Rule Conflict Detection and Resolution Technology in Smart Home IoT

Hanwen Xia

{xhw9648@gmail.com}

School of Computer Science, School of Cyber Science and Engineer, Nanjing University of Information Science & Technology, Nanjing, Jiangsu, China

Abstract. With technological advancements, smart homes are becoming increasingly widespread and integrated into daily life. Their operation relies on rule constraints, making rule formulation a crucial aspect that urgently needs optimization in the development of IoT-based home appliances. During device operation, the same device may generate conflicting instructions due to different rules. Effectively identifying and handling rule conflicts has become an indispensable key issue in IoT home rule formulation. The paper begins with a definition of the rule conflicts in the IoT home devices and typical types of rule conflicts. It subsequently discusses and examines the approaches to identifying conflicts of rules and how conflicts of rules are found in homes. Moreover, the existing mainstream rule conflict resolution methods, i.e., the static resolution and dynamic resolution, are reviewed and analyzed in this paper. Lastly, this paper provides a conclusion of the key dilemmas that are being encountered as well as the outlook of the various trends as far as the future direction of the rule conflict detector-resolution strategies goes.

Keywords: Smart home; Rule conflicts; Conflict detection; Conflict resolution.

1. Introduction

With the development of IoT furniture devices, human-computer interaction is increasingly used in the lives, and IoT furniture devices are a good example. In IoT furniture devices, users need to set rules for IoT furniture devices [1] to make the furniture devices run in the direction they like. Such human-computer interaction scenarios frequently occur in IoT furniture usage scenarios, but they also bring some problems [2]. The linkage complexity of existing IoT furniture devices is rapidly increasing. In a home, the IoT furniture network may connect to more than ten kinds of furniture devices at the same time. Various sensors or control terminal devices will give various instructions to each furniture machine and the instructions regulate the conduct of the devices. Thus, two or more conflicting commands are highly likely to be issued to the same device in close succession, as in the case of the curtains, where there can be two conflicting instructions issued to it, namely open and close. How to deal with mixed orders is a pressing issue that has to be addressed.

ICIAAI 2026, May 22-24, Singapore, Singapore
Copyright © 2026 EAI
DOI 10.4108/eai.22-5-2026.2365091

When a set of rules is activated either at the same time or in a sequence, there can be inconsistencies in the actions taken by the system when they are executed. As an illustration, there is an automatic rule whereby at temperatures higher than 25 ℃, the air conditioner will be on to reduce the temperature. However, simultaneously, once the temperature decreases to lower than 25 o C, another automatic regulation will be activated. The complexity of rule interactions and the complex nature of devices themselves, due to the growing complexity of IoT furniture networks, has also caused a rising issue of rule conflicts. These may lead to failures and even breakage of domestic appliances, making the user experience a poor one as well as resulting in safety risks. In the meantime, not all users have the expertise to compose rules, and are restricted to the simple and general rules offered by the platform, which are not able to cater to their unique and varied lifestyle desires [3]. Up to this time, this issue has drawn a mass of attention from industry participants both locally and worldwide and a verifier of solutions has risen. Nonetheless, these studies are still disjointed and inactive, as they do not have systematic organization and summarization. Thus, the rationale behind this paper is to synthesize, categorize, and survey applicable research on how to resolve conflicts of rules in IoT furniture devices and thus, form the basis on which future developments can be based.

The proposed paper will serve the purpose of giving the researchers in this field a base of knowledge to build upon, to enable new entrants into the industry grasp the development of this industry very fast and provide direction to upcoming innovative researchers. It also has the aspiration to offer a remedy to the present IoT furniture trades, better use of the existing products, and technical references, hence creating safer, dependable, and user-friendly IoT products at this point and enhancing the user experience.

2. Classification of rule conflict

IoT devices also have numerous forms of rule conflicts and various methods of classification [3].

Execution conflict. The conflict can happen within a rule system between two or more rules acting on the same sensor variable at the same time and their triggering conditions overlap in the value range and the two associated execution actions are at odds. Various rules are triggered concurrently in this case, however, the system cannot perform competing tasks simultaneously, resulting in failure of the decision.

Using the example of the rules in Table 1, the rules will be thrived together when the soil moisture is in the overlapping area between 30 and 40. One of them is rule 1, which says to turn on the sprinkler, and the other is rule 2, which says to shut-off sprinkler. The conflict between these two rules at the level of their execution is absolute and the system cannot meet in the same time these two opposing requirements of action, therefore, creating a standard execution conflict.

Shadow conflict: A situation of a rule system is called shadow conflict when an entire interval of sensor values that the triggering condition of one rule satisfies is fully within a triggering interval of a second rule, and the actions induced by the two rules are identical. There are also only a few rules that significantly influence this case, and the rules included can not be activated independently. This is literally a redundancy of a rule, which will lead to the failure of the rule.

Table 1. Example of execution conflict

Rule 1	If the soil moisture is less than 40%, turn on the sprinkler.
Rule 2	If the soil moisture is >30%, turn off the sprinkler.

Taking the rules in Table 2 as an example, rule 2, soil moisture less than 20%, is completely within the range of rule 1, soil moisture less than 30%. Rule 1 has ensured that the sprinkler is turned on when the moisture is below 30%. Therefore, rule 2 is redundant because the sprinkler has already been turned on by rule 1 when the soil moisture is less than 30%. Rule 2 is completely contained by rule 1 [3].

Table 2. Example of shadow conflict

Rule 1	If the soil moisture is less than 30%, turn on the sprinkler.
Rule 2	If the soil moisture is less than 20%, turn on the sprinkler.

Independent conflict refers to a situation in a rule system where two rules act on the same sensor and their triggering conditions may be met simultaneously under different circumstances. However, the instructions of these two rules to the sensor are exactly opposite. For example, in the rules in Table 3, when the soil moisture is below 20% in rule 2 and there is a weather forecast of heavy rainfall, rule 1 is triggered. The system will simultaneously issue instructions to the sprinkler to turn it off and on. However, the system cannot execute conflicting operations at the same time, which leads to decision failure [3].

Table 3. Example of independent conflict

Rule 1	If the weather forecast predicts rainfall of more than 5 millimeters in the next hour, turn off the sprinklers.
Rule 2	If the soil moisture is less than 20%, turn on the sprinkler.

The occurrence of rule conflicts is not accidental, but rather a concentrated manifestation of the inherent complexity of smart home systems at the technological, interactive, and ecological levels. The root cause of the conflict lies in two characteristics of the Internet of Things (IoT) environment: resource sharing and automation and concurrency. During the operation of IoT devices, multiple IoT device applications, rules, and services run in the same physical environment. Therefore, when multiple rules or applications are activated concurrently, they are very likely to issue commands to shared resources simultaneously without coordination, thus triggering conflicts.

3. Methods for detecting rule conflicts

Rule conflict detection is the first line of defense in building a reliable smart home system. Its goal is to identify potential risks before or during the execution of conflicting rules. With indepth research, detection methods have evolved from early static logic checks to a multi-dimensional technical system integrating formal verification, semantic reasoning, and data-driven approaches. The specific classifications of conflict detection methods are shown in Table 4 below, mainly divided into three categories: formal verification methods, ontology and knowledge reasoning methods, and data-driven and machine learning methods.

Table 4. Classification of rule conflict detection methods

Method Category	Core idea	Key technologies& representative models	Advantages	Limitation
Formal verification methods[4]	Abstract the system into a mathematical model and prove its security through exhaustive enumeration or logical reasoning.	Model checking, theorem proving, Petri nets, and temporal logic (LTL, CTL)	It is rigorous and comprehensive, providing formal security guarantees.	State space explosion is a fundamental bottleneck, severely limiting its application in large and complex systems. Furthermore, it struggles to effectively model real-world uncertainties such as sensor noise and the randomness of user behavior[5].
Ontology and knowledge reasoning methods [6]	Construct a domain knowledge graph and use an inference engine to check semantic consistency.	OWL ontology, SWRL rules, description logic, inference engines (such as Pellet, HermiT)	It endows the system with the ability to "understand," enabling it to discover semantic-based target conflicts and indirect conflicts, and the reasoning process is interpretable.	Knowledge engineering is extremely burdensome. Building and maintaining a high-quality, complete ontology requires the participation of a large number of domain experts and is difficult to scale with device updates. The complexity of reasoning also increases dramatically with the size of the ontology [6].
Data-driven and machine learning approaches [7]	Learn normal or conflict patterns from historical data to make predictions or classifications.	Association rule mining, sequence pattern learning, graph neural networks, anomaly detection	Its powerful ability to learn from data enables it to adapt to changes in the environment and the evolution of user habits, and it is expected to discover previously unseen patterns of conflict.	Performance is highly dependent on a large amount of high-quality labeled data, and it performs poorly in the cold start phase. Its biggest controversy lies in the "black box" decision-making process, making it difficult for users to understand why the system determines a conflict, thus damaging trust. Hybrid systems are an important direction for solving this problem[7].

4. Rule conflicts resolving strategies

Presently, at least there are several strategies of the resolution of the rule conflicts, which may be roughly divided into two approaches: one of dynamic resolution and another one of static resolution. The initial one is the type of static resolution that is used; it entails the pre-elimination of possible conflicts to the system by means of reconstruction and sorting before deployment or loading of rules to the system. They are priority sorting, rule synthesis and reconstruction, and weight-based pre-resolution. The following is an appropriate instance of static resolution: Zhang Lin et al. [8] suggested that rule weights are computed according to the TF-IDF concept, and then integrated with the K-modes clustering algorithm in clustering, which effectively resolves the possible conflicts within a large-scale rule set prior to the deployment of the system.

Different approaches to weight-based and clustering based, weight-based and clustering based are common and typical approaches in resolving conflicts. According to this approach, the two-phase approach involves measuring the weight of each strategy, and the second part involves clumping of similar strategies, that is, dividing them into smaller groups, thereby making it far easier to solve conflicts around the world. In the rule implementation, this approach uses the dynamic situation in the decision-making process, and decisions are made on conflicts that are not in the static phase or are even emerging. Its strengths are high flexibility and the capacity to manage unpredictability and the versatility of preference. It has a low level of runtime computational overhead, but it is also characterized by the drawback of possible decision lag. In particular, this approach computes the TF (Term Frequency, or weight of the attribute in the individual strategy) and the IDF (Inverse Document Frequency, or the weight of the attribute-value pair compared to the whole strategy library) of the attributes in order to get the weight of a given pair of attribute-value. The total weight of no strategy is 0, since the weight of any attribute, no strategy, is 0. The weight of any strategy is equal to the sum of its attribute-value weights. The larger weights represent a more precise and significant strategy. This approach lifts the conflict resolution process to the level of a quantitative decision-making between the options of what is better, instead of the option of do or not and the concept may be applied to the typical home rule settings, where invisible weights like user comfort and the power required by the device are used.

The other method is the dynamic resolution, which recognizes that not everything that is run time can be predicted and thus makes decisions depending on the real time state upon conflict. A suitable instance of dynamic resolution is here: [9] Dynamic resolution algorithm for multi-objective optimization. This is a dynamic resolution strategy that assumes solving the conflict with a dynamic multi-objective optimization problem. The system takes environmental measurements in real-time and autonomously adjusts a combination of best equipment control parameters (like temperature and brightness of lights) based on optimization algorithms, arriving at the most effective compromise between a number of competing goals. This is based on objective functions such as comfort deviation functions and energy consumption functions, which are solved with the use of genetic algorithms against complex, nonlinear multi-objective optimization problems to produce a set of elegant solutions, known as the Pareto optimal solution set, which users may select. The algorithm operates by first getting environmental information on a system state shift, solving the primary set of Pareto optimality solutions with an adaptive genetic algorithm and then solving a single solution for implementation within the solution set according to the personal requirements of the user in terms of targeting energy saving this month. Meanwhile, the user can at any point in time switch to other solutions, thus fulfilling the objective of dynamic conflict resolution. Such a data-driven solution can flex to sluggish evolutions inside the environment and user behavior and does personalized dynamic optimization as well as symbolizes the progression path of the level of automation.

In comparison with the two methods, the method of static conflict resolution, which is realized on the stable attributes and fixed values of the weights of rules, is mainly concerned with prevention, by which it is hoped to reduce the conflict through the design and preprocessing of rules. This is a good conflict resolution approach, which is highly interpretable, well structured and with low operational overheads. In contrast, dynamic conflict resolution is grounded in real-time data and optimization algorithms, which are usually grounded in control. It does real time optimization and trade-offs as conflicts arise, which provides it with more adaptability and the capacity to cope with complicated dynamic conflict situations.

5. Current Challenges and Future Research Directions

The current rules for conflict detection and resolution face many bottlenecks in scalability and computational efficiency. Many existing methods, especially formal verification and ontology reasoning, face the combinatorial explosion problem when the number of rules and devices increases. Static analysis may become computationally infeasible, while the delay of dynamic decision-making will affect the user experience. For example, a smart home with hundreds of devices has an astronomical number of potential rule interaction paths, making exhaustive detection almost impossible. This forces existing research to be verified in small, simplified pseudo-environments, and the effectiveness of its conclusions in large-scale real-world scenarios is questionable. How to achieve efficient conflict management on limited computing resources (such as home gateways) is a key obstacle to practical application. At the same time, the lack of robustness in dynamic and uncertain environments is also an urgent problem to be solved [10]. Real home environments are full of uncertainty, sensors have noise and false alarms (such as identifying people by moving pets), user behavior is accidental, and physical devices may fail. Most static or deterministic logic-based detection methods are helpless against such "soft conflicts". The one false localization of a sensor can lead to a set of false regulations or the system itself is asking the user for arbitration repeatedly, as it cannot distinguish between ambiguous situations, making the automated system seem malfunctioning and putting the onus on the user.

6. Conclusion

Overall, this paper provides a systematic review of the management technologies of the rule conflicts in smart homes. As the observer, indeed, since they can accurately identify the conflict, with the abundance of choices of solutions, this sector has gained valuable research experience, establishing the basis of the creation of a more stable and smarter environment in the house. In the future, the development of smart homes requires more sophisticated solutions. The paper is convinced that a computing architecture that unites edge intelligence with cloud cooperation, coupled with a profound comprehension of the user intent encoded in large language model will play a major role in overcoming the existing bottlenecks. At the same time, the liberation of the value of data in a privacy protection system and a promotion of cross-platform, verifiable policy standards would be the keys to the maturity of the industry and an open one. Finally, technological advancement will also benefit humanistic care and artificial intelligence home systems will not only computerize processes but also successfully fit into family life, be habit-sensitive, adaptable, and reliable home consumers.

References

[1] Shah, T., Venkatesan, S., Ngo, T., Pratima, Neelamegam, K.: Conflict detection in rule based IoT systems. Proceedings of the IEEE International Conference on Consumer Electronics (ICCE). pp. 1–6 (2019)

[2] Huang, B., Chaki, D., Bouguettaya, A., Lam, K.-Y.: A survey on conflict detection in IoT-based smart homes. ACM Computing Surveys. Vol. 56, No. 5, Article 122, pp. 1–40 (2023)

[3] Cheng, Y., Xu, M., Zhang, Y., Li, K., Wang, R., Yang, L.: AutoIoT: Automated IoT platform using large language models. Proceedings of the IEEE/ACM International Conference on Automated Software Engineering (ASE). pp. 1–13 (2024)

[4] Zhang, X., Dong, W., Qi, Z.C.: Conflict detection in runtime verification based on AOP. Journal of Software. Vol. 22, No. 6, pp. 1224–1235 (2011)

[5] Wu, D., Wei, S.J.: Policy conflict detection method based on model checking. Journal of Southeast University (Natural Science Edition). Vol. 43, No. 5, pp. 1022–1026 (2013)

[6] Wang, S.S., Huang, Y.H., Yang, Y.J.: Knowledge reasoning method based on non-revision inconsistent ontology. Ph.D. Thesis, Dalian Maritime University (2014)

[7] Bao, Z., Binbin, B., Fan, W., et al.: Rock: Cleaning data with both ML and logic rules. Proceedings of the VLDB Endowment. Vol. 17, No. 9, pp. 4373–4376 (2024)

[8] Zhang, L., Wang, X.M., Zhao, X.: A conflict resolution method for IoT access control policies based on weight analysis and clustering. Chinese Journal of Computers. Vol. 44, No. 8, pp. 1520–1535 (2021)

[9] Zhao, P., Huang, J., Liu, S., et al.: Multi-objective optimization for energy and comfort management in smart homes using an adaptive genetic algorithm. Proceedings of the IEEE Congress on Evolutionary Computation (CEC). pp. 1–8 (2020)

[10] Al Farooq, A., Al-Shaer, E., Moyer, T., et al.: IoTC^2: A formal method approach for detecting conflicts in large scale IoT systems. Proceedings of the ACM SIGSAC Conference on Computer and Communications Security. pp. 2102–2104 (2018)