ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

A Rule Verification and Resolution Framework for Smart Buildings

Jayashree S. Somani[1], Prof. V. L. Kolhe[2]

PG Student, Department of Computer Engineering, D. Y. P. College of Engineering, Akurdi, Pune, Maharashtra, India[1]

Assistant Professor, Department of Computer Engineering, D. Y. P. College of Engineering, Akurdi, Pune, Maharashtra, India[2]

ABSTRACT: IOT(Internet of Things) mainly consists of the wireless Sensor-Actuator Networks. The sensor senses the environment and gives the signals(data) to the server. The server processes this data and gives command to the actuator to take the specified action.Wireless sensor-actuator networks improve the practicality and flexibility of smart building systems. In smart buildings the services are nothing but the rules which are achieved by rule analyzing and executing. But sometimes, in the system some confusion and maloperation takes place because of the irrational contents of the rules and the conflicts among the rules. This paper includes description about a lightweight rule verification and resolution framework to solve the problems of confusion and confliction among the rules. This framework can balance the verification quality with speed and guarantee the rule system working appropriately.

KEYWORDS: IOT, smart building, rule verification in creating, rule verification in executing, rule conflict, content anomaly

I. INTRODUCTION

In a smart building system, to achieve flexible control and management, control logics are usually expressed in the form of service rules and scheduled by a rule engine. However, the increase of users as well as the uncertainty of user's demands will bring many problems. With the increasing scale of system, a large number of users appear, and users can customize their exclusive services according to their own needs. However, service increase and customization randomness make the service execution in smart building systems face severe challenges.

Because different users have different individual demands and service contents customized are various, there may exist conflict between services. For example, the user A subscribes service 1 "if smoke concentration from smoke sensor is greater than 0.65% FT, then open window". In this service, the smoke sensor and the actuator responsible for opening window are directly involved. The trigger condition is "if smoke concentration from smoke sensor is greater than 0.65% FT"; and the execution behavior is "open window". The user B subscribes service 2 "if temperature is less than 20□, then close window". When the room temperature captured by temperature sensors meets the trigger condition of service 2, window will close. At this moment, there might exist someone who smokes in this room. The value of smoke also meets the trigger condition of service 1, then the action of opening window will be executed. In this situation, a conflict between two services happens. Since the rules are added by users individually, manual input may lead to fuzzy logic or inaccurate input, while multi-user rule making may result in rule redundancy or content conflicts between rules, and thus reduce operation efficiency even bring abnormal operation. So, by rule verification and resolution framework, we can solve such problems.

II. RELATED WORKS

Smart building system, as an important part of IoT, has been researched and explored for years [2]. Jafer et al. [3] presented a building automation system based on WSANs, which can be used to build various indoor monitoring applications such as resident tracking, energyefficient electrical appliance control and building security. Recently, the research fields related to smart building systems have been extensive, such as activity recognition [4], service discovery and composition [5], as well as efficient rule engine [6]. Chen et al. [4] introduced an ontology based hybrid approach

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

to activity modeling that combined domain knowledge based model specification and data-driven model learning. Although the rules can be extracted through daily activities learning and recognition, in a smart building system, most of the rules are created by numerous users. On one hand, manual input may cause inexact logics or input errors. On the other hand, multi-user rule making may bring content conflicts among multiple rules. Hence, rule conflict detection is essential to ensure the correctness of rule execution [7].

Ma et al. [8] presented a rule-map based technique for data inconsistency detection, where rule-map was used to describe the hierarchical structure of rules and estimate judgment standards for consistency dynamically. Shehata et al. [9], [10] proposed a semi-formal method to detect policy interaction in smart home with a scenario-based approach, which can be applied to detect policy interaction. Leelaprute et al. [11] proved a classification method and resolution schemes of feature interaction in home network systems, and further introduced model checking techniques to automatically analyze possible feature interaction [12]. Hu et al. [14] proposed a complementary methodology SPIDER, to detect the feature interaction problem on service composition of smart home.

Yan et al. [15] explore rule verification mechanism with considering both rule content anomaly and rule conflict, and present the rational degree of rule in a probability way.

III.RULE VERIFICATION AND RESOLUTION SYSTEM

A rule verification system(RVS) contains three modules: 1) Rule verification in creating(RV_C) module 2) Rule verification in executing(RV_E) module and 3) Conflict resolution (CR) module. The conflicts among the rules are classified into two types – Rule conflicts on device and Rule conflicts on environment. Rule conflict on device is the direct conflict between the two rules R1 and R2 for one device d. Rule conflict on environment is the conflict between R1 and R2 for different devices d1 and d2. This type of conflict occurs under the environment parameter confusion which is controlled by two devices. The RVS verifies rule conflicts on device in RV_C module and rule conflicts on environment in RV_E module. If any rule conflicts occurs, then the CR provides conflict resolution strategies to resolve the conflicts and improve the effectiveness of system. Following Figure1 represents the flow diagram of Rule verification and resolution Framework:

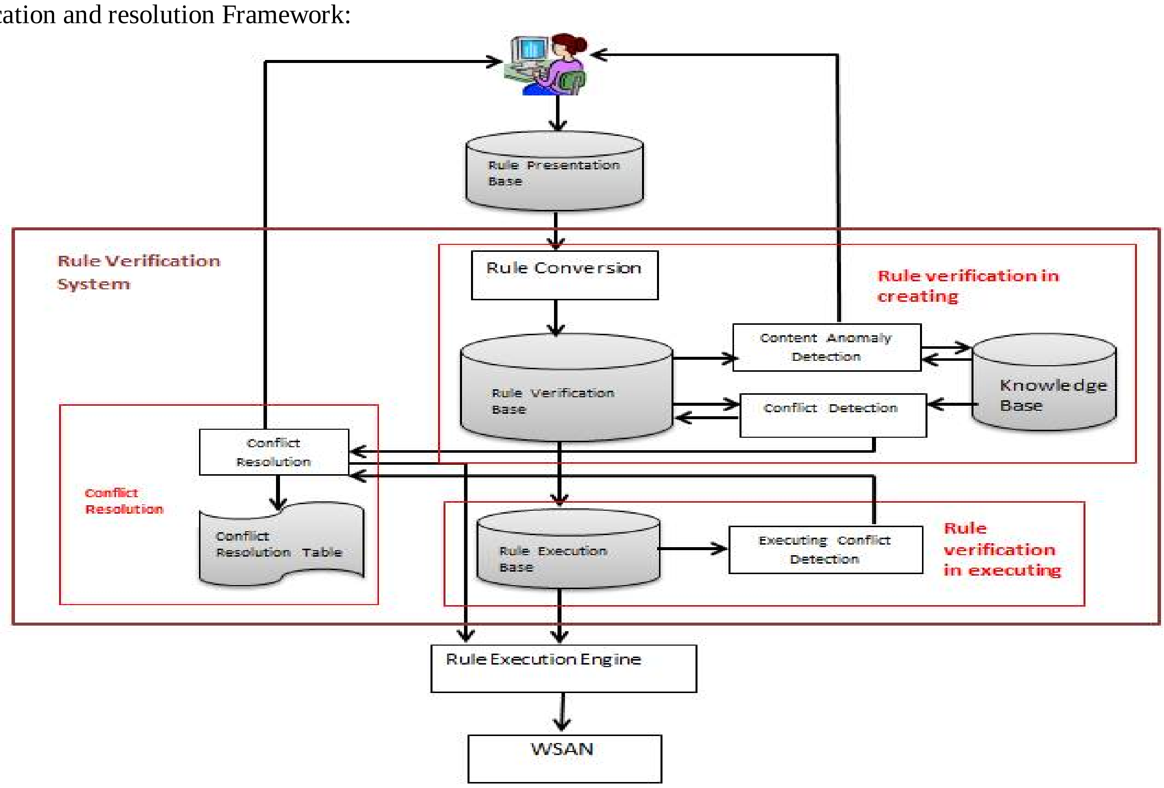

Fig.1: Flow Diagram of Rule Verification and Resolution Framework

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

To improve the efficiency of the rule verification, the system includes two-segment rule verification. As already mentioned, the Rule Verification system has three module – Rule verification in creating module, Rule verification in Executing module and Rule conflict resolution module.

A. Rule Verification In Creating:

When any user adds the rule through human–computer interface, then it is firstly taken into the rule verification in creating module where the rule is firstly decomposed into simple and smaller forms of rules with the help of Rule Conversion (RC). Then this semi-formal rule description is transferred to the verification oriented formal rule representation. After this, the Content Anomaly Detection(CAD) detects whether the contents of the transformed rule is inconsistent with the domain knowledge in the Knowledge Base (KB). As the Rule Verification in Creating module detects the conflicts of rules on device, the Creating Conflict Detection (CCD) compares the rule with the other related rules in Rule Verification Base (RVB) to perform the conflict detection and analysis. If the conflict exists, then CCD will calculate the degree of confliction and reports the user at the same time. If the confliction does not exist, then the new rule is added to the RVB. Here, the principles, common sense, rational values and ranges of referenced parameters in the system are recorded in the Knowledge Base.

<u>*Algorithm:*</u>

```
Input: Rule X
           LocationTree
Begin
R←RuleConvert(X)
Ano←ContentAnomalyDetection(R)
report Ano
node←R.posID //get the position of R in LocationTree
while node is not LocationTree.root do //reverse traversal the tree till root
 for each rule ER of node do
  DeviceConflictDetection(R, ER)
  if detect device conflict then
   ConDeg←ConflictDegreeCompute(R, ER) //compute conflict degree
    Res←ConflictResolution(R, ER) //get the most suitable resolution
    report ConDeg and Res
  node←node.parent
subTree←thesubtree of LocationTree with root R.posID
for each node in subTree do
  for each rule ER of node do
   DeviceConflictDetection(R, ER)
   if detect device conflict then
    ConDeg←ConflictDegreeCompute(R, ER)
    Res←ConflictResolution(R, ER)
    report ConDeg and Res
End
```

B. Rule Verification In Executing:

After being triggered by the Rule Verification in Creating (RV_C) Module, the rule will be sent to the Rule Verification in Executing (RV_E) Module. The RV_E module detects the rule conflicts on environmental parameters. In this, the rules are first taken into the Rule Execution Base where the rules which are going to execute are stored and then they are transferred to the Executing Conflict Detection (ECD), which detects the rule conflicts again to make sure that the execution of this rule won't cause failure to the system.

<u>*Algorithm:*</u>

```
Input: XML-format rule R
Begin
for each existing rule ER in ExecutionBasedo
```

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

2 if R.ServiceID equals to ER.ServiceID then
3 if R.ControlArea overlaps with ER.ControlArea then
4 if R.Action.Statething equals to ER.Action.Statething then
5 report Write-Write Conflict
6 else if R.TriggerEvent.Statething equals to ER.Action.Statething then
7 report Read-Write Conflict
End

C. Conflict Resolution:

If the conflict exists in RV_C and RV_E modules, then the rule is sent to the Conflict Resolution (CR) module. This module works as the assistance of the RV_C and RV_E modules. CR provides rational conflict resolution strategies to improve the effectiveness of the system. If the conflict is resolved then it is send to the Rule Execution Engine (REE) otherwise, it reports the user about this conflict. Then the REE transfers this verified rule to the Wireless Sensor Actuator Network (WSAN) which then gives the command to the devices to take action.

Algorithm:

Input: XML-format rule R, ER
Begin
1 getR.ServiceID and ER.ServiceID
2 get R.ServiceDescription.ServiceType and ER.ServiceDescription.ServiceType from KnowledgeBase according to ServiceID
3 get R.User and ER.User
4 if R.ServiceDescription.ServiceType equals to ER.ServiceDescription.ServiceType then
value.service←s
6 else
value.service←m
8 if R.User equals to ER.User then
value.user←s
10 else
value.user←m
12 compare (value.service, value.user) in ResolutionTable
13 Res←corresponding resolution method
End

IV. CONCLUSION

In this paper, we have discussed about the lightweight rule verification and execution framework which verifies the rules and avoids the confusion and maloperation occurs in the system because of the irrational contents of the rules and the conflicts among the rules. This system verifies the correctness of the rules so that the system will work smoothly and will achieve the flexible control and management on rules.

REFERENCES

[1] T. Wark, D. Swain, C. Crossman, et al, “Sensor and Actuator Networks: Protecting Environmentally Sensitive Areas,”. IEEE Pervasive Computing, vol. 8, issue 1, pp.30–36, January-March 2009. (Conference proceedings)

[2] H. Lee, Y.K. Jeong, I.W. Lee, “A Mechanism of Ontology-based Rule Management for Smart Building Energy Saving Service,” 2012 International Conference on Digital Object Identifier, pp. 737-738, 2012. (Conference proceedings)

[3] E. Jafer, B. OFlynn, C. OMathuna, W. Wang, “Design of Miniaturized Wireless Sensor Mode and Actuator for Building Monitoring and Control,” 17th International Conference on Telecommunications, Doha, Qatar, pp. 887-892, 2010. (Conference proceedings)

[4] L.M. Chen, C. Nugent, G. Okeyo, “An Ontology-Based Hybrid Approach to Activity Modeling for Smart Homes,” IEEE Transactions on Human-Machine Systems, vol. 44, Issue1, pp. 92-105, 2014.(IEEE Transactions)

[5] S.N. Han, G.M. Lee, N. Crespi, “Semantic Context-Aware Service Composition for Building Automation System,” IEEE Transactions on Industrial Informatics, vol. 10, Issue 1, pp. 752-761, 2014. (IEEE Transactions)

[6] R. Velik, H. Boley, “Neurosymbolic Alerting Rules,” IEEE Transactions on Industrial Electronics, vol. 57, Issue 11, pp. 3661-3668, 2010. (IEEE Transactions)

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

[7] Y.m. Xu, W.J. Niu, H. Tang, "A Policy-Based Web Service Redundancy Detection in Wireless Sensor Networks," Journal of network and systems management, vol. 21, Issue 3, pp. 384-407, 2013.(Journal)
[8] J. Ma, J. Lu, and G. Zhang, "A Rule-Map based Technique for Information Inconsistency Verification," In Proc. of 2007 Information, Decision and Control, IEEE Computer Society, Adelaide, Australia, pp. 296-301, 2007. (Conference proceedings)
[9] M. Shehata, A. Eberlein, and A. Fapojuwo, "Using semi-formal methods for detecting interactions among smart homes policies," Sci. Comput.Program., vol. 67, Issue 2-3, pp. 125–161, July 2007.(Journal)
[10] M. Shehata, A. Eberlein, and A. Fapojuwo, "A taxonomy for identifying requirement interactions in software systems," Computer Network, vol. 51, Issue 2, pp. 398–425, February 2007.(Journal)
[11] P. Leelaprute, "Resolution of feature interactions in integrated services of home network system," in Proc. Asia-Pacific Conf. on Communications,Bangkok, Thailand, pp. 363–366, October 2007. (Conference proceedings)
[12] P. Leelaprute, T. Matsuo, T. Tsuchiya, and T. Kikuno, "Detecting feature interactions in home appliance networks," in Proc. ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Phuket, Thailand, pp. 895–903, August 2008. (Conference proceedings)
[13] H. Luo, R.S. Wang, X.M. Li, "A Rule Verification and Resolution Framework in Smart Building System", 19th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2013), Dec. 2013, Seoul, Korea. (Conference proceedings)
[14] H. Hu, D. Yang, L. Fu, et al. "Semantic Web-based policy interaction detection method with rules in smart home for detecting interactions among user policies," IET Communication, vol.5, Issue 17, pp. 2451-2460, November 2011.(Journal)
[15] Yan Sun, Tin-Yu Wu, Xinming Li, and Mohsen Guizani "A Rule verification System for Smart Building" IEEE Transactions on Emerging Topics in Computing, vol. 5,Issue 3, pp. 367-379, February 2016. (Magazines and Journals)