International Journal of Smart Home
Vol. 10, No. 1, (2016), pp. 101-114
http://dx.doi.org/10.14257/ijsh.2016.10.1.11

A Smart Home Context-aware Model Based on UML and Colored Petri Net

Xiaopeng Niu* and Zhiliang Wang

School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
**supernxpeng@126.com*

Abstract

Smart Home is one of the main application domains of Pervasive Computing, and it can provide Context-aware Services to homeowners. As context information is obtained from different sources in the Smart Home environment, the context information may be heterogeneous, inaccurate and dynamic. How to effectively model these different types of context information is a main research issue. To address this issue, we propose a hybrid Context-aware Modeling approach based on UML and Colored Petri Net in this paper. We also model leave home scenario using this approach, and use Coverability Graph to verify the leave home scenario model.

Keywords*: Pervasive Computing, Context-aware, Smart Home, Colored Petri Net (CPN), Unified Modeling Language (UML), Context model*

1. Introduction

Smart Home is one of the main research fields of pervasive computing. Smart home can provide context-aware services to homeowners by using sensor, modeling, reasoning, and other technologies. Context-aware is a key research issue of Smart Home. The definition of context varied from different researchers [1, 2, 4, 6, 21, 22], Dey [2] defined context as any information that can be used to characterize the situation of an entity.

How to effectively use different types of context information [3] to provide context-aware service is a core issue, which requires context modeling technology. A good context model can not only well characterize the features of context information, but also support context reasoning to obtain high-level context [3, 5].

Ahn [6] defined context as a set of interrelated events, which can be classified into discrete events and continuous events. There are a lot of context information types in the Smart home environment, such as people, resource, system, and some discrete events. The Petri Net (PN) is suitable for modeling context in Smart Home environment as it has the ability of modeling dynamic systems, discrete-event systems, and concurrent systems [7]. UML is an Object-Oriented Modeling Language [15], which can facilitate the requirement analysis when design systems.

In this paper, we propose a hybrid context-aware modeling approach based on UML and CPN. This approach uses UML Activity Diagram to describe the context-aware service logic, and then maps the UML Activity Diagram into CPN. The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents the proposed modeling approach, Section 4 models a Smart Home scenario using the proposed apporach, and Section 5 draws the conclusion.

ISSN: 1975-4094 IJSH

2. Related Work

Earlier researchers have proposed context modeling approach using PN or CPN. Sun *et al.*, [8] proposed an approach to model the context of the Smart Car environment using hierarchical context modeling. PN was used to model context situation. The places of PN were represented context situations, the transitions were represented the relationships between the context situations. Nabih *et al.*, [9] used PN to model and analyze Discrete-Event Control in Smart Home, such as Fire System and Climate Control System. Reignier *et al.*, [10] used a synchronized Petri net to model a smart conference scenario. Situations and the concepts of them (activity, roles or relations) were considered as context meta-models. The places were represented situations. An event associated with a transition was represented the state change of the concepts.

Hu *et al.*, [11] proposed a Context-aware Service Modeling based on ECA (Event-Condition-Action) rules and Timed CPN. The place in Timed CPN was represented event, the value of the token was represented the type of event. Transitions were represented actions, and arc expressions or guard functions were used to describe conditions. Silva *et al.*, [12] proposed a smart library modeling based on CPN. The features of entities (user, object) were represented by the colored token, the states of entities were represented by places, and a token in a place was represented an entity. The actions of user and system were represented by transitions. Haiouni *et al.*, [13] used CPN to model context and different services which were applied to Smart Home scenarios. Context information was represented by token. A context was a set of context information. The places were classified into two types: resource places and action states places. Kwon [14] proposed Amended CPN to model Smart Campus Context-aware System. The system was decomposed into different patterns, which were represented different types of Context-aware Services. The context net was represented the change of context, which was independent from patterns. Han *et al.*, [7] proposed a context-aware model methodology to an automobile collision avoidance service used extending interval timed CPN (ITCPN). The place was represented context, and the color of the token was represented context information. The transition was represented the rule for deriving inferred context or operation, and an arc was represented the relationship among the rule and context.

3. Proposed Modeling Approach

Context-aware System is a dynamic, concurrent system [9, 14] in which context information is changing dynamically. When design Context-aware Systems, PN is an effective approach as it can verify formally, simulate dynamically [5, 14]. CPN can model complex dynamic system because CPN not only inherits the properties of the basic PN, but also has properties such as color set, and guard functions. UML can facilitate the requirement analysis in an intuitively way, and clearly describe the logic of the system [25]. UML Activity Diagram as a Dynamic Modeling Diagram, can model concurrent system. The researchers as mentioned in Section 2 mainly focused on PN or CPN modeling approach. Bettini et al. [3] suggested that different modeling approaches need to be integrated with each other in order to satisfy the requirements for context modeling.

In this paper, we propose a hybrid modeling approach based on UML and CPN. This approach integrates the merits of UML and CPN, avoids “deadlock”, and allows formal verification.

3.1. Definition of CPN

A non-hierarchical Colored Petri Net is a nine-tuple [16].

[Definition 1]: CPN= (P, T, A, ∑, V, C, G, E, I), where:

1. P is a finite set of places.
2. T is a finite set of transitions T such that $P \cap T = \phi$.

3. $A \subseteq P \times T \cup T \times P$ is a set of directed arcs.
4. Σ is a finite set of non-empty color sets.
5. V is a finite set of typed variables such that $Type\ [v] \in \Sigma$ for all variables $v \in V$.
6. $C : P \rightarrow \Sigma$ is a color set function that assigns a color set to each place.
7. $G : T \rightarrow EXPR_V$ is a guard function that assigns a guard to each transition t such that $Type\ [G(t)] = Bool$.
8. $E : A \rightarrow EXPR_V$ is an arc expression function that assigns an arc expression to each arc a such that $Type\ [E(a)] = C(p)_{MS}$, where p is the place connected to the arc a.
9. $I : P \rightarrow EXPR_\phi$ is an initialization function that assigns an initialization expression to each place p such that $Type\ [I(p)] = C(p)_{MS}$

3.2. Mapping Rules from UML Activity Diagram to PN/CPN

Many researchers have been researched how to map UML 2.0 Activity Diagram into PN/CPN. Figure 1 shows the mapping rules from UML Activity Diagram to PN/CPN [17-19].

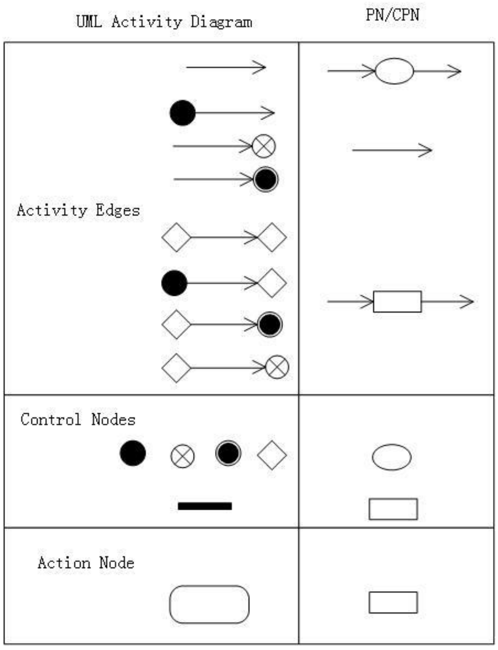

Figure 1. Mapping Rules from UML Activity Diagram to CPN

3.3. The Features of the Proposed Modeling Approach

1) UML can well describe the logic of the system, and avoid the “deadlock” of PN model.

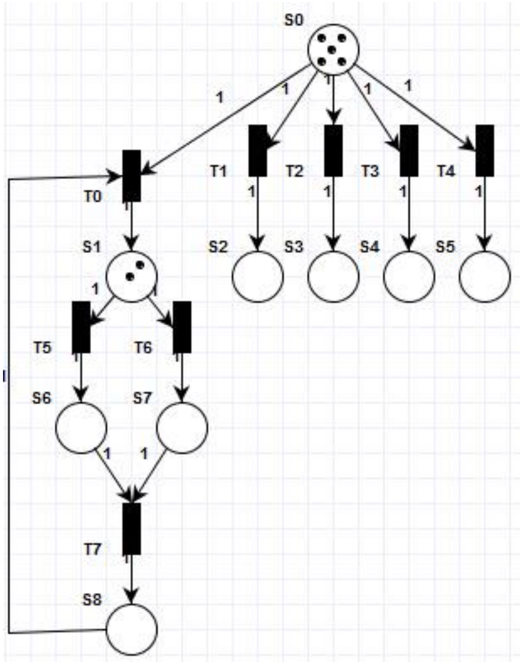

Figure 2. The Sub-model of the Smart Car Environment

Figure 2 shows a sub-model of Smart Car environment which was proposed in [8], this model has some problems such as "confusion" and "conflict". These problems may cause "deadlock" of this model because the logic of the system in [8] was not understood clearly. To resolve this issue, we can use UML to model the logic of the system, and then map the UML model into PN. Figure 3 shows the UML Activity Diagram (the loop component of Figure 2), and Figure 4 shows the revised PN model.

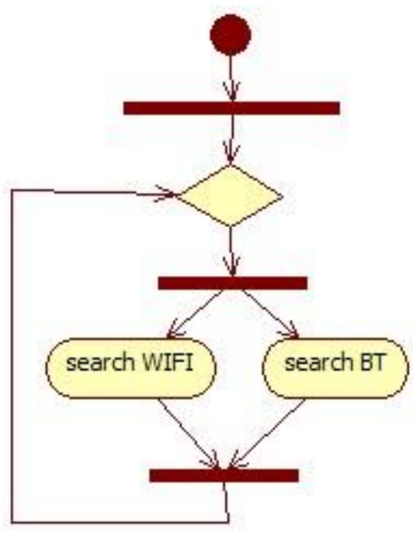

Figure 3. The UML Activity Diagram

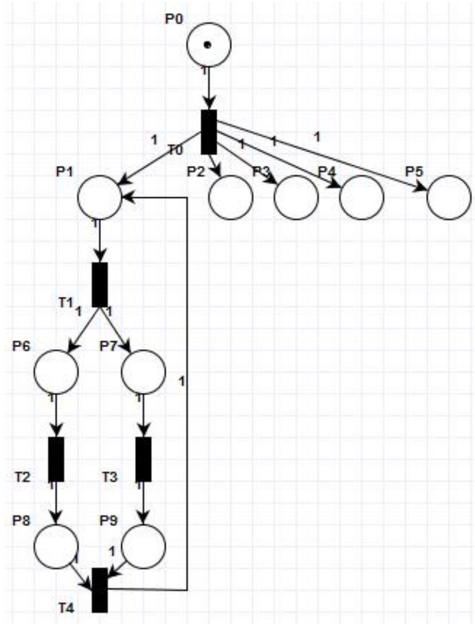

Figure 4. The Revised PN Model

2) PN can verify formally, and find error logics of the UML model through dynamic simulation. UML can better describe the logic of the system, but cannot simulate dynamically.

At the early stage of this research work, we used UML Activity Diagram to model the Fire-alarm scenario, as shown in Figure 5. After we mapped it into CPN model as shown in Figure 6, we simulated the model, and then we found the wrong logic. The wrong logic is that when one of the three types of context information doesn't satisfy the reasoning rule, the transition will be loop until the context information satisfy the rule, and the other two transitions will be waiting. The right logic is that the three transitions are concurrent, and if the reasoning rule is not satisfied, all of the three transitions ought to be repeated.

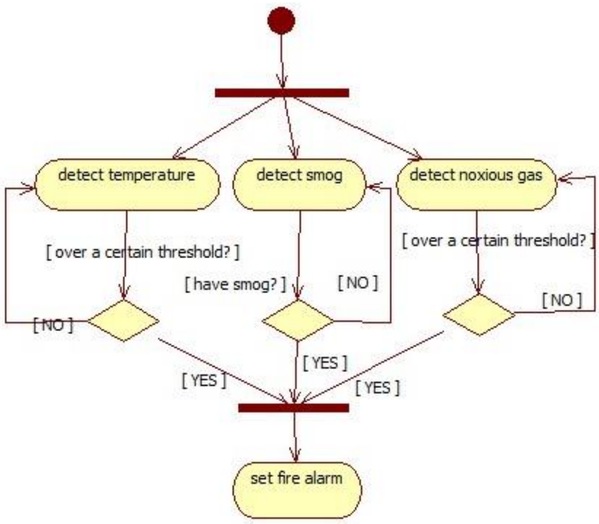

Figure 5. The Fire-alarm Scenario Model

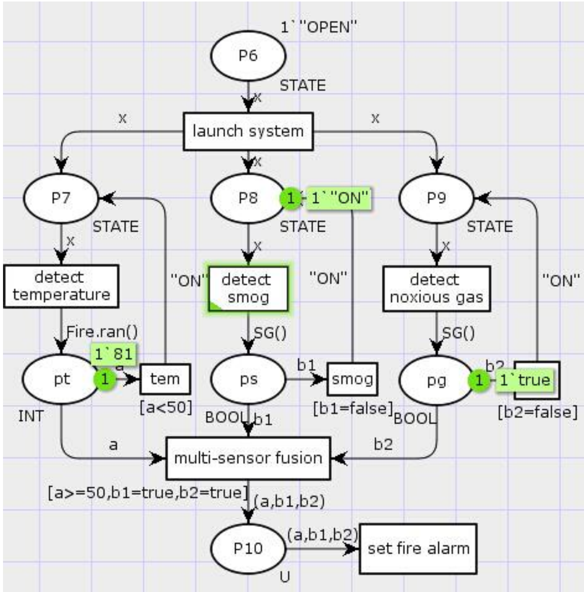

Figure 6.The CPN Model

Combine UML and CPN can merge the merits of these two approaches, and avoid the limitations of each one. So, it is an effective approach to model context-aware system.

4. Case Study

Smart Home Scenarios include leave home scenario, go home scenario, sleep scenario, *etc*. Each Scenario has several concurrent operations. In this paper, leave home scenario Service is modeled using the proposed approach.

4.1. Service Logic Model Based On UML Activity Diagram

Leave home scenario has two concurrent operations: open fire detection system and open security system. Each concurrent operation also has several concurrent operations (detect temperature, detect smog, *etc*.).

UML Activity Diagram is suitable for represents these concurrent operations. Figure 7 shows leave home scenario model based on UML Activity Diagram using StarUML. This model is divided into three systems: homeowner, fire detection system and security system.

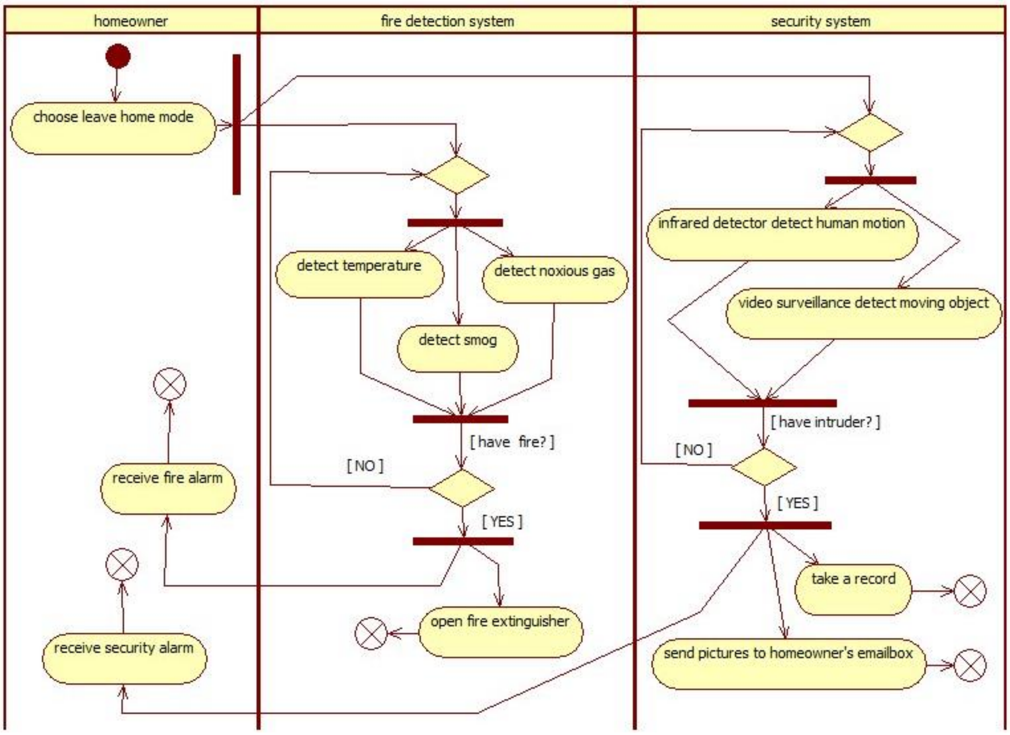

Figure 7. Leave Home Scenario Model based on UML Activity Diagram

4.2. Mapping UML Activity Diagram into CPN

By using mapping rules in Figure 1 and UML model in Figure 7, a Context-aware Service is modeled based on CPN which is shown in Figure 8.

In this model, there are two Context-aware Services (Fire-detection Service, Security Service). Each Service has several operations, and some of the operations are associated with inference rules which can generate high-level context information. The types of Context in this model are categorized in Table 1.

Table 1. Types of Context

Context Type	Context Information
Person	The state of person
Environment	Temperature, smog, gas, etc.
Resource	A tuple $\langle State, Alarm \rangle$, the first element represents the state of resourc (ON, OFF), the second element represents whether the alarm of the syster is launched successfully.
System	A tuple $\langle State, Alarm \rangle$, the first element represents the state of the syst and the concurrent sub-systems, the second element represents whether th alarm of the system is launched successfully.

The place in the leave home scenario model represents context, sensors (temperature sensor, gas sensor, *etc.*,) are considered as resource context which can generate Primary Context information [1]. Resource constraint is one of the context modeling requirements [5]. In this model, operations are not compete resources with each other. So, we do not consider constraint on resources. The places of Figure 8 are listed in Table 2.

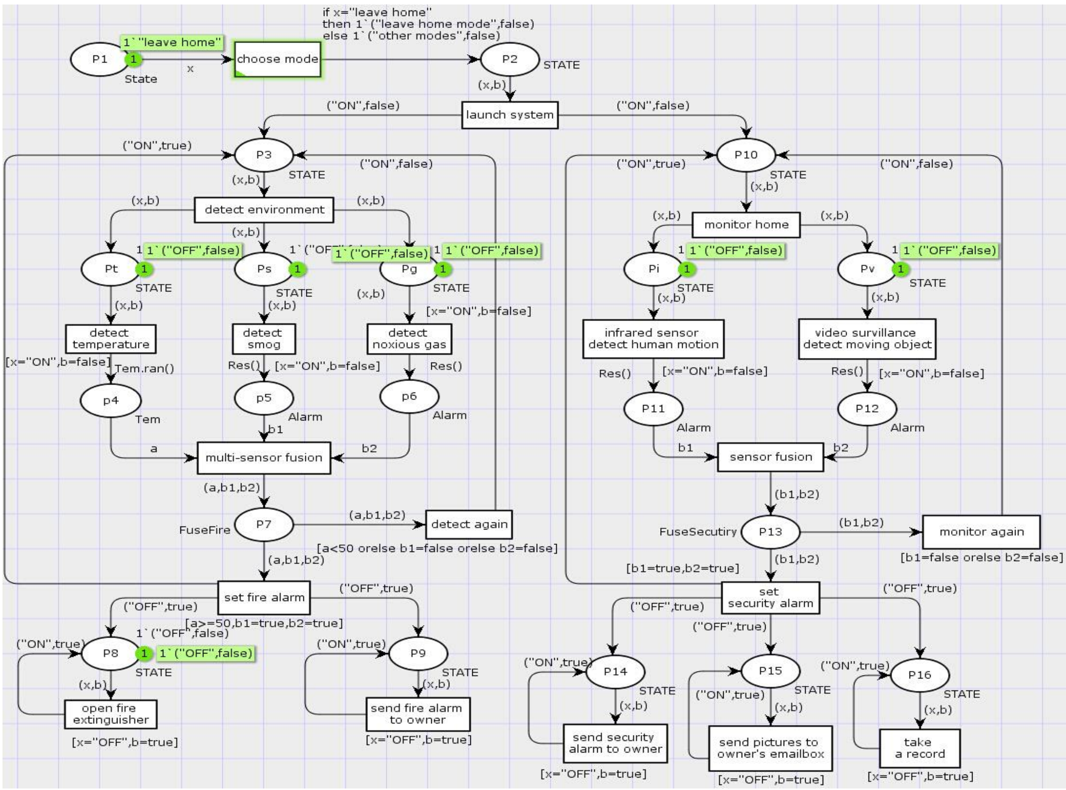

Figure 8. Leave Home Scenario Model based on CPN

Figure 9 shows the Color Sets of this model. The Color Set of each place represents context information. The state of the homeowner is represented by color set *State*. The context information of the system and resources are represented by color set *STATE*.

```
▼colset State=string;
▼colset Alarm=bool;
▼colset Tem=int with 10..80;
▼colset FuseFire=product Tem*Alarm*Alarm;
▼colset FuseSecutiry=product Alarm*Alarm;
▼colset STATE=product State*Alarm;
▼var x:State;
▼var b,b1,b2:Alarm;
▼var a:Tem;
▼colset S= int with 0..1;
▼fun Res()=
  if S.ran()=0
  then false
  else true;
```

Figure 9. Color Sets of Leave Home Scenario Model

Table 2.The Places of Figure 8

Places	Descriptions
P1	homeowner
P2	Smart Home Service System
P3	Fire-alarm sub-system
P10	Security-alarm sub-system
Pt	Temperature sensor
Ps	Smog sensor
Pg	Gas sensor

P4	Temperature
P5	Smog
P6	Gas
P7	Fire detection System fused information
P8	Fire extinguisher
P9	Send fire alarm to owner sub-system
Pi	Infrared sensor
Pv	IP Camera
P11	The infrared sensor information
P12	The video information
P13	Security System fused information
P14	Send security alarm to owner sub-system
P15	Send pictures to owner's email sub-system
P16	Take records sub-system

The transitions represent operations, and the guard functions represent the inference rules. The arcs represent the relationship between context and operations, the arc from place to transition sends the required context information to the operation, and the arc from transition to place sends the generated context information to the context. In order to simulate these concurrent operations, random functions $Tem.ran()$, $Res()$ are used in this model. We restrict the temperature from 10℃ to 80℃, and we suppose the threshold of the temperature is 50℃. The function $Tem.ran()$ is used to simulate the detected temperature, and the function $Res()$ is used to simulate the alarm signal of the smog and the noxious gas sensor. If the value of the smog or noxious gas is over a certain threshold, the value of $Res()$ is true. The inference rule is $a >= 50, b1 = true, b2 = true$. If the rule is not satisfied, the environment will be continually detected.

4.3. Verification

As shown in Figure 8, the leave home scenario model has two concurrent services. These two concurrent services are independent from each other and they have similar functions. So, we can simply verify the sub-model in order to get the properties of the entire model. In this paper, we use CG (Coverability Graph) [23, 24] to verify the Fire-alarm sub-model, and use State Space Report [16] to verify the entire leave home scenario model.

[Definition 2]: $CG = \langle V, E \rangle$, where:

$V = \{M_0, M_1, ..., M_n\}$ is a set of reachable markings, each marking is a one-dimensional vector which represents the number of tokens that each place has after the enabled transition fired. If $\forall t \in T, \neg M_i[t\rangle$ ($M_i \in V$), then M_i is the "end marking". $\{0, 1, ..., n\}$ is a set of nodes, set 0 as a root node. $(0, i)$ is a directed route from 0 to i. $\forall M_i, M_j \in V$, if $M_i = M_j \wedge i \neq j$, and $(0, i)$, $(0, j)$ are the same route from root node 0, then M_i and M_j can be composed as one node "$M_{i,j}$" [24].

$E = \{\langle M_0, M_1 \rangle_{t_1}, \langle M_1, M_2 \rangle_{t_2}, ..., \langle M_i, M_j \rangle_{t_n}\}$ is a set of arcs which are labeled with fired transition $t_a \in T$, $M_i [t_a \rangle M_j$ ($M_i, M_j \in V$). $t_a \wedge \neg t_b$, $t_a \vee t_b$ represent conflict and concurrency between transitions.

According to the algorithm in [23, 24], we can construct CG of the sub-model, as shown in Figure 10. The transitions in Figure 10 are listed in Table 3.

Table 3. Transitions of Figure 10

Transitions	Descriptions
t_1	Choose mode
t_2	Launch system
t_3	Detect environment
t_4	Detect temperature
t_5	Detect smog
t_6	Detect noxious gas
t_7	Multi-sensor fusion
t_8	Detect again
t_9	Set fire alarm
t_{10}	Open extinguisher
t_{11}	Send fire alarm to owner

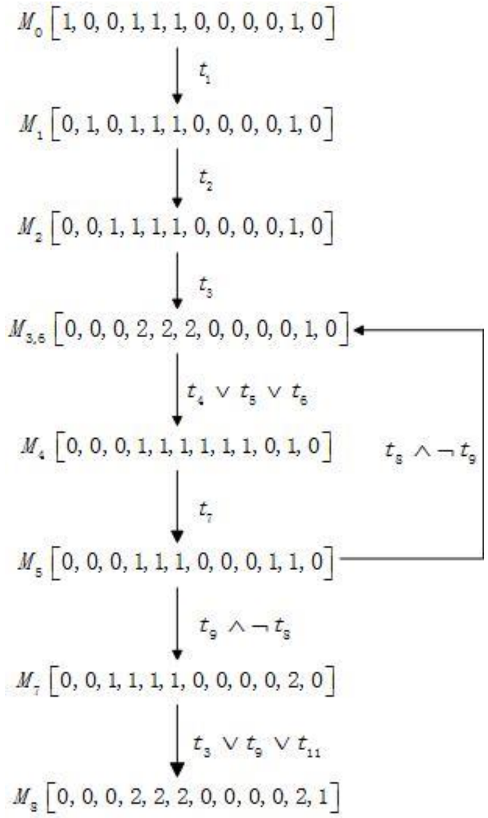

Figure 10. CG of Fire-alarm Sub-model

The maximum number of token in M_i of Figure 10 is 2, the minimum number is 0, and all the markings are reachable. So, the Fire-alarm sub-model is bounded and reachable.

There is only one "deadlock" M_8 which is the end marking in the sub-model. In M_8, all the Context-aware Service operations are executed successfully, the rest of things are the business of homeowner and firefighters. So, there is no "deadlock" during the execution of the operations.

So, we can conclude that the entire leave home scenario model is bounded, reachable, and no "deadlock" during the execution of the operations. In order to verify this conclusion, we use CPN Tools to generate State Space Report of this model.

The State Space Statistics of the State Space Report are listed in Table 4. There are 10309 Nodes and 26826 Arcs in State Space, 6968 Nodes and 20243 Arcs in Scc (Strongly-connected-component) Graph. This means that the model has finite Nodes, Arcs, and cycles. This is correct due to the randomness of the temperature and other context information which were used in this model. We can also find a cycle in Figure 10.

Table 4. State Space Statistics

	Nodes	Arcs
State Space	10309	26826
Scc Graph	6898	20243

The Boundedness Properties are listed in Table 5. The entire leave home scenario model is bounded because the Upper of the numbers is 2, and the Lower is 0.

Table 5.Boundedness Properties

Best Integer Bounds	Upper	Lower
P1	1	0
P2	1	0
P3	1	0
P10	1	0
Pt	2	1
Ps	2	1
Pg	2	1
P4	1	0
P5	1	0
P6	1	0
P7	1	0
P8	2	1
P9	1	0
Pi	2	1
Pv	2	0
P11	1	0
P12	1	0
P13	1	0
P14	1	0
P15	1	0

The Home and Liveness Properties of the State Space Report are listed in Table 6. The leave home scenario model has none Home Marking and only one Dead Marking [9507]. It is correct in this model also due to the randomness of the context information. If the inference rules are not satisfied, the detection tasks will be continually executed. So, there is none Home Marking in this model and the only one Dead Marking [9507]. The Dead Marking is the final state, which indicates that all the operations are executed successfully. There is no Dead Transition Instance, all the Markings are reachable.

Table 6. Home and Liveness Properties

Home Markings	Dead Markings	Dead Transition Instances	Live Transition Instances
None	[9507]	None	None

5. Conclusion

In this paper, we have proposed a Smart Home Context-aware modeling approach based on UML and CPN. UML can not only facilitate requirement analysis when design Context-aware Systems, but also effectively avoid “deadlock” when only use PN. A possible shortcoming of UML is that it cannot verify the system dynamically and formally. CPN is an appropriate approach to model Context-aware system due to the formal verification, dynamic simulation, and other properties. But when model complex system, CPN may cause “deadlock” or “conflict” as the requirement analysis of the system is not understood clearly. The proposed modeling approach combines the merits of these two approaches, effectively avoids “deadlock”, and verifies the model formally.

Support context reasoning is one of the requirements of context modeling, and context reasoning is another key feature of Context-aware System. So we will focus on the context reasoning technology based on this modeling approach in the further work.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61170117 & 61432004), thanks for financial aid for the research.

References

[1] C. Perera, A. Zaslavsky, P. Christen and D. Georgakopoulos, “Context Aware Computing for The Internet of Things: A Survey”, Communications Surveys & Tutorials, IEEE, vol. 16, no. 1, **(2014)**, pp. 414-454.

[2] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith and P. Steggles, “Towards a better understanding of context and context-awareness”, Lecture Notes in Computer Science, vol. 1707, **(1999)**, pp. 304-307.

[3] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan and D. Riboni, “A survey of context modeling and reasoning techniques”, Pervasive and Mobile Computing vol. 6, no. 2, **(2010)**, pp. 161-180.

[4] K. Henricksen, “A framework for context-aware pervasive computing applications”, Queensland: University of Queensland, **(2003)**.

[5] S. Han and H. Y. Youn, “Petri net-based context modeling for context-aware systems”, Artificial Intelligence Review, vol. 37, no. 1, **(2012)**, pp. 43-67.

[6] S. Ahn and D. Kim, “Proactive context-aware sensor networks”, Wireless Sensor Networks, Springer Berlin Heidelberg, vol. 3868, **(2006)**, pp. 38-53.

[7] S. Han and H. Y. Youn, “Modeling and Analysis of Time-Critical Context-Aware Service Using Extended Interval Timed Colored Petri Nets”, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions, vol. 42, no. 3, **(2012)**, pp. 630-640.

[8] J. Sun, Y. Zhang and K. He, “A petri-net based context representation in smart car environment”, Advances in Grid and Pervasive Computing, Springer Berlin Heidelberg, vol. 6104, **(2010)**, pp. 162-173.

[9] A. K. Nabih, M. M. Gomaa, H. S. Osman and G. M. Aly, “Modeling, simulation, and control of smart homes using petri nets”, International Journal of Smart Home, vol. 5, no. 3, **(2011)**,.

[10] P. Reignier, O. Brdiczka, D. Vaufreydaz, J. L. Crowley and J. Maisonnasse, “Context-aware environments: from specification to implementation”, Expert Systems, vol. 24, no. 5, **(2007)**, pp. 305-320.

[11] Z. Hu, T. Lu and Z. Zhao, “Context-aware service system modeling using timed CPN”, Service Systems and Service Management (ICSSSM), 10th International Conference on IEEE, **(2013)**.

[12] J. L. Silva, Ó. R. Ribeiro, J. M. Fernandes, J. C. Campos, M. D. Harrison, “The APEX Framework: Prototyping of Ubiquitous Environments Based on Petri Nets”, Human-Centred Software Engineering, Springer Berlin Heidelberg, vol. 6409, **(2010)**, pp. 6-21.

[13] H. Haiouni and R. Maamri, "Context-Aware Automatic Service Selection Mechanism for Ambient Intelligent Environments", Ambient Intelligence-Software and Applications, Springer International Publishing, vol. 291, **(2014)**, pp. 1-11.
[14] K. O. Byung, "Modeling and generating context-aware agent-based applications with amended colored Petri nets", Expert Systems with Applications, vol. 27, no. 4, **(2004)**, pp. 609-621.
[15] "Unified modeling language (uml)", Available: http://www.uml.org/.
[16] K. Jensen and L. M. Kristensen, "Coloured Petri nets: modelling and validation of concurrent systems", Springer, **(2009)**.
[17] H. Störrle, "Semantics and verification of data flow in UML 2.0 activities", Electronic Notes in Theoretical Computer Science, vol. 127, no. 4, **(2005)**, pp. 35-52.
[18] T. S. Staines, "Intuitive mapping of UML 2 activity diagrams into fundamental modeling concept Petri net diagrams and colored Petri nets", Engineering of Computer Based Systems, ECBS, 15th Annual IEEE International Conference and Workshop, IEEE, **(2008)**.
[19] A. S. Staines, "A triple graph grammar (TGG) approach for mapping UML 2 activities into Petri nets", Proceedings of the 9th WSEAS international conference on Software engineering, parallel and distributed systems, World Scientific and Engineering Academy and Society, **(2010)**.
[20] K. Jensen, L. M. Kristensen and L. Wells, "Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems", International Journal on Software Tools for Technology Transfer, vol. 9, no. 3, **(2007)**, pp. 213-254.
[21] D. Raptis, N. Tselios and N. Avouris, "Context-based design of mobile applications for museums: a survey of existing practices", Proceedings of the 7th international conference on Human computer interaction with mobile devices & services, ACM, **(2005)**.
[22] S. Yanwei, Z. Guangzhou and P. Haitao, "Research on the context model of intelligent interaction system in the internet of things", IT in Medicine and Education, International Symposium, IEEE, **(2011)**.
[23] T. Murata, "Petri nets: Properties, analysis and applications", Proceedings of the IEEE, **(1989)**.
[24] C. Yuan, "The theory and application of Petri Net", Publishing House of Electronics Industry, **(2005)**.
[25] Y. Yu and X. Chen, "UML modeling language and its development tools Rose", China Railway Publishing House, **(2011)**.

Authors

Xiaopeng Niu, he is a Ph.D. student in School of Computer and Communication Engineering, University of Science and Technology Beijing. His research interests include Pervasive Computing, Internet of Things, machine learning, Human-Computer Interaction, Artificial Intelligence.

Zhiliang Wang, he is a professor in School of Computer and Communication Engineering, University of Science and Technology Beijing. His research interests include Affective Computing, Internet of Things, and Human-Computer Interaction.