Intelligent control system of smart home for context awareness

Minbo Li¹,² and Yu Wu¹

Abstract
In recent years, the rapid development of Internet of Things smart hardware has increased the demand for intelligent control of devices, mainly in the smart home industry. The framework to solve the problem of equipment intelligent control is studied. The advantages and disadvantages of existing context modeling strategies are analyzed. According to the characteristics of household context activities, combined with the design principles of graph databases, a new context modeling method based on object and attribute graph is proposed, which is suitable for Internet of Things scenarios with limited resources. Rule control, mode control, and voice control approaches of smart home interaction are designed. An inference engine is used to map the data of context awareness to the Internet of Things control services that executed automatic control of the system, and a framework of smart control system based on context awareness of Internet of Things is proposed. Considering the behavior habits of users, the concept of user preference is introduced to provide more personalized services. Performance tests with simulated data show that the new context modeling method has a faster system response time than the ontology modeling control mode.

Keywords
IoT, smart home, context aware, intelligent control, context modeling

Date received: 11 December 2020; accepted: 2 February 2022

Handling Editor: Peio Lopez Iturri

Introduction
With the development of intelligent hardware, more diverse sensing devices have emerged in the field of the Internet of Things (IoT), from the original single indicator sensor, such as temperature sensor, to the multi-indicator integrated sensor. The number of smart home sensing devices has increased from a few to hundreds. In addition to sensing devices, smart devices are also growing rapidly, such as smart audio, smart refrigerators, and smart air-conditioners. The development of hardware makes intelligent control possible. According to the research results of literature, the infrastructure of the IoT at this stage can only provide the basic communication function of the device. There are a lack of corresponding realization mechanisms and solutions for intelligent interactions and services in complex environments.

As one of the application fields of the IoT, this problem also exists in the smart home environment. Although several solutions have been proposed by some scholars, most of them are not intelligent enough to realize automatic control and cannot meet the needs of users. Other solutions that provide automatic control are biased to engineering implementation and lack of theoretical methods. What’s more, they are not mature enough in terms of data abstraction and context expression. Therefore, it is necessary to realize...
intelligence of equipment and automatic invocation of services. The research goal in the field of home intelligent control is to provide the required services to users according to their identity, preference, and current environment.

This article presents an adaptive control system of smart home, which call Intelligent Control System (IConS). The system uses the middleware architecture to shield the heterogeneity of devices. Instead of using the ontology method to model the context, the system establishes the context model of smart home by abstracting specific objects involved in smart home. The rules of control logic and user preference model are defined. A new context modeling is proposed based on object-oriented method and attribute graph model. The new modeling approach retains the high portability and high performance of the object-oriented method, and can express the relationship of object attributes with graph database. Using the context model and personalized rules, the system controls the IoT devices automatically without the user’s active control.

The remaining structure of the article is as follows. Section “Related works” covers the brief summary of latest research achievement. In section “The IConS Architecture”, the intelligent control model of context awareness and intelligent control framework are illustrated. In section “Context-aware modeling”, IConS model method based on object and attribute graph is described. We present intelligent control method and prototype system in section “Home Intelligent Control Technology”, and section “prototype system and experimental comparison.”

**Related works**

In order to provide personalized intelligent services, it is necessary to bind the services provided by the device with the needs of users. It is necessary to represent and store information about the users’ current smart home status, including the attributes of each home equipment and its relationships. This involves context modeling issues, which emphasize the “human” role interaction in the IoT. Context modeling was first proposed by Schilit and Theimer. Context Awareness is classified into explicit perception and implicit perception. Explicit perception is obtained directly through devices or network services, such as various environmental parameters information, and time information, etc. Implicit perception requires the users’ manual inputs or deep system analysis to obtain, such as user preference habits. Sun and Sauvola use a four-tuple pattern (Entity Name, Feature, Value, and Time) to describe the data objects corresponding to physical entities, which uses a unique identifier to distinguish physical entities and uses features, values, and time to characterize the context. Strang and Linnhoff-Popien use the CoOL (Context Ontology Language) model language to build ontological inference to share and reuse physical story information. The TEA project of Gellersen et al. used an object-oriented approach to context modeling, and proposed a concept called Cue, which can be seen as a function whose inputs are the value at a certain moment of a physical or logical sensor. A symbolized output is obtained by this function. It is finally interpreted and executed by the system.

Knowledge graph can be used to efficiently manage resources of CPS (cyber-physical system) or IoT, and describe the relationship between their capabilities. Knowledge graph method of CPS resource capability includes both the conceptual knowledge layer and the instance knowledge layer. Table 1 shows the comparison of main modeling methods.

Taking the measures that meet the needs of users according to the current scenario is the key to achieving intelligent control in the IoT scenario. Mo et al.
proposed a framework of context-aware service system that binds the context data model to business service through the configuration model. While ensuring the automatic and all-weather characteristics of the context-aware service, it reduces the degree of system coupling and makes the binding of context and service can be changed flexibly. DogOnt, developed by Politecnico di Torino, supports user control of home devices through semantic models. Based on machine learning methods, Kishore Ramakrishnan et al. built correlations between contexts. The Bayesian network was used to build a model of the user’s historical behavior, and to achieve intelligent control by matching the user’s behavior and contexts. Based on user habits, Chen proposed a BP-Mine framework to recommend life services. Berkane et al. have mapped contexts and services through pattern matching. Uddin introduced a reasoning engine that defined the correspondence between contexts and services by defining rules.

In addition to completing intelligent control according to the current scene, to meet the needs of users from all aspects and provide richer user interaction, some scholars have also researched into the interactive technologies such as gesture control and voice control. Users can control household appliances through gestures. The mapping relationship between gestures and control commands is stored in its database, and a classifier is trained by the random forest method to recognize the gesture information perceived by the depth imaging sensor. Vacher et al. proposed a voice-based home control system that uses deep learning methods to recognize user speech. Mofrad and Mosse presented a method to separate and recognize speech in multi-user, multi-speech contexts.

Many scholars had studied context-aware services, but considering the shortage of computing resources in the IoT context, further analysis is needed from the perspective of performance and reuse. The present research methods based on context modeling are not suitable for IoT context. At the same time, in the field of household intelligent control, it faces the bottleneck of “false intelligence.” It requires users to take the initiative to participate. The degree of its automation is not high, and personalized service is insufficient. Sometimes, the users need to make decision by themselves rather than the intelligent system. A smarter, full-service, automated control and high-performance home control system is needed.

Based on the above research, this article follows the research of the IoT context area, classifies and abstracts the entities involved in the home interaction context, and proposes a new context modeling method called Intelligent Control System (IConS). The IConS model combines the advantages of the attribute graph model with the object-oriented method, retaining high portability and high performance. The rule engine is used to map the context to the appropriate control service through the rule to achieve automatic control of the system. For personalized services, on the basis of binding users and rules, the concept of user preferences is introduced to enhance home intelligence, and graph database is introduced to improve the system performance.

The IConS architecture

In this section, a new context modeling method based on object and attribute graph model is proposed for home interaction context.

The focus of IoT intelligent control is to analyze the potential needs of users according to the current situation, invoke the control services of the corresponding devices, and achieve the goal of automated control. Therefore, it can be divided into three steps. The first step is to model the application context so that the current interactive context information can be obtained based on sensor data. The second step is to produce the control decision based on the context information, and the last step is to call specific equipment to complete the round of service.

Intelligent control model

The basic process of context awareness includes data acquisition, data processing, scenario generation, and scenario reasoning, as shown in the following formula

\[
C = M_{D\rightarrow C}(D) \tag{1}
\]

\(D\) represents the data space in the context environment, including all the data that can be collected and can be provided to the system, given in the form of key-value pairs, as shown in formula (2)

\[
D = \{(key_1 : value_1), \cdots, (key_n : value_n)\} \tag{2}
\]

\(key_i\) refers to the monitored indicator. For example, the temperature sensor corresponds to temperature, the weather service corresponds to weather, and their values’ range also depends on the types of indicators.

\(M\) maps the data space \(D\) to the context space \(C\). The original collected data are used as inputs, and the outputs obtained through the mapping are context \(C\), which are also the ultimate goal of context modeling and reasoning in the following. In the case of a smart home, the context “\(C\)” consists of the Environment, Time, Subject, and Activity. Temperature data, air quality data, humidity data, and so on are the data in its data space, and the context corresponding to these data are the home environment context model, reflecting real-time environmental information of smart home.

After collecting, preprocessing, and inferring the context information, the corresponding high-level context is obtained, which can be used as the basis for the
system service call. Therefore, the mapping from the context to the service is shown in the following formula (3)

$$S = M_{C \rightarrow S}(C)$$  \tag{3}$$

C is a set of context, S is a callable service set, and M is a mapping from the context-aware information model to the service. There are two types of mapping methods for acquiring contexts to services, which are rule-based methods and machine learning-based methods. Rule-based method has high accuracy but weak generalization ability. Machine learning methods have high generalization ability but require large initial training data and numerous computing resources. Due to the limited computing resources of the IoT and limited numbers of data in the home area, this article uses a rule-based inference method to realize the mapping of contexts to services. After the advanced context information is transmitted as input to the inference engine, the business service identifier to be called is generated as output through the rules. Then the system calls the controller to complete the service invoke.

The rule configuration model is identified by a triple

$$\text{conf} = \langle \text{confID}, \text{CXT} - V, \text{BS} \rangle$$

$$\text{CXT} - V = \{ \langle \text{ext}_1, v_1 \rangle, \langle \text{ext}_2, v_2 \rangle, \cdots \langle \text{ext}_n, v_n \rangle \}$$  \tag{4}$$

ConfID identity this configuration rule, CXT – V is a set of context information-value pairs, and BS is a business service, for example, remote control services for smart electrical equipment. IoT devices are mainly divided into three categories based on their capabilities: perception devices, control devices, and IoT gateways with communication capabilities. The main function of the sensing device is to collect the data of the surrounding environment. Therefore, the virtualization elements of the sensing device include characteristics of the sensing device collecting data, such as index name, index value, acquisition time, and data unit. The control devices are used to receive and forward instructions, including instruction data, instruction execution results, and behavior operation times.

**System framework**

According to the cycle of the context, this article designs an intelligent control framework under the smart home scene. System framework is divided into four layers: the sensor layer, the middleware layer, the control decision layer, and the service layer, as shown in Figure 1.

The bottom layer is sensor layer, which is used to collect IoT data. The collection of context information mainly includes various sensible hardware of smart home environment. In addition to physical devices, perception layer also contains virtual sensors, which obtain the required data from other data sources, such as network services that provide weather information.

The middleware layer is used to solve the heterogeneity of the bottom device. Due to the lack of communication standards for IoT devices, the communication protocols of the IoT devices vary from different manufacturers. Therefore, the middleware of IoT is used to parse the protocol of the data which sensors collect and convert it into a unified data structure that the upper layer can handle.

The control decision layer is the core of the system framework. It processes the data transmitted by the underlying device to obtain the call strategy of the corresponding service. This layer can be regarded as a context awareness engine, which includes a data access interface (DAI), a poller, an inference engine, and a call controller. Although the middleware assists the system to identify the data transmitted by each registered device, the context information sources of corresponding data are quite different. To further classify the data for the inference engine, DAI and poller work together to make the input data conform to the standards of the context-aware information model. Different context-aware modeling methods have different characteristics, and the appropriate context modeling method should be selected according to the business. Similarly, the inference engine has different options depending on the modeling method, but the purpose of the inference engine is to search for business services that match the current context, and then complete the service call by calling the controller.
The service layer contains various business services provided to users. It can be a business service packaged as a WebService, or it can be an API interface for remote control service provided by various smart devices.

**Context-aware modeling**

The purpose of context-aware modeling is to map the data to the corresponding interactive context, so that the control system can further analyze and process scenarios and provide support for future service decisions.

**Existing modeling approach**

Context modeling approaches are mainly divided into three categories: key-value-based modeling, object-based modeling, and ontology modeling.

The modeling approach–based key-value expresses different scenario by storing data from interactive scene as key-value pairs, as shown in the following formula (5).

\[
\text{context} = \{ (\text{key}_1 : \text{value}) \cdots (\text{key}_n : \text{value}) \} \quad (5)
\]

This kind of approach extracts key index categories in interactive tasks as keys, and combines these keys and their values at a certain moment or a certain time period to express the corresponding scenario.

Object-based modeling approach is developed on the basis of object-oriented programming idea, which generally abstracts contexts into the following forms

\[
\text{context} = \left\{ \begin{array}{l}
\text{Class}_1(\text{object}_1, \cdots, \text{object}_n) \\
\text{Class}_2(\text{object}_1, \cdots, \text{object}_n) \\
\text{Class}_n(\text{object}_1, \cdots, \text{object}_n)
\end{array} \right\} \quad (6)
\]

By abstracting the key entities involved in the interactive scene into corresponding categories, we build corresponding properties and methods for different categories. The current context awareness can be expressed based on the number of entities in the scenario and the attribute methods within the entity. However, its ability to express relationships between entity categories is weak, and only a few built-in relationships are available.

Ontology-based modeling is the current mainstream modeling method, which is similar to object-oriented method. By abstracting the key concepts in the interactive scenario and defining the relationship between the concepts based on the actual business, its expressive ability is better than other two approaches. The basic composition of the modeling results is shown in the formula (7):

\[
\text{Subject} \xrightarrow{\text{Predicate}} \text{Object} \quad (7)
\]

This formula is called *SPO triplet*. Subject is a category or entity. Correspondingly, Object is the corresponding category or the attribute value of the entity. Predicate is the relationship or attribute name between categories. Through this basic element, each concept and the relationship diagrams among entities are formed to express the corresponding scenario. However, the ontology method also has certain defects, which have high requirements for hardware devices and poor execution performance. It is not suitable for the real-time control of IoT.

Different approaches have great differences in expression ability, performance, and complexity. So, the most suitable context-aware modeling should be selected for different business scenario. This article proposes a new modeling method for the home environment, which is more suitable for the smart home than the mainstream approach.

**Object-oriented entity abstraction**

At this stage, the sensing devices in the field of smart homes are still dominated by various sensors, and the perceived native scenario information is transmitted to the server through the IoT gateway or other types of network relays. The native scenario information is converted into corresponding high-level context information on the server based on the context model library, so that it can be used as input to the inference engine, and finally obtains the required operations and calls related services.

Through the analysis and comparison of existing context modeling approaches, this article explores a new modeling method suitable for the field of smart home. This article introduces the attribute graph model with graph database technology to make up for the shortcomings of object-oriented method expression. The new modeling approach retains the high portability and high performance of the object-oriented method. This method is more suitable for the reality that the IoT home resources are limited.

Based on the characteristics of the home interactive environment and the analysis of the interactive tasks, the key elements involved in each interactive scene of the home environment are extracted and abstracted into corresponding entity classes as components of the home interactive context model. In the home environment, the physical space is divided into multiple subspaces, and the user’s activities at home are limited to these subspaces. The subspaces provide users with different services, and smart devices are installed and deployed in different subspaces. The subspace uses these smart devices as carriers to provide services to users. It can be found that smart devices are the key factors for intelligent control of the home environment. Smart devices are divided into perceptive devices and controllable
devices. A house space can be abstracted into six physical categories: home space, room (divided from home space), users, sensor object, smart object, and control object, as shown in Figure 2. The entity is the object of the corresponding category, such as the master bedroom object of the room category. The home space can interact with each other as smart space objects after deploying the intelligent control system.

The home space class represents the environment in which the device monitoring and interactive services are provided. It is defined as follows:

\[
\text{Space} = \{\text{space}\_\text{id}, \text{owner}, \text{address}\}\tag{8}
\]

The \text{space} class contains the user to which the living space belongs, the latitude and longitude of the physical location, and the house address.

The \text{room} class is the physical division of the home space and the division of the usage of each subspace. It is used to depict the physical location of each specific interaction activity. In addition to the real room in the home, it also includes rooms with no specific functions such as aisles and corridors in the home, so it can comprehensively cover all home locations. It is represented in the following formula:

\[
\text{Room} = \{\text{room}\_\text{id}, \text{owner}, \text{tag}\}\tag{9}
\]

The label of the room class marks the category to which it belongs. For example, there can be more than one bedroom for living in a commercial house, while their labels are both bedrooms, which is further distinguished by the ID identifier. The attribute of the belonging user is to provide more personalized service, for example, a certain device in a certain room can only be used by the owner.

The \text{user} class describes the user’s identity, the preference, and corresponding permissions so as to provide support for the user’s personalized service. It is represented as follows:

\[
\text{User} = \{\text{user}\_\text{id}, \text{age}, \text{gender}, \text{role}, \text{preference}\}\tag{10}
\]

Attributes such as age and gender are used to provide the basis for future recommendation services. The role attribute identifies the user’s role in the system and assigns the corresponding authority. The preference attribute refers to the users’ preferences, such as water temperature, room temperature, air humidity, and brightness in home life.

The \text{sensor} class includes all sensing devices under the IoT, ignoring the difference in structure and function. It is used to perceive context information, including parameter information corresponding to the surrounding environment, device status, user position, and so on. Sensor class is represented as formula (11). The target identifies the category of indicators to be monitored.

\[
\text{Sensor} = \{\text{sensor}\_\text{id}, \text{target}, \text{value}\}\tag{11}
\]

The \text{control} class refers to smart devices that can be controlled remotely through the network or other methods, such as smart door locks, smart refrigerators, and smart switches. It is defined as follows:
Device = \{id, Action_1(para_1, \ldots, para_n),
              Action_2(para_1, \ldots, para_n),
              \ldots,
              Action_n(para_1, \ldots, para_n)\} \quad (12)

The controllable device class contains all the actions it provides, and the para refers to the parameters corresponding to each operation.

The above results of object-based modeling are not ideal, because its ability to describe the relationship between various entities is weak. The built-in relationship of the object-oriented method is not enough to describe interactive tasks, such as the affiliation between the device and the room.

**Attribute graph model of IConS**

Based on object-oriented entity and graph database, an attribute graph model is proposed to describe the relationship of objects, which makes the context model more complete and faster than traditional databases in performance.

The attribute graph model is a directed graph composed of nodes, edges, labels, relationship types, and attributes. Edge is also called relationship. The entity classes abstracted above can be used as the label attributes of nodes. The nodes having the same label belong to the same type, and the instance objects correspond to the nodes in the attribute graph model. Different nodes have different attributes. The attribute graph model provides the definition of the relationship between nodes. The relationship is pointed from the start node to the end node. The advantage is that the relationship can define attributes for it, making the description of the relationship more diversified and the expression ability stronger.

As shown in Figure 3, based on the six types of entities obtained with the object-based method, the association relationship between them is analyzed in the home scene and described using the attribute graph model.

It can be seen from the figure that the user owns the house (User-[HAS HOUSE]-> House), and the reverse relationship is that the house belongs to the user. The house is composed of rooms (House-[COMPOSED_OF] Room), otherwise it is defined that the room is located in the house (Room-[LOCATE_IN]-> House). Different sensing devices and controllable devices are deployed in different rooms (Room-[HAS SENSOR]-> Sensor), (Room-[HAS DEVICE]-> Device), otherwise the devices are located in different rooms (Sensor-[LOCATE_IN]-> Room), (Device-[LOCATE_IN]-> Room). In addition to describing the relationship between each entity category, the attribute graph model can also provide attribute definitions for the defined relationships. As shown in the relationship between houses in the figure, the orientation attribute and distance attribute between two houses are defined. House-[LOCATION

---

**Figure 3.** Home entity relationship diagram.
When the controllable device and the sensing device are located at the room, their attributes also define the position of the device in the room, such as air conditioning on the wall and temperature sensor on the ceiling. This is also the advantage of attribute graph model over ontology modeling method.

Figure 3 shows the correlation with the entity category as the granularity unit. Different entity objects that fall into the same category can also define relationships for them, which provide an interface for future access to external systems such as social networks. Different users have different relationships. Therefore, the Figure 4 shows an attribute model diagram of a house with two bedrooms, one bathroom, and a living room, in which each node represents one physical object. For viewing convenience, only the basic associations of each entity object are identified in Figure 3.

The relevant information of the home environment can be clearly understood from Figure 4. After modeling the home scene in the above method, entity objects can be flexibly added or removed based on the graph model. It provides a basis for future expansion such as recommendation services.

Smart home modeling results

The IConS model combines the advantages of the attribute graph model with the object-oriented method, which enables the original data perceived by sensors to generate object instances of the appropriate type of entities quickly after preprocessing. However, the system is not sufficient to express a specific home interaction scenario, such as a scenario where the user is sleeping. Therefore, by referring to the top-down design ideas of the ontology and the multi-agent architecture model of the context-aware system, the environment, behavior, and subject model are put forward to fully describe the current users’ situation.

The following formula gives a formal expression of the home context

\[
\text{Context} = (\text{Environment} , \text{Time} , \text{Subject} , \text{Activity})
\]

The highest abstraction level of contexts includes the context where the interaction is, the current time, the subject involved, and the behavior of the subject.

Environment = \{Location \parallel \text{Weather} \parallel \cdots \text{(Sensor Type)}\}

Location = \{\text{Space}, \text{Room}, \text{Position}\}

The context model includes the location of the interaction, the weather where the interaction occurred, and all sensor indicators that can sense environmental factors, such as ambient temperature, humidity, and air quality. The location is identified by the address of the

\[
\text{(direction: “north,” distance: “10 km”)} \rightarrow \text{House}
\]
house in the database and the space-room-location hierarchy.

The \textit{time} model \{\texttt{DateTime}, \texttt{Label}\} is divided into two parts: \texttt{DateTime} format value and label. The absolute value of \texttt{DateTime} can be converted into business-related semantic information through the label, such as season, morning, evening, and working day. By adding semantic label information related to the interaction, future adaptive control is more precise

\begin{equation}
\text{Subject} = \{\text{User} \mid | \text{Device} \mid \text{Environment}\}
\end{equation}

The \textit{subject} model identifies the main objects in the current interactive activity. In most cases, it contains user objects. In a small number of cases, there are also device objects and environment objects. For example, a device failure will cause the system to push messages to the user's mobile phone.

\begin{equation}
\text{Activity} = \\
\{\text{UserActivity} \mid | \text{DeviceAction} \mid \text{EnvironmentAction}\}
\end{equation}

The \textit{activity} model corresponds to the subject model, and its value range will change correspondingly with the subject model. For example, user behavior includes a series of home activities such as getting up, sleeping, reading, leaving home, and returning home. Device operations include the unique behavior of the smart devices, such as the cooling of the smart air conditioner, the wind swing, and the common opening and closing of all devices. Environmental actions include natural factors such as rain, thunder, strong winds, and low air quality.

At this point, the native data are transformed into a unified structure of high-level context.

\section*{Context-base fusion control technique}

Intelligent control means that the system automatically takes appropriate control actions to meet the users' expectation according to the real-time situation without the users' participation. Therefore, the system needs to map the context to the service. There are two ways to get this mapping which are rule-based approach and machine-based approach. Rule-based approach has high accuracy rate, but is weak in generalization. Machine learning approach has high generalization ability, but requires an initial training data set as well as greater computing power. Considering the higher accuracy and semantic comprehensibility of rule-based system, it is still the preferred intelligent control method in the home scene.

Rule-based approach stores the business rules separately from the main processing code, so that the control system can change the business rules without changing the code, which greatly enhances the scalability of the system. When we build a rule-based control system, a rule base should first be built and initialized based on context-aware objects. By combining different contextual objects with their logical relationships on the condition side and the result side, the rules that meet business requirements are defined. When the context data are processed into the corresponding context object model, it is passed as the fact to the rule inference engine. The inference engine can be understood as a tool that performs the mapping of contextual information to control services.

\section*{Rule-based control}

The rule-based approach enables the system to obtain corresponding control decisions based on the situation, and the control decision is composed of atomic control service of devices. Through the interface of the most fine-grained device control services, the intelligent control system can call the control services provided by any intelligent device, and combine the finest-grained control services according to business logic.

Atomic control is to control the specific smart device. Different smart devices provide different services, so they need to be abstracted and encapsulated into corresponding service interfaces. After investigating the current mainstream technologies, considering the robustness of the system and future scalability, the popular micro-service architecture has been selected. This article adopts the RESTful architectural style, which treats everything on the network as a resource. Each resource has a unique identifier, and the operation of the resource will not change the identifier. The operations corresponding to the device are encapsulated as the addition, deletion, and modification of the corresponding resource identifier. Household equipment is divided into two categories: perception equipment and control equipment. The design principle of RESTful interfaces is to query the current data of a sensor device, pass in parameters, and call the corresponding operation functions of the control device. For example, the operation of the air conditioner can be encapsulated into multiple operation functions.

Based on atomic-level control, business logic rules are constructed by using the state of context models as the conditional part and the atomic-level control behavior as a constituent unit of the result part. Rules can be abstracted as shown in the following formula:

\begin{equation}
\text{Rule} = (\text{name}, \text{priority}, \text{condition}, \text{result})
\end{equation}

Each rule contains the identifier name, the priority value used for conflict resolution, condition, and the result. The conditional and resulting parts of a rule are made up of a scenario model library as a component
and its logical relationship. The structure of the condition part is as follows:

\[
\text{Condition} = \left\{ \begin{array}{l}
\text{model}_\text{ctx}_\text{1}(\text{attri} : \text{val}) \cdot \cdot \\
\text{&} \cdot \text{model}_\text{ctx}_\text{2}(\text{attri} < \text{val}) \\
\text{&} \cdot \text{model}_\text{ctx}_\text{3}(\text{attri} > \text{val})
\end{array} \right. \quad (18)
\]

The condition part contains the atomic conditions which are composed of the judgment of the range of attribute values of context models. These atomic conditions are combined by logical relation symbols to form the complete condition part of the rule. For example, the temperature attribute value of the environmental context model is greater than 26 degrees Celsius, and the corresponding value of the time context model is 11 o’clock in the evening, and the location attribute value of the user context model is the bedroom. These three atomic conditions together constitute the condition part of the user’s sleep rule.

The structure of the result part is as follows:

\[
\text{Result} = \left\{ \begin{array}{l}
\text{model}_\text{ctx}_\text{1}(\text{attri} : \text{val}) \cdot \cdot \\
\text{&} \cdot \text{model}_\text{ctx}_\text{2}(\text{attri} : \text{val}) \\
\text{&} \cdot \text{model}_\text{ctx}_\text{3}(\text{attri} : \text{val})
\end{array} \right. \quad (19)
\]

The structure of the result side is similar to that of the condition side. It is a combination of atomic operations that set the state of the corresponding device through the context model.

An example of a security rule is shown below.

```
Rule gas
  Salence 10
  When
    $gasSen: \text{gasSensor(densityVal > 36)}
    $window: \text{Window(address==’kitchen’)}
  Then
    $window.open()
```

**Mode-based control**

Various situations in the home scenario lead to a large number of rules. The granularity of the rule-based control method is still somewhat small, and in the home scene users often need a series of actions to achieve a certain purpose or create a certain environmental atmosphere. Therefore, the mode control method with higher granularity than rule control is proposed. According to user preferences and usage habits, mode-based control combines the behaviors of multiple rule actions, and the condition part is composed of the above context models and the user-defined preference library, such as work mode and sleep mode. The work mode includes wake-up service, on-time broadcast, automatic opening of breakfast toaster and coffee machine. There is a nesting property between the modes. For example, the romantic mode of the light will also trigger when using the user’s entertainment mode. This nested relationship increases reusability and reduces coupling between workload and modes.

The expression of mode control is shown as follows.

\[
\text{Mode} = \left\{ \begin{array}{l}
\text{(Context}_\text{model}_\text{1}, \text{attribute}_\text{i} : \text{val}) \\
\Rightarrow \cdot \text{(atomAction}_\text{1}, \cdot \cdot \cdot , \text{atomAction}_\text{n})
\end{array} \right. \quad (20)
\]

The structure of the mode control includes the conditions for triggering the mode control and the operations on the result side. The following gives examples of reading mode.

```
Mode “readMode”
  When
    $user: \text{User(location==’study’)}
    $study: \text{Room(flag==’study’&&duration > 5)}
    $tempSen: \text{tempSensor(address==’study’)}
    $lightSen: \text{lightSensor(address==’study’)}
  then
    if ($lightSen.val < $user.LowLightPreference)
      $study.openLight(’read’)
    if ($tempSen.val > $user.HighTempPreference)
      $study.openAirconditioner(’cold’)
      $study.closeAllWindows()
```

This mode sets the temperature and light requirements of the user when reading, and adjusts the study room environment according to the user’s preferences.

**Voice-based control**

According to the business characteristics in the home field, this article uses the speech control technology. Figure 5 shows the voice-based control process.

The process of voice-based control has these steps:

1. A specific wake-up word wakes up the voice control module. If the correct wake-up word is recognized, go to step two, otherwise refuse service.
2. The system converts speech to text data and identifies the user’s identity.
3. According to the dictionary customized for the home scene, the text data are segmented.
4. The system extracts information from the word segmentation and convert it into a unified data structure UDS (Uniform Data Structure).
5. The system determines the type of voice command according to UDS. If it is the rule setting, write it into the rule base. If it is a control
command, the corresponding control service is called.

To protect user privacy, when the microphone detects voice input, it decides whether to include the wake word (such as Nico, the wake-up word can be customized). If a wake-up word is included, the next step of analysis is performed, otherwise it is ignored.

For user identification, we can utilize the current mature voiceprint recognition technology and configure by using third-party tools to achieve a higher level of user recognition efficiency. For voice-to-text, considering the resource-constrained devices and the stability of the network, a third-party offline jar package is chosen to convert the voice file to a text file. The household vocabulary is updated less frequently.

By extracting commonly used control sentences in the home field and related devices, sensors, user perception attributes, and other nouns, a home scene dictionary is constructed. Therefore, according to the semantic dictionary of smart home, the system can split words from the text file. The advantage of dictionary-based word splitting is accurate, which is very much needed in the field of smart home. Dictionary making is not as complicated as other fields.

After the word is split, the information is extracted and the extracted information is constructed into a unified data structure. This forms a voice control command. The following formula gives the structure of the control command.

\[
\text{Control}_{DS} = \text{User} + \text{Time} + \text{Device} + \text{Action} + \text{Supplement}
\]  

\[
\text{User} \text{ identifies the user’s identity and is used to determine whether it has the authority to operate related equipment. At the same time, it can be associated with the user preference database to facilitate the supplement of missing information (implicit information) so that device control can be more precise and personalized.}
\]

\[
\text{Time} \text{ identifies the time when the control instruction is executed.}
\]

\[
\text{Device} \text{ identifies the device and action identifies the operation. They are interdependent, because the operations vary from device to device. For example, action of air conditioning will involve modes (cooling and heating), specific temperature values, air volume, and wind direction. But the lights may only have two operations, on and off. Action can be understood as the API calling method corresponding to the device, and the information converted by the user’s voice is often not enough to accurately locate the corresponding API. Therefore, system inference is required to obtain relevant supplementary information, which is called implicit information extraction.}
\]

\[
\text{Supplement} \text{ is used to supplement information. The extracted hidden information includes complete information of control command and supplemental information for future users’ habit mining and service recommendations. For example, in order to locate the device to be operated, you need not only the device name included in the user’s voice “turn on the air conditioner.” It is also necessary to reason about the air conditioning entity of the specific room according to the user’s location. It infers that the air conditioner will be set to refrigeration according to the temperature information. As for the setting of parameters such as temperature value and wind direction, it can be completed by searching user preference information. The voice control used to set the rules will be converted into the following structure.}
\]

\[
\text{Rule}_{DS} = \text{User} + \{\text{Condition}, \text{Result}\}
\]

\[
\text{Condition} = \{\left(\text{type, object, attr, val}\right), \ldots\}
\]

\[
\text{Result} = \{\left(\text{device, action, para}\right), \ldots\}
\]

This structure is based on the rule structure in the inference engine of this system.

\section*{Intelligent control process}

The rule structure in the system reasoning engine contains both conditions and results. For the description of the conditions, a four tuple (type, object, attribute, value) is used to describe the atomic conditions.
corresponding to entity categories, specific objects, specific properties, and values, based on the context modeling method of the object-oriented and attribute graph model in the previous section. The combination of atomic conditions constitutes a regular condition. The result is based on the atomic-level control method, using (device, action, parameter) triples, through specifying the device identification, device packaging services, and service parameters to complete the atomic-level control function. Finally, it is combined into a regular result.

Home control should be divided into two ways to reflect personalization and intelligence. One way is to automatically control without user participation. It predicts user needs based on changes in the situation and completes device control independently. The other way is the user’s voice control. When the user does not accurately express his purpose, the user’s command is supplemented by extracting the implicit information of the current situation to complete the device control.

The automatic control method is mainly realized by setting the rules, mode control, and preferences of certain indicators in advance. For rule settings, scholars often set the user’s identity as part of a rule condition. This method traverses all the rules into the rule engine after the context changes. In most cases, however, it only needs to traverse and match the rules that correspond to the users involved. The system binds users to rule files through one-to-one relationship so that a small number of rules need to be traversed when the context changes. User preference refers to the definition of the range of comfort indicators in the home life over a period of time, such as temperature, humidity, and water temperature. In this article, the concept of user preference is introduced to the rule file to reduce the coupling of the system. The system has the flexibility to change policies as the user’s situation changes.

As for voice control, a unified structure control command is generated on the cloud server side by matching the text and home dictionary. For the missing user, time, specific room device, and operating parameter information in the voice control command, the system performs implicit information extraction to improve the intelligence of the system. The control instructions are fully supplemented by the extraction of implicit information. However, if voice commands cannot be supplemented by implicit information extraction, the user needs to change his words to balance the accuracy of the user intention recognition and user experience.

**Prototype system and experiment**

**Prototype system**

The prototype system is developed by JAVA and is divided into five modules. As shown in Figure 6, from bottom up, the architecture of prototype system includes device management, middleware, data preprocessing, intelligent control and user interaction. After analyzing the functions provided by the existing smart home system, IConS supports two ways of control. The first approach is the user’s active control, which the user manually controls the phone by clicking in the APP, or by using the microphone of the mobile phone to input commands. The second approach to control is to automatically control the system without user involvement. This approach processes the context data perceived by the sensing device into high-level context information and inputs it into the inference engine to match the rules in the rule base. Finally, the processing result is obtained, and the appropriate service is invoked.

In this prototype system, raw data perceived by each sensor is filtered through the middleware filter and passed to the background. The background constructs the context object by using the data from each data source as the attribute of the relevant context model. At this point, the newly generated contextual model object is compared with the current system cache. If the attribute changes, it proves that the current scene has changed from the previous one. The newly generated context object replaces the current cache and acts as a fact in the Drools rule engine, which forms a
conditional network based on the Rete algorithm by decomposing the condition part of the rules. When facts are passed as input nodes, the algorithm can output the rules that meet the criteria. Thereby, the control commands are obtained that need to be executed in the current scenario.

The control flow of the prototype system is shown in Figure 7. For automatic control, the perception data from heterogeneous device is parsed first, such as the sensor data from the Ruff board (IoT development board). The second step is to compare the sensor data with the attribute data of each object in the system cache. If the data does not change, the process terminates, otherwise it goes to third step: mapping the changed data to the object. The system changes the object properties in the cache and transfers them to the reasoning engine as a new fact. The fourth step is to match the facts with the rule base. If there is no match, no action is triggered. Otherwise, it goes to fifth step and selects execution from a set of rules that meet the priority criteria. The sixth step is to convert the result of the rule execution to the appropriate service call. Finally, the appropriate smart device performs action according to the control command.

For proactive control, the user logs in the system through the APP authentication, selects the bound house, and displays the index data of the house and the status of each controllable device according to the user authorization. Users can directly click the interface button to complete the device control, or use the microphone to input voice through the interface. APP performs the text conversion operation on the machine and transfers the text to the cloud server. The server compares the text with the home dictionary to extract the keywords required by the context model, namely, users, devices, operation, and parameters. If the control information is complete, the system directly generates the execution commands, otherwise the system obtains the full control commands based on implicit information (e.g. current time and temperature). After the final authentication operation is completed, the appropriate service is invoked to complete the control.

The prototype system uses Drools rule engine to match the rules between input data and control instructions. Drools is an open source rules engine based on Charles Forgy’s Rete algorithm. However, most IoT manufacturers in China do not provide open API interfaces for their devices, we use simulated data to simulate sensor data in the home.

The following are two examples of personalization rules about light intensity, corresponding to two users, A and B, with different sensitivity to light intensity and the different priority of rules. The business of the two rules is similar, but their priorities and user preferences are different in sensitivity to light intensity. When the system detects that users A and B enter room c, the location attributes of user objects A and B in the system memory change, that is, a new fact is generated, which drives the rule engine to be called again to determine whether the rule is triggered. The system passes the attribute of the graph model which can search the list of devices connected to the room c node, and the sensing device can perceive the various parameters of room c, and pass it into the inference engine to compare with the condition end of the rule. If the conditions of the rule are all satisfied, enter the conflict set and wait processing, when all the condition nodes are traversed, if there is only one rule to be executed in the conflict set, the function on the result side is directly called, otherwise the conflict resolution process is executed. And the light in room c is lower than the lowest brightness that user A can tolerate. At this time, the light in room c should be opened for user B, the brightness of room c is greater than the maximum brightness tolerated, and the light in room c should be turned off. Therefore, there are two rules that meet the conditions in the conflict set, but because user A’s rule has a higher priority, the light in room c is turned on. When user A leaves room c, the rule engine restarts, so that user B’s rules are executed and the light in room c is turned off.
Different from the mainstream ontology modeling, IConS adopts the modeling methods based on object and attribute graph. This section evaluates the new modeling approach by testing the performance of the system and comparing it with the ontology modeling approach.

The experiment used a computer with an Intel’s 5.3 GHz CPU and 4G memory. The control variables of the experiment are the rules in the rule base and the number of household devices, including sensing devices and controllable devices. Figure 8 shows the results of performance tests and comparative experiments with simulated sensors data. The (a), (c), and (e) graphs of the figure are the system performance tests, which uses the IConS method of object and attribute graph. The other graphs of the figure are the system performance tests based on the ontology modeling method. As can be seen from the experimental results in the figure, with the increase in the number of rules, the memory utilization of both modeling methods has been significantly improved. Because the reasoning engine used is based on the Rete algorithm, which stores the intermediate state of the rule conditions in memory, it increases exponentially with the increase of the number of rules. The response time of both systems increased, but the response time of the ontology modeling method system is around 10 s, while the IConS method in this article is around 1 s. Considering the user experience of the home application scenario, this new modeling method is more suitable for this scenario. As users spend more time with the system, the number of rules is bound to increase. The system constructed based on the proposed modeling can still provide stable services.
The prototype system implements personalized services by binding users, rule files, user preferences. The system transmits the rule files matching the user identity to the inference engine, and the condition side of each rule needs to extract the user preference data, so that it can provide personalized services according to the user identity and user preferences. This system uses Neo4j graph database for storage, thereby improving the search efficiency.

The experiment also compares the search efficiency of MySQL traditional relational database and Neo4j graph database on user rules and preferences by increasing the number of users, user rules, and user preferences. The experimental results are shown in Figure 9.

The experimental results from the Figure 9 show that when the number of users, rules, and corresponding preferences of a single user are at a low level, the query efficiency of the graph database and the relational database is almost the same, but with the increase of these three indicators, the query efficiency of MySQL database has been significantly reduced, but Neo4j has remained stable at a lower latency level. Especially when the number of users increases, the difference of performance is huge. Because as the number of users increases, the number of rules and preferences will double, so the total number of searches increase exponentially.

**Conclusion**

This article proposes a context modeling method of smart home based on object and attribute graph model, called IConS. After creating the context-aware models with this method, devices controls and personalized services can be integrated into the smart home by defining the interaction rules. An intelligent control scheme based on rule reasoning is proposed. The system provides a multi-granularity, multi-mode control method, and has a high degree of intelligence.

However, most IoT manufacturers in China do not provide open API interfaces for their devices; hence, the performance experiments cannot be carried out widely in real scenarios. The experiment results with simulated data show that the IConS model has better performance, higher scores, less resources cost, and shorter response time than the baseline model based on ontology modeling.

The rule engine maps the contexts to the appropriate control services through the rules to achieve automatic control of the system. The context acts as the conditional part of the rule. Web services abstracted from smart devices serve as the atomic components of the result part. Custom rules are added based on the interaction needs of smart home. Thereby, the intelligent control process from environment awareness to target service call is realized. By adding mode control and voice control, the interactive control of the system is further enriched. The strategy of implicit information extraction is added to the voice control to improve the intelligence of the system.

This article provides some ideas for further research and we must admit that there are inevitably shortcomings in the research work, which need to be further improved in the future work. In the process of
prototype system building and performance testing, the number of physical devices used is small-scale simulation data, while in real scenarios, the number of devices is large.

Because the system is limited by business rules and context modeling, mainly from the user’s active definition, the system’s generalization ability is weak. IConS context modeling and personalized rule base requires a lot of work to manually define objects, attribute relation graph and rules, which limits the application of this method in other intelligent environments. In the future, potential rule mining will be performed on the historical records of system rule execution to make up for this defect.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This article is based on a research project supported by National Natural Science Foundation of China (Grant No.61671157) and National Key Research and Development Project (Grant No. 2018YFB1703205).

ORCID iD
Minbo Li https://orcid.org/0000-0003-1072-7652

References
1. Sun Q, Liu J, Li S, et al. IoT: summarize on concepts, architecture and key technology problem. J Beijing Univ Post Telecom 2010; 3(3): 1–9.
2. Suryadevara NK and Mukhopadhyay SC. Smart home related research. In: Suryadevara NK and Mukhopadhyay SC (eds) Smart homes. Cham: Springer, 2015. 11–51.
3. Schilit BN and Theimer MM. Disseminating active map information to mobile hosts. IEEE Netw 1994; 8(5): 22–32.
4. Sun JZ and Sauvola J. Towards a conceptual model for context-aware adaptive services. In: Proceedings of the fourth international conference on parallel and distributed computing, applications and technologies. Chengdu, China, 29 August 2003, 90–94. New York: IEEE.
5. Strang T and Linnhoff-Popien C. A context modeling survey. In: First international workshop on advanced context modelling (Ubicomp), 2004, https://elib.dlr.de/7444/1/Ubicomp2004ContextWSCameraReadyVersion.pdf
6. Gellersen HW, Schmidt A and Beigl M. Multi-sensor context-awareness in mobile devices and smart artifacts. Mobile Netw Appl 2002; 7(5): 341–351.
7. Mo T, Li WP, Wu ZH, et al. Framework of context-aware based service system. Chin J Comp 2010; 33(11): 2084–2092.
8. Bonino D and Corno F. Dogont-ontology modeling for intelligent domotic environments. In: Sheth A (ed.) International semantic Web conference. Berlin, Heidelberg: Springer, 2008. 790–803.
9. Kishore Ramakrishnan A, Preuveneers D and Berbers Y. Enabling self-learning in dynamic and open IoT environments. Proc Comp Sci 2014; 32: 207–214.
10. Chen Z. Mining individual behavior pattern based on significant locations and spatial trajectories. In: Proceedings of the 2012 IEEE international conference on pervasive computing and communications workshops, Lugano, 19–23 March 2012, 540–541. New York: IEEE.
11. Berkane ML, Seinturier L and Boufaida M. Using variability modeling and design patterns for self-adaptive system engineering: application to smart-home. Int J Web Eng Technol 2015; 10(1): 65–93.
12. Uddin I. A rule-based framework for developing context-aware systems for smart spaces. Nottingham: University of Nottingham, 2019.
13. Dinh DL, Kim JT and Kim TS. Hand gesture recognition and interface via a depth imaging sensor for smart home appliances. Energ Proc 2014; 62(62): 576–582.
14. Vacher M, Lecouteux B, Aman F, et al. Recognition of distress calls in distant speech setting: a preliminary experiment in a smart home. In: Proceedings of the SLPAT 2015: 6th workshop on speech and language processing for assistive technologies, Dresden, 23 September 2015, 124–129. Stroudsburg, PA: Association for Computational Linguistics.
15. Mofrad MH and Mosse D. Speech recognition and voice separation for the IoT. In: Proceedings of the 8th international conference on the IoT, Santa Barbara, CA, 15–18 October 2018, 1–8. New York: IEEE.
16. Drool Java-based powerful open source rule engine [EB/OL].http://www.drools.org.cn/