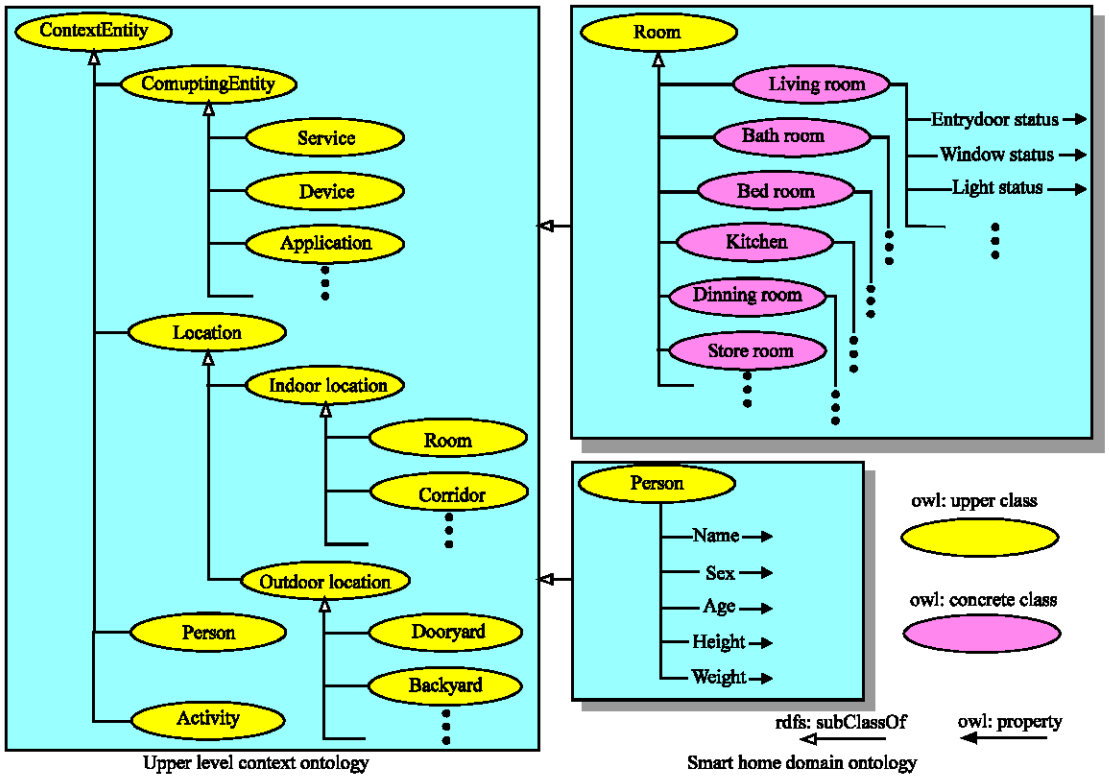

Fig. 2: The definition of upper level context ontology and domain ontology

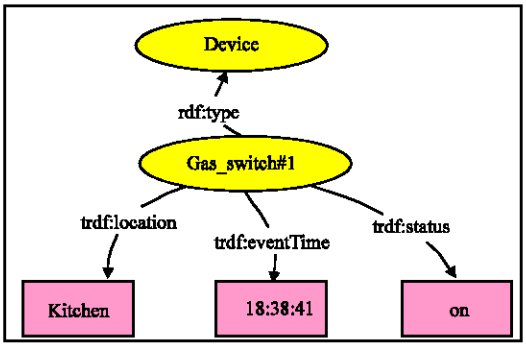

Fig. 3: RDF representation of a context predicate

The above representation is based on the SVOT format. It is only represents the primary information of a temporal event. It lacks the actor of the event. The lack of such information may influence the reasoning process in some cases.

The actor of the event is able to acquire via sensor or visual monitoring technique. For example, Philipose *et al.* (2004) designed a wearable glove with a RFID reader. It enables the system know the object touched by the user from the attached RFID tag. Peursum *et al.* (2004) proposed a method for smart-home monitoring system. The method can label the location of objects, such as chairs or floors from the captured video. It is also easily to know the actor of an object.

If the additional information of an event is available when the event occurs, the RDF specification can represent the temporal event in a sophisticated way. For the above example, if the actor is John, he is in the kitchen according to a previous event. The context predicate is shown below. Then, the RDF-based temporal context of the turning on of the gas switch is represented in the Fig. 4.

The RDF-based temporal context can represent not only the basic non-temporal resources or context predicates, but also the relationship among context predicate for a temporal event. The relationship is very important to allow the reasoning engine to detect the dangerous situation in the reasoning process.

Probability density function of device status: Different home may have different usage behaviors of devices. For example, the usage times of a gas stove are diverse in different home. The usage behavior should be incorporated into the reasoning process in order to generate feasible the reasonable results. A probability density function is used to model the usage behaviors of devices. The functions are used in the temporal reasoning process for generating a reasonable dangerous level.

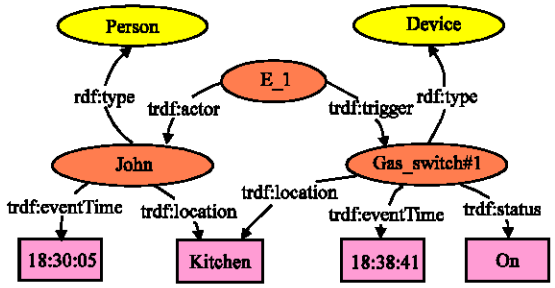

Fig. 4: An example of RDF-based temporal context

In TempCRM, we only focus on the usage behaviors of devices since a dangerous situation is usually caused by devices, such as gas stove, water heater and so on. The status of a device is considered separately since the corresponding behavior is different. For one status of a device D_i, a normal distribution is used to model its behavior currently. Therefore, its average interval (μ_i) and standard deviation (σ_i) is computed from the history temporal contexts. When D_i enters the status, the elapsed time for the status is denoted as TE. The probability corresponding to TE can be computed using the probability density function, denoted as f_{Di} (TE), as shown in Eq. 1.

$$f_{D_i}(TE) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left(-\frac{(TE-\mu_i)^2}{2\sigma_i^2}\right) \quad (1)$$

In general, most of the dangerous situation occurs when the TE lager than the average interval (μ_i) marked by the grey area in the figure. Besides, the dangerous level is expected between zero and one. The level zero means no dangerous. Oppositely, the level one means extreme dangerous. The expected level is the reverse of the normal distribution. Therefore, the probability density function is transformed into the function of dangerous level with respect to TE, denoted as DL_{Di} (TE), as shown in Eq. 2.

$$DL_{D_i}(TE) = \begin{cases} 0 & , \text{if } TE \le \mu_i \\ 1 - f_{D_i}(TE), & \text{otherwise} \end{cases} \quad (2)$$

When TE is larger than the status interval μ_i, the dangerous level is increased from zero. If the status deviation σ_i of D_i is small, the dangerous level is increased rapidly. DL_{Di} (TE) is used in the reasoning process when necessary.

Temporal reasoning rules: Temporal reasoning rules are used to detect and compute the level of potential dangerous situations. Basically, the reasoning engine executes the rules periodically, e.g., every 30 sec or one minute. However, the engine execution also be activated when a temporal event is occurred. The periodical execution keeps the update of dangerous level. The event-driven execution can update the dangerous level immediately.

```
P1.  ComputeDangerousLevel(CT, Level) :-
P2.       ∃Device D (DeviceType(D, gas_switch) ∧
P3.            Device(D, is, on, TS) ∧
P4.            Location(D, in, Place, _) ∧
P5.       ¬∀Person P (Location (P, in, Place, _)) ∧
P6.            GetElapsedTime (TS, CT, TE) ∧
P7.            ComputeLevel(D, on, TE, Level))
```

Fig. 5: The rule for inferring and computing the dangerous level of forgetting to turn off the gas switch

A potential dangerous situation is consequence of a series of temporal events. An example is used to illustrate how to define a reasoning rule for a dangerous situation. The example is the potential dangerous of forgetting to turn off the gas switch. Suppose an inhabitant turns on the gas switch of a gas stove. But he does not stay at the location of the gas stove. The dangerous level is increased when the elapsed time is larger than the average usage interval of the gas switch. This reasoning rule represented in first-order predicate (Fig. 5).

The predicate P1, ComputeDangerousLevel, represents the reasoning of dangerous level (Level) at the Current Time (CT). The predicate P2 represents the checking of all Devices (D) where the type is gas_switch. Then, the device D is checked whether its status is on at a specific time (TS) in the predicate P3. The location of D is also retrieved and stored into the variable Place in the predicate P4. In the predicate P5, reasoning engine tries to find out a Person (P) where stays at the same Place of the device D. There is a negation operator means this predicate is true when no person P stays at the same Place of D. If P5 is true, the elapsed time TE from TS to CT is computed in the predicate P6. In the last predicate, the probability density function of the on status of D is used to compute the level according the TE.

The above rule is operated on RDF-based temporal contexts. A graph similar to that in the previous research of Forstadius *et al.* (2005) is used to illustrate the reasoning process as shown in Fig. 6. Only the query predicates from P2 to P5 are marked in the graph. They are responsible to unify the temporal contexts. The rule consists of two conditions. There are three predicates in the first condition. For the predicate P2, it tries to unify a device resource where its type is Device and its deviceType is gas_switch. The name of unified resource

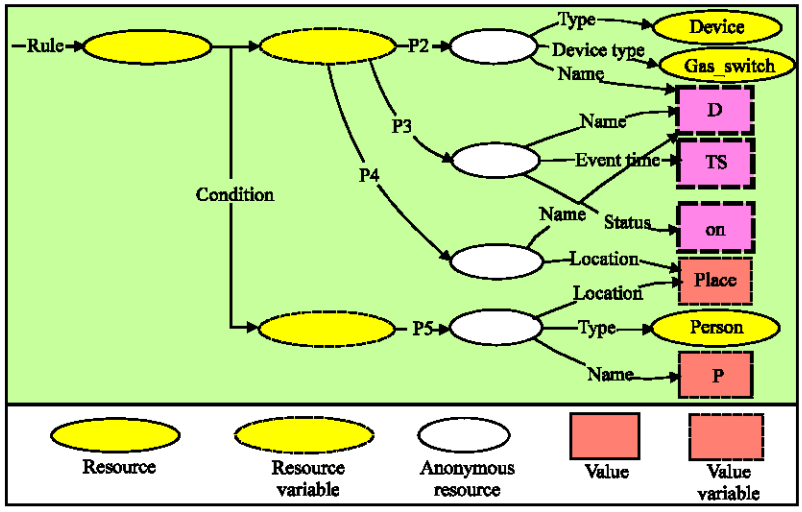

Fig. 6: The graph representation of a rule

is stored in a variable D. The variable D is then used in the unification of the predicate P3 and P4. After the first condition is unified successfully, the variable Place is used in the unification of the second condition. If the four anonymous resources can be unified successfully, the variables D, TS and on is used to compute the dangerous level in the predicate P6 and P7.

According to the above reasoning process, when any inhabitant enters the location of the device, the predicate P5 is fail and then caused the fail of this rule. The dangerous level equals the default value zero which means there is no dangerous. When no inhabitant at the location, the dangerous level is computed based on the elapsed time.

The smart home ontology is used in the reasoning process. For example, it is used in the predicate P2 to allow this rule applying to any kinds of gas switch. It is unnecessary to define rules for individual devices with the same type.

In the home environment, it is possible to encounter more than one potential dangerous situation at a specific time. A final dangerous level at the time CT is denoted as FinalDL(CT). When there are more than one dangerous levels at the time CT, FinalDL(CT) equals the maximum dangerous level Eq. 3. That is, the final dangerous level is the level of the most emerging situation.

$$\text{FinalDL}(\text{CT}) = \text{Maximum Level among all ComputeDangerousLevel(CT, Level)} \quad (3)$$

In advance, the numeric value of FinalDL(CT) is not clear enough to inhabitants, especially older adults. Therefore, the numeric value is mapped into three modes, including normal, warning and emergent modes. In TempCRM, two values, T_w and T_e, are defined to determine the thresholds entering warning or emergent mode, respectively. They can be adjusted by the inhabitant. A conservative inhabitant can assign lower thresholds to receive more warnings of potential dangerous situations.

SIMULATION STUDIES

A simulation was designed to valid the operation of TempCRM. The simulation is based on a series of temporal events. These events were designed to cause three potential dangerous situations. All the reasoning rules, smart home ontology and related RDF-based temporal contexts were implemented in Prolog language. The dangerous levels were computed and recorded periodically. The set of events are shown in Table 1. They are represented in context predicates since it is easier to understand than plain text descriptions. The set of events represent a script that John drives back to home, enters the kitchen, turns on the gas switch and leaves the kitchen. Peter enters the bathroom, turns on the water heater for taking a shower and so on.

The simulation parameters are shown in Table 2. The parameters include the averages and derivations of three device status, two thresholds for warning and emergent modes and the reasoning interval. The final dangerous level is computed for every minute by submitting a Prolog query command manually.

The recorded dangerous levels within one hour after John coming home are depicted as curves (Fig. 7). There are three curves in the figure. Three curves represent the dangerous levels of $_{D_0}$(on), D_2(on) and D_3(open),

Table 1: A series of events for the simulation

Event ID	Description (Context predicate)
E_1	Door(garage_door#1, is, open, 18:35:13)
E_2	Location(john, entering, garage, 18:35:25)
E_3	Location(john, entering, kitchen, 18:38:41)
E_4	Device(gas_switch#1, is, on, 18:39:25)
E_5	Location(peter, entering, bathroom, 18:39:40)
E_6	Device(water_heater#2, is, on, 18:40:02)
E_7	Location(john, entering, living_room, 18:40:05)
E_8	Device(tv_set#1, is, on, 18:42:36)
E_9	Location(john, entering, garage, 18:50:22)
E_10	Door(garage_door#1, is, close, 18:50:35)
E_11	Location(john, entering, living_room, 18:51:28)
E_12	Location(john, entering, kitchen, 19:13:42)
E_13	Location(john, entering, bedroom, 19:17:57)
E_14	Device(water_heater#2, is, off, 19:25:02)
E_15	Location(peter, entering, living_room, 19:26:40)

Table 2: The settings of simulation parameters

	Settings		
Parameters	D_1(on) gas_switch#1	D_2(on) water_heater#2	D_3(open) garage_door#1
Average (μ_i)	1,121 sec	1,200 sec	460 sec
Deviation (σ_i)	808 sec	935 sec	168 sec
T_w		0.6	
T_e		0.8	
Reasoning Interval		60 sec	

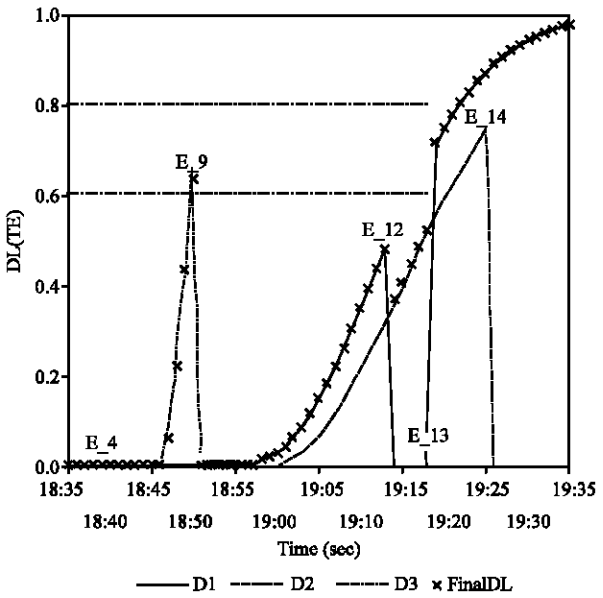

Fig. 7: The curves of the simulation result

respectively. The fourth curve represents the FinalDL that equals the maximum value of three curves. In the event E_4, John enters the kitchen and turns on the gas switch. Then, he enters the living room in the event E_7. The elapsed time of the device status, $_{Do}$(on), is increased. The dangerous level is increased when the time is larger than the average interval as shown in the curve of $_{Do}$.

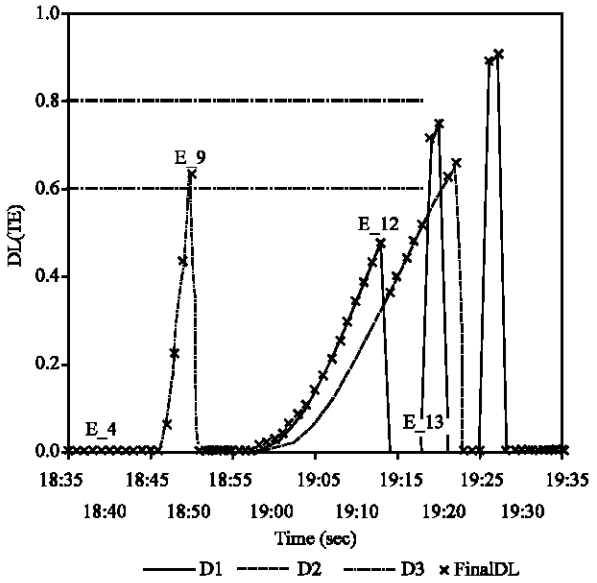

Fig. 8: The simulation result of solving masking effect by choosing remind me every five minutes

In the event E_12, John goes back to the kitchen. It causes the dangerous level of $_{Do}$(on) back to zero since there is a person at the same location of $_{Do}$. However, John leaves the kitchen in the event E_13 causes the dangerous level returning to the expected value computed from the elapsed time as shown in the curve. It also causes the home entering the warning mode. A flash light or voice message can remain John this dangerous situation. On the other hand, the open of garage door causes the dangerous level of D_2(on) is increased after the elapsed time larger than the average. It level is back to zero when John enters the garage in the event E_9.

The value of FinalDL is the maximum among all the dangerous levels. It enables the inhabitant concentrating on the most emerging dangerous situation. However, it may cause a problem called masking effect. In the above example, the inhabitant really needs to cook for several hours. As a result, the dangerous level of $_{Do}$(on) is increased continuously. It masks the dangerous level of D_2(on) except the period between the event E_12 and E_13. In fact, a simple user interface is useful to solve the problem of the masking effect. When the FinalDL causes the home entering the warning or emergent mode, three choices can be provided by the user interface: keep warning, cancel warning, or remind me later. The last two choices enable the inhabitant being aware of a minor dangerous situation. If the inhabitant chooses remind me for every five minutes for solving the mask effect in the above simulation (Fig. 8). It is obvious that the dangerous

situation of D_2 can be revealed after the dangerous level of $_{D0}$ is reduced to zero.

The above simulation results illustrate TempCRM can detect and compute the level of the potential dangerous situations to prevent the occurrence of incidents. The level can be increased as the time elapsed. This is the key difference of TempCRM compared with current temporal reasoning models. Besides, a reasoning rule mixed with RDF-based contexts usually suffers from the modification of a house, e.g., house renovation. However, the rule is also mixed with a smart home ontology in TempCRM. It enables a rule applying to the same type of devices or rooms. That is, the house modification does not lead to an overhaul of the entire rule set. TempCRM is useful to increase the safety of a smart home.

CONCLUSIONS

Temporal context-awareness is an important issue. Some temporal-related information or high-level contexts can only be inferred from temporal context reasoning. The detection of potential dangerous situation in a home environment is an important application of temporal context reasoning. The proposed TempCRM can be used to detect the potential dangerous situations. A warning or emergent alarm can be activated when the dangerous level is larger than the customizable threshold. The usage behaviors of devices are modeled by the probability density functions that are established from the history temporal contexts. It causes the inferred dangerous level fits inhabitant's behavior. TempCRM can be an effective approach to increase the safety of a smart home.

The limitation of TempCRM is the lack of temporal contexts. Some temporal contexts may be unavailable for privacy or cost considerations. Reasoning rules cannot be deduced without desired contexts. It causes that the dangerous level cannot be computed successfully. In addition, the definition of the reasoning rules is another limitation of this study. The rules for different homes may be diverse and it is not easy to define rules for inhabitants. Therefore, the definition of reasoning rules may limit the performance of TempCRM.

In the future works, TempCRM will be implemented by incorporating the Jena and Jess (Java expert system shell) reasoning engine. It is expected to be a temporal context reasoning platform. Besides, the detection of a potential dangerous situation depends on the temporal reasoning rules. The definition of the rules becomes a critical task. It also determines the effectiveness of TempCRM. The dangerous situations are diverse for different home environments. It is unable to define a complete set of rules for all the homes. Therefore, when an inhabitant encounters a potential dangerous situation that is not detected by existing rules, the design of a semi-automatic mechanism for establishing a corresponding reasoning rule of the dangerous situation is another important work in the future.

REFERENCES

Chen, H. and T. Finin, 2004a. An ontology for context aware pervasive computing environments. Knowledge Eng. Rev., 18: 197-207.

Chen, H., T. Finin, A. Joshi and L. Kagal, 2004b. Intelligent agents meet the semantic web in smart spaces. IEEE Internet Comput., 8: 69-70.

Forstadius, J., O. Lassila and T. Seppanen, 2005. RDF-based model for context-aware reasoning in rich service environment. The 3rd IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom), 8-12 pp: 15-19.

Man, J.F., Q. Chen, X.H. Deng and Y.A. Qiu, 2005. The design and implementation of shared ontologies for smart space application. International Conference on Machine Learning and Cybernetics, 1: 125-131.

Moldovan, D., C. Clark and S. Harabagiu, 2005. Temporal context representation and reasoning. The Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, July 30-Aug. 5, pp: 1099-1104.

Nogueira, V.B., S. Abreu and G. David, 2004. Towards temporal reasoning in constraint contextual logic programming. The 3rd International Workshop on Multiparadigm Constraint Programming Languages (MultiCPL'04), Saint-Malo, France, pp: 6-10, 119-131.

Peursum, P., S. Venkatesh, G.A.W. West and H.H. Bui, 2004. Using interaction signatures to find and label chairs and floors. IEEE Pervasive Comput., 3: 58-65.

Philipose, M., K.P. Fishkin, M. Perkowitz, D.J. Patterson, D. Fox, H. Kautz and D. Hahnel, 2004. Inferring activities from interactions with objects. IEEE Pervasive Comput., 4: 50-57.

Ranganathan, A. and R.H. Campbell, 2002. A middleware for context-aware agents in ubiquitous computing environments. USENIX International Middleware Conference.

Ranganathan, A. and R. Campbell, 2003. An infrastructure for context-awareness based on first order logic. Personal and Ubiquitous Comput., 7: 353-364.

Ranganathan, A., J. Al-Muhtadi and R.H. Campbell, 2004. Reasoning about uncertain contexts in pervasive computing environments. IEEE Pervasive Comput., 4: 62-70.

Roman, M. and R.H. Campbell, 2000. GAIA: Enabling active spaces. The Ninth Workshop on ACM SIGOPS European Workshop, pp: 229-234.

Roman, M., C.K. Hess, R. Cerqueira, A. Ranganathan, R.H. Campbell and K. Nahrstedt, 2002. Gaia: A middleware infrastructure to enable active spaces. IEEE Pervasive Comput., 1: 74-83.

Wang, X.H., T. Gu, D.Q. Zhang and K.P. Hung, 2004a. Ontology based context modeling and reasoning using OWL. IEEE Annual Conference on Pervasive Computing and Communications Workshops, pp: 18-22.

Wang, X.H., J.S. Dong, C.Y. Chin and S.R. Hettiarachchi, 2004b. Semantic Space: An Infrastructure for Smart Spaces. IEEE Pervasive Computing, 3: 32-39.

Zhu, H., S.E. Madnick and M.D. Siegel, 2004. Reasoning about temporal context using ontology and abductive constraint logic programming. Lecture Notes in Computer Science, 3208: 90-101.