Feature Interactions Detection and Resolution in Smart Homes Systems

A. S. Alfakeeh and A. H. Al-Bayatti
STRL, De Montfort University, Leicester, UK
Email: asalfakeeh@kau.edu.sa, alihmohd@dmu.ac.uk

***Abstract*—Smart homes systems (SHS) have become an increasingly important technology in modern life. Typical smart home systems comprise several services and appliances which are deployed in a dynamic environment serving multiple residents. Services may include home security, power management, HVAC (heating, ventilating, and air-conditioning) or entertainment. Each service may include several features which have to be carried out in order for the service to function according to the house residents' preferences. Each feature is a task to control a single device. In SHS most of devices such as lights, windows and AC's are shared by several services. What is more, residents vary in their preferences of the services which are provided to them like the preferred room temperature or even the favorite TV show. Such complexity leads sometimes to undesirable interactions between services within the SHS. That is because two or more services which include conflicted features work simultaneously. The interaction occurs when these services have conflicting features or the services work according to conflicted residents' preferences. The outcome could affect the functionality of the overall system. Several studies have targeted feature interactions detection and resolution in smart homes systems. However, none of these approaches adopts a mechanism of negotiation between such services or residents' preferences to reach a compromise enabling them to work at the same time. Therefore, this study proposed a flexible Agent-Based Negotiation System (ABNS) of feature interactions detection and resolution in SHS.**

***Index Terms*—feature interactions, smart homes systems, negotiation, agents**

I. Introduction

Smart Homes Systems (SHS) are an important technology in today's life. Apart from safety, security, convenience and entertainment, they offer significant benefits for the elderly, disabled and others who find it difficult to live independently [1]. What is more SHS are environmentally friendly through carefully managing energy consumption [2].

SHS enable homes to sense their environment and to be aware of their resident's preferences and provide services accordingly. Resident preferences (which is known as a user policy) include any desire or preferable performance in every aspect of home automation [3].

To close curtains, dim light or to follow certain order when security service is triggered are all examples of the preferences of the house residents. In literature, the definition of smart home varies due to variation of services which to be provided by the SHS. A general definition of smart home is "a residence equipped with computing and information technology, which anticipates and responds to the needs of the occupants, working to promote their comfort, convenience, security and entertainment through the management of technology within the home and connections to the world beyond" [4].

SHS are a result of the need to take what were once a number of separate devices designed to improve the home and integrate them into a distributed system, so that these devices can work together to provide services to residents. Such integration of services provides an intelligent response to resident preferences.

SHS mainly consists of the following general services: security, safety, power management, HVAC, lighting, home appliances, healthcare, communication support and entertainment [5]. SHS could include all the above services or some of them based on the user (resident) need. Each of these services may include other services referred as subservice, for example, security in addition to its apparent function in monitoring suspicious activities, may include access control or remote surveillance [6]. Another example is a power management service where its main function is to reduce energy consumption, and an additional subservice could include a 'smart meter' function which automatically send energy meters readings to the energy provider and manage the home appliances, thereby stagger the peak period of electricity load in order to reduce the power load of peak period to save energy [7].

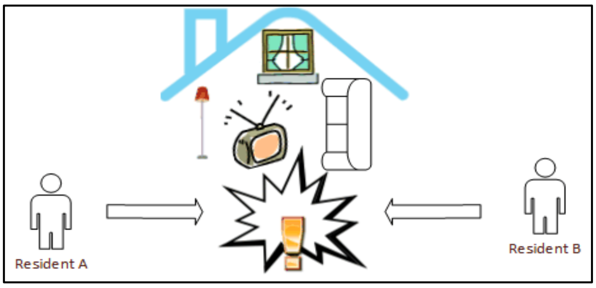
Because there is a multitude of different services provided by SHS in addition to multiple resident preferences, the associated interactions are complex and may result in undesirable system behaviour that may affect overall functionality [8]. In order for a service to achieve a goal, such as ventilating a room, a number of features (tasks) need to be carried out, for example, switching on an extractor fan, opening a door and opening a window, unfortunately, these features are also associated with other services. In the example provided here (ventilating a room), these services could include the security service concerned with doors and windows, or the power management service which may be

Manuscript received January 6, 2015; revised April 17, 2015.

doi: 10.18178/ijeee.4.1.66-73

concerned with extractor fan usage, or the air conditioning service which would be concerned with both. Therefore, there is a conflict between features, referred to as Feature Interactions (FI). In the smart home environment, such interactions are varied and occur as a result of either service-based interactions, which includes conflict goals within or between services, or as a result of policy-based interactions which take place when there is conflict between resident preferences. Let's consider two scenarios where the undesirable interaction could occur. First, when residents are away from home accordingly, the power management service will be activated to save energy by switching off electronic devices. Meanwhile, If there is a suspicious movement around the house the security service will be triggered and go through several actions, for example send SMS to the landlord and switch on some devices such as light and TV to give impression that the house is occupied (Fig. 1A).

(A) Service-Based interaction



(B) Policy-Based interaction

Figure 1. Types of interactions

Apparently, there will be a conflict as one service works to switch on some devices while another one works to keep them off. The second scenario raises when two house residents are occupied the same room each one of them have different preferences (policies) and acquire SHS to provide services according to their preferences for example to set TV in the sitting room to the favourite programme (Fig. 1B).

If one of those residents prefers documentary programs while the second one prefers sports programme, due to the contrast in preferences between house residents utilising the same services (in this scenario entertainment service) there will be interaction known as policy-based interaction.

The aim of this research is to explore a novel framework for FI detection and resoultion in SHS. The reminder of this paper is organised as follows. Section II discusses the previous literature related to the problem. Section III presents a general archticture for SHS followed by a discussion about features hiraracy concept. As a foundation of the proposed soultuion, an agent-based archtitucture for SHS is presented in Section IV. Section V gives insight on negotiation mechanism in Agent-Based Negotiation System ABNS and how it works. A case study is presented in Section VI. Finally, Section VII presents our conclusion.

II. Related Work

Taking into account the different terms used for SHS such as Home and Building Automation Systems (HBAS), Networked Home Appliances (NHA) and Home Network Systems (HNS), several studies have been carried out in order to detect and resolve FI in SHS. Most of the mechanisms, protocols and techniques utilised in these studies, similar to research on other software systems, have been inspired by prior research on FI detection and resolution in telecommunications systems which was the first domain FI problem has been addressed [9].

Some of the proposed solutions are implemented during the design phase of the system, for example, the use of formalism [10]. One such approach is Linear-time Temporal Logic (LTL) which is used to model features using the Promela language [11]. This approach proposes a SPIN model checker to detect any possible interactions between features. Another off-line detection approaches is a service-centric framework for feature interactions in the HNS integrated services [12]. It proposed to reveal all potential interactions within home network systems. Both approaches adopt an off-line technique, which is performed before the deployment of the system, even though most FI problems can be discovered only at run-time. Additionally, they consider a wide scope of unlikely feature interactions which would not normally appear in a real run. What is more, off-line technique does not support features scalability.

As mentioned, the limitation of the off-line technique is that it is performed before the deployment of the system, missing problems that occur at run-time. Hence, using only the off-line technique to solve FI problems is insufficient. As a response to this issue some approaches use the on-line technique. One such example is the Policy Interactions Manager Module (PIMM) proposed to extend the traditional networking systems which use KNX communications standard for home and building automation [13]. PIMM is integrated with the KNX home network as part of the Engineering Tool Software (ETS) suite to work as a run-time interactions manager to detect and resolve any undesirable interactions. Similarly, a proposed formal method approach which is identified as Semantic Web-based policy interaction detection method (SPIDER) [14] and a Semi-formal method known as Identifying Requirements Interactions (IRIS) [15], both methods are proposed to detect and resolve features interactions in SHS however, PIMM, SPIDER and IRIS can only solve feature interactions that take place as a result of policy-based conflicts and not service-based or device-based conflicts. In the same sequence, the work in [16] uses the Belief-Desire-Intention (BDI) agent model also to solve conflicts which only are result of the contrast between inhabitants' desires i.e. policy-based. The approach keeps alternative proposals in a library allowing the agent with minimal risk to choose alternative proposal in case of conflict. Although the negotiation between agents is adopted in this study [16], only the agent with high priority which is referred to as maximum risk agent will take the advantage. In other words, there is no compromise allowing conflicted agents of working simultaneously. Likewise, Entity-Relationship (ER) model is presented in [17]. The model creates a corresponding domain which captures the relevant components having context information to determine the conflict between occupants' preferences. Besides, not covering service-based conflicts, ER model resolves only conflicts related to user comfort such as light, sound and temperature. Another study introduces the notion of resource locking [18]. In this approach the access attribute to each device is set to Not Shared (NS) or Shared (S). NS means a device can only be used by one service at any one time, S means multiple services can control the device concurrently. Resource locking approach employs a three-layer architecture service, device and environment to present a clear and helpful taxonomy of feature interactions, it works as a feature manager and is implemented as a service on the OSGi platform to detect and resolve any FI. However, this approach is not flexible. The notion of resource locking requires the execution or the non-execution of features, in other words, there is no way for features to reach a compromise. Moreover, it prevents interactions in the device level and not the service level which is the main concern of SHS. Additionally, looping interactions where two or more services go on infinite loop because the activation of one of them leads unintentionally to activate another one and vice versa are not resolved [18].

The work in [19] covers both off-line and on-line FI. It models each appliance as an object consisting of properties and methods in object-oriented fashion. Although, our study is inspired by the notion of negotiation and compromise which are mentioned in that work, the proposed solution does not target FI which is caused by residents' policies also the scheme to resolve the on-line FI is not clarified. The negotiation agents approach is firstly adopted in the telecommunication domain to resolve FI [20]. The proposed system was able to detect and resolve FI at run-time (on-line) automatically without the need for users or service providers to know about other users or service providers have of features or settings. However, the proposed solution is not applicable for smart homes domain because SHS are more complicated and have much more features and they implemented in a dynamic environment in which they have to respond to variety of context.

From the discussion above the works which have been done so far for FI detection and resolution in SHS fall in three categories namely off-line, on-line and both (off-line and on-line together). The adopted mechanisms were vary and most of the works targeted only one type of FI i.e. service-interaction [18] or policy-interaction [14], [16], [17], [21]. Although, there are some works targeted both interactions [11], [22], the negotiation approach to reach a compromise has not been adopted yet. The significant of that approach is enabling services with conflicted features or contrary residents' policies of working simultaneously.

III. Architecture and Hierarchy Concept

Below is a visual representation of the general architecture of SHS as shown in Fig. 2. The main components of this architecture are the residents and their preferences, the devices and the services provided by those devices, sensors and actuators and the system manager. The SHS and the associated components operate within and respond to context variations such as light or heat. According to the residents' preferences and the features of each service the system manager responds to the triggering context by controlling devices to be operated in the relevant way. For example, between 7 pm and 9 pm if resident x enters the office room, the reading mode which is a service within entertainment service will be activated, accordingly, lights will be dimmed and volume of the TV in the next room will be kept below 20.

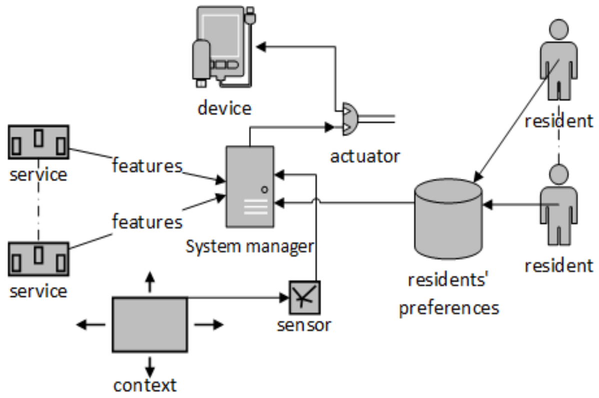


Figure 2. Smart homes systems (SHS) architecture

Most of the services within SHS could be broken down into simple features [18], [23]. To fulfil the goal of the service, all features have to be carried out. If we take the air conditioning as an example, such features would include the following:

1) Switch on the air conditioning
2) Close window 1
3) Close window 2
4) Close the door
5) Switch off the ventilation fan

It is important to note that from the above-listed features there is a hierarchy of priority, whereby it is necessary to switch on the air conditioning; however, closing the door is less important.

Another example is the home cinema service within the entertainment service. The functions of the home cinema service are to play films and adjust the room settings to create an ideal atmosphere for watching films. When this service is activated, in order to achieve its goals, the following features have to be carried out:

1) Switching TV channel to DVD player.

2) Set speaker volume to 70
3) Dim lights to 10.
4) Close the curtains
5) Close the door

Again, as with the previous example, switching the TV channel is a necessity and the additional features follow are in order of priority. However for quick and optimum performance all features should be performed.

In isolation if the service is run only by itself at a particular time, all features will operate optimally. This principle also applies where only one resident is utilising a particular service where all features will operate optimally in response to the resident's preferences. However, most of the time more than one service needs to function or more than one resident is making requests. Hence, there are potential conflicts in the case where there are multiple services using the same device, multiple users using the same service or multiple users using multiple services. In each of these cases, undesirable interaction may occur between the features of the involved services.

Thus, from the problems presented in the above it is clear that there will be conflicts within the same service and between services in terms of the features that they use. The aim therefore, is not for a service to disable a feature in order to answer the request of one resident over another or to prioritise one service over another, but rather to reach a compromise by allowing two conflicting services to run simultaneously with slightly less efficiency, which is more desirable than totally disabling one service in favour of another.

As has been mentioned earlier, a service is comprised of a hierarchy of features organised according to their importance in terms of their functional contribution to the overall service. The optimum performance of each service within SHS will be set by the system provider before the system is operated for the first time. Features are categorised according to their priority, whereby some features have high priority and are essential for the service to function and other features have a lower priority. For the latter, such features can be compromised to a certain extent while still allowing the service to perform its function.

In the following examples as shown in (Fig. 3), for features that are ranked at the top no compromise is possible, while for other features, which are considered preferable but not essential, there is the possibility for compromise. This is the case for both order of priority of features within a service and order of priority of features according to resident preferences.

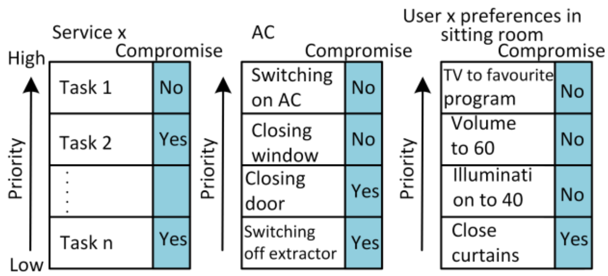


Figure 3. Features hierarchy

Priority of features within services and resident preferences should be accurately established before setting up the SHS. In establishing such a system it is important to keep in mind how to ensure services run (at least with minimum efficiency i.e. at worst only the important features will be activated) in a situation where features need to be compromised.

To provide an example, the preference of resident x in the sitting room is to switch the TV channel to his favourite program as a top priority, which cannot be compromised, while closing curtains is preferable, where there could be a compromise (Fig. 3). An example of such a compromise could be when there is another resident in the room who prefers the curtains to be open, and the compromise is that the curtains are partly opened.

IV. SHS Agent-Based Architecture

It is necessary to model each service and resident as an agent within the SHS (Fig. 4) because agents possess the following qualities [24]:

- Autonomous: refers to the capability of an agent to act based on its own experiences without direct external control
- Interactive: an agent could communicate with its environment and with other agents.
- Adaptive: is able to respond to its environment and to other agents in different situations because it is able to change its behaviour according to new conditions.
- Proactive: its reaction is goal oriented and not driven by other entities. It is designed to be capable of taking initiative.
- Coordinative: agents are able to achieve shared goals with other agents by using an adequate mechanism of communication and cooperation.

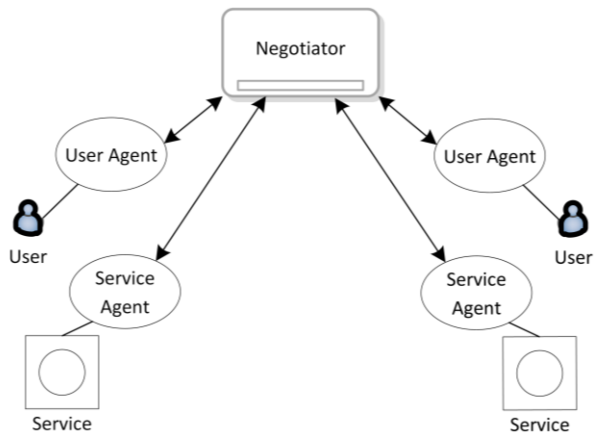


Figure 4. The structure of the negotiation system in SHS

The agent of the service registers the features of that service. Similarly, the agent of resident registers the preferences of that resident. Both service agent and resident agent register either features or preferences to be performed in order of priority for a given situation. When a service agent is activated it needs to enable its features. To avoid undesirable interactions between the features of different services there is a need for a negotiating mechanism that will first, detect feature interactions (FI), and secondly, provide a resolution based on compromise. Specifically, agents will send features individually as proposals to a negotiator agent who acts as a coordinator and decision maker between agents. The following section will explain the negotiating system.

V. ABNS Mechanism

The negotiation system in SHS establishes a communication channel between concerned agents every time a feature needs to be carried out. The main goal of the negotiation process in ABNS is to enable two or more agents, which share the same devices, and thus, potentially have conflicting goals, to work simultaneously. The negotiation session will involve the initiating agent, the negotiator and the involved agents. The involved agents are any other agents that utilise the targeted device. For example, on a warm day in order to save energy, the power management service agent sends a proposal to open a window allowing breeze to lower room temperature instead of activating the AC. The negotiator, using a register of all service agents that utilise the window controller device such as security, safety, HVAC and lighting in addition to the power management service agent, will send the proposal to each of these services requesting their approval. Each involved agent will evaluate the received proposal then send back a response to the negotiator indicating whether the proposal could be accepted, compromised, counter proposed or rejected. In the following, the implications of each response are presented:

Accept: This decision will be taken either when the involved agent is currently ‘off’ and is not concerned with controlling the device or the proposal is in line with the current feature of the involved agent, for example, both agents wish to open the window.

Compromise: There will be a compromise if an ‘accept’ position could not be achieved. Many features in SHS have a degree or level of performance, for example:

- Lighting has a range between 0-100, where 0 means totally dark (lights off) and 100 means full brightness.
- Sound between 0-100, where 0 is mute and 100 is full volume.
- AC 18-30 degrees Celsius.
- Windows, doors and curtains 0-100, where 0 is fully closed and 100 is fully opened.

A compromise is reached by taking the middle point between the performance values of two conflicting features. For example, if an agent wants to set volume to 80 while another wants to set it to 40 the middle point will be 60, the same principle applies to windows, where if an agent wants a window to be closed and another wants it to be open, the compromise will be that the window is half opened.

Counter proposal: If there is no possibility of a compromise due to the nature of the feature or because the resident does not allow for a compromise, there will be a counter proposal. As mentioned previously an agent registers features in hierarchical order. If the received proposal is neither accepted or compromise cannot be reached, the involved agent will send a counter proposal in an attempt to satisfy the initiating agent. For the counter proposal only the feature with least priority will be offered for compromise. For example, if the AC agent is active and receives a proposal to switch off the air conditioning, this particular feature has the highest priority and cannot be compromised as shown in Fig. 3. Therefore, the counter proposal will be to switch on the ventilation fan. Before considering the counter proposal, the initiating agent will check if it has priority over the service through consulting the negotiator, if the initiating agent has priority the counter proposal will be rejected.

Reject: A proposal is rejected if a service agent is ‘ON’ and there is no possibility for the proposal to be accepted or to be compromised and also there is no counter proposal to be proposed.

Although, our proposed solution of FI detection and resolution in SHS aims to avoid utilising priority as much as possible, there are some situations where the negotiator has to decide which agent has priority to control a device. Specifically, these situations include where a counter proposal is rejected or the original proposal is rejected. In order to determine priority, the negotiator has a register in advance shows the priority of each agent whether service agent or resident agent. For example, because being safe and secure are absolute priorities for any householder, some service agents are considered super agents that have priority over all other agents, namely; safety and security agents.

In a given proposal if one of the agents involved in negotiation has a feature that can be compromised and does not have a level of performance they can give the initiating agent the right to fully control the device that the feature utilises if there is no possibility of compromise for the initiating agent. If a feature can be compromised for both agents the full control of the feature will be given to the agent with the highest priority. In the situation where none of the agents have priority, the initiating agent will take control of the feature.

A. *The Negotiating Algorithm*

When a service agent is triggered it responds by activating a number of features (Fig. 5). These features control some devices in order to fulfil the request. The service will send a proposal for each feature in turn in order of importance to the negotiator. The negotiator will receive the proposal then do the following:

1) Check which other agents (involved agents) should approve the proposal.
2) Send the proposal to involved agent or agents.
3) Receive agent’s reply which can be accepted, compromised, counter proposed or rejected.
4) Based on the involved agent’s response one of the following scenarios takes place:

- Accept: The negotiator will inform the initiating agent that the proposal is accepted and it will send the proposal to the targeted device to be performed.

- Compromise: The negotiator will check if the proposal can be compromised, specifically, the negotiator checks with the initiating agent if the compromise is acceptable, if so the negotiator calculates the middle point performance value of the feature then sends the instruction to the device. If no compromise can be achieved priority will be applied.

- Counter proposal: The negotiator sends a counter proposal from the involved agent to the initiating agent, if the initiating agent has priority the counter proposal will be rejected and the original proposal will be performed. Otherwise, the initiating agent will check if the counter proposal is possible through checking its features list, if so the counter proposal will be accepted, if not it will be rejected and priority will be applied.
- Reject: The negotiator will determine priority between the two agents (initiating and involved). If the initiating agent has priority the proposal will be sent to the targeted device to be performed and if the involved agent has priority the proposal will not performed.

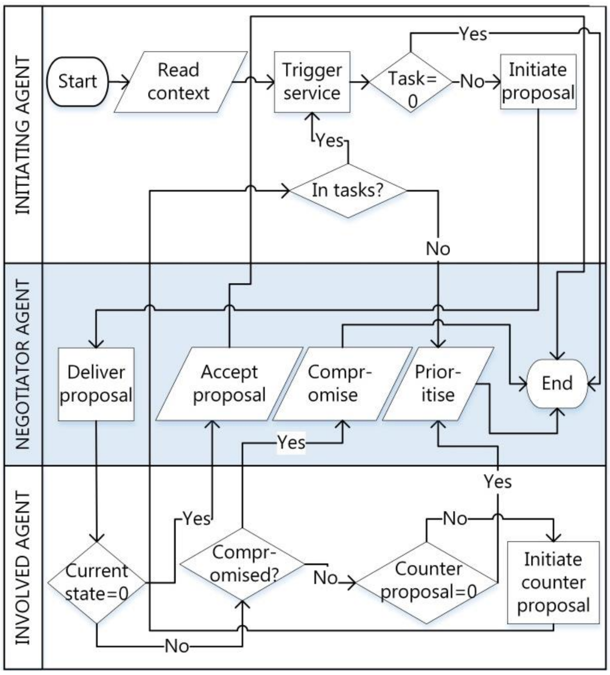


Figure 5. Negotiation process

VI. CASE STUDY

Here a case study is presented in order to illustrate the processes described in the above. The ventilation service is concerned with ventilating the air in a room. This service will be triggered, for example, if there is an unpleasant odour. In order to achieve the goal of this service the following features have to be carried out:

TABLE I. PROPOSALS OF VENTILATION SERVICE

Proposal ID	Action	Device	Value	Compromise
1	switch on	ventilation fan	--	No
2	set	window 1	100	No
3	set	window 2	100	Yes
4	set	door	100	Yes

As shown in Table I, the first two features, to switch on the ventilation fan and open window 1, are essential and cannot to be compromised, while other features are preferable but can be compromised. In order for the first feature to be performed, a proposal to switch on the ventilation fan will be sent from ventilation service agent (initiating agent) to the negotiator. The negotiator has a registry of all devices within the SHS and all agents that have a link with each device. Table II shows a view of the negotiator registry.

TABLE II. VIEW OF NEGOTIATOR REGISTRY

Device ID	Device Name	Controlling Agents
101	air conditioning	Security, Safety, AC, Ventilation, Heating, Power management
102	ventilation fan	Security, Safety, AC, Ventilation, Heating, Power management
103	window 1	Security, Safety, AC, Ventilation, Lighting, Entertainment
104	window 2	Security, Safety, AC, Ventilation, Lighting, Entertainment
105	door	Security, Safety, AC, Ventilation, Lighting

For each of the features that need to be performed there is an associated device, in the case study presented here these devices are the ventilation fan, window 1, window 2 and the door (Table II). It is important to note that the device does not refer specifically to the window or the door, but rather to the device which opens and closes the windows or door. These devices have a value of between 0 to 100, where 0 means fully closed and 100 means fully opened. Besides the ventilation service agent there are several other agents that have a link with those devices, namely; Security, Safety, AC, Ventilation and Power Management, Lighting and Entertainment service agents (Table II). The negotiator (as previously mentioned) sends each proposal to all agents that have a link with the device which is to be controlled in the proposal. However, for the purposes of this case study only the response of the AC service agent will be considered, the assumed condition here is that it is active due to hot weather. The status of the features is presented in Table III.

TABLE III. PROPOSAL OF AIR CONDITIONING SERVICE

Proposal ID	Action	Device	Value	Compromise
1	switch on	Air conditioner	18	No
2	set	window 1	0	No
3	set	window 2	0	Yes
4	set	door	0	Yes
5	Switch off	ventilation fan	--	Yes

A. *Negotiation Mechanism*

As has been stated previously, agent priority is established in the negotiator based on the services weight and residents preferences, for the purposes of the case study we will consider the situation when the initiating agent (in our scenario the ventilation agent) has priority.

The ventilation service agent sends the first proposal to the negotiator and waits for a reply before sending the next proposal, explained in more detail below:

- The first proposal is to switch on the ventilation fan. It cannot compromise and it has no level of performance, therefore, the agent with the higher priority will take the advantage. The AC agent will receive the proposal and check its current status and because it is "ON" it cannot accept the proposal because one of its own features is to have the ventilation fan switched off (Table III), however, it will inform the negotiator that there is a possibility for compromise. The negotiator will relay the response 'compromise' of the AC agent to the ventilation agent. As it is a feature that cannot be compromised by the ventilation agent it will be rejected. The negotiator will give the advantage to the ventilation agent because the feature is a necessity and not to the AC agent as the feature can be compromised, and priority is not a consideration here. Accordingly, the order will be sent to the ventilation fan to be switched "ON".
- The second proposal from the ventilation service is to fully open window 1, which cannot be compromised, will be received by the negotiator then delivered to the AC agent. The AC agent will check its current status which is "ON" and therefore, will inform the negotiator that is cannot accept the proposal and there can be no compromise for this feature (Table III). However, because the AC agent has other features which are less important than the proposed feature and can be compromised, it will send a counter proposal offering the advantage of its least important feature. In this case the counter proposal will be 'set door 100' i.e., fully open the door. The negotiator will deliver the counter proposal to the ventilation agent. Before considering the counter proposal the ventilation agent will first check if it has priority over the AC agent. Because in this case it has higher priority, it will reject the counter proposal and the negotiator will send the original proposal, to fully open window 1, to the targeted device.
- The third and the fourth proposals are similar to each other whereby the ventilation agent requests to fully open window 2 and the door, both of which can be compromised by both the initiating and involved agents. The negotiator will send the proposals one by one to the AC agent which will examine its current state and send the 'compromise' reply accordingly. Because these features can also be compromised by the ventilation agent, the negotiator will calculate a compromise performance value for each, this is achieved through the following equation:

$$(\text{feature performance value in proposal+feature performance value in involved agent})/2$$

$$(100+0)/2 = 50$$

The outcomes from the above scenario were to keep the ventilation fan and air conditioner switched on, window 1 fully opened and window 2 and the door half opened. In this negotiation both services reached a compromise allowing them work simultaneously with a slight advantage for ventilation agent as it had priority.

VII. Conclusion

Smart homes systems (SHS) are complex and include several services that respond to the preferences of multiple residents. FI is an inevitable consequence which can prevent services functioning properly and may result in unpredictable behaviour of services provided by SHS.

This study utilised an agent-based approach to model the stakeholders within the SHS because an agent possesses the required attributes of the actors of the system, namely; services and residents. The study showed that several features need to be achieved in order to satisfy the goals of these agents. A hierarchy of such features was presented to illustrate their priority in relation to the performance of each agent.

A negotiation process as a solution to the problem of FI was presented. This negotiation included the initiating agent, the negotiator and involved agents who share the device controlled by associated features. The negotiation process successfully negotiated a compromise between two or more conflicting agents, taking into consideration their need to function simultaneously and then the priority that they have. A case study was used to show the outcome from the negotiation process in case of a conflict scenario. The agent-based negotiation system (ABNS) is a novel contribution because it seeks to find a compromise between conflicting agents before considering priority through detection and resolution of FI in SHS.

References

[1] H. Lee, S. J. Park, M. J. Kim, J. Y. Jung, H. W. Lim, and J. T. Kim, "The service pattern-oriented smart bedroom based on elderly spatial behaviour patterns," *Indoor and Built Environment*, vol. 22, pp. 299-308, 2013.

[2] A. S. Al-Sumaiti, M. H. Ahmed, and M. M. Salama, "Smart home activities: A literature review," *Electric Power Components and Systems*, vol. 42, pp. 294-305, 2014.

[3] P. Rashidi and D. J. Cook, "Adapting to resident preferences in smart environments," in *Proc. AAAI Workshop on Preference Handling*, 2008, pp. 78-84.

[4] F. Aldrich, "Smart homes: Past, present and future," in *Inside the Smart Home*, R. Harper, Ed., Springer London, 2003, pp. 17-39.

[5] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homes—Past, present, and future," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, pp. 1190-1203, 2012.

[6] T. Perumal, A. R. Ramli, C. Y. Leong, S. Mansor, and K. Samsudin, "Interoperability for smart home environment using web services," *International Journal of Smart Home*, vol. 2, pp. 1-16, 2008.

[7] H. J. Wang, H. Y. Li, K. M. Sun, and L. S. Yi, "Three-Meter AMR system with grid demand-side energy management in smart home system," *Applied Mechanics and Materials*, vol. 473, pp. 153-159, 2014.

[8] P. Zave and M. Jackson, "New feature interactions in mobile and multimedia telecommunications services," *FIW*, pp. 51-66, 2000.

[9] T. F. Bowen, F. Dworack, C. Chow, N. Griffeth, G. E. Herman, and Y. Lin, "The feature interaction problem in telecommunications systems," in *Proc. Seventh International Conference on Software Engineering for Telecommunication Switching Systems*, 1989, pp. 59-62.

[10] M. Calder, E. Magill, and D. Marples, “Hybrid approach to software interworking problems: managing interactions between legacy and evolving telecommunications software,” *IEE Proceedings - Software*, vol. 146, pp. 167-175, 1999.

[11] P. Leelaprute, T. Matsuo, T. Tsuchiya, and T. Kikuno, “Detecting feature interactions in home appliance networks,” in *Proc. Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 2008, pp. 895-903.

[12] M. Nakamura, H. Igaki, and K. Matsumoto, “Feature interactions in integrated services of networked home appliance,” in *Proc. Int’l. Conf. on Feature Interactions in Telecommunication Networks and Distributed Systems*, 2005, pp. 236-251.

[13] M. Shehata, A. Eberlein, and A. O. Fapojuwo, “Managing policy interactions in KNX-based smart homes,” in *Proc. 31st Annual International Computer Software and Applications Conference*, 2007, pp. 367-378.

[14] H. Hu, D. Yang, L. Fu, H. Xiang, C. Fu, J. Sang, C. Ye, and R. Li, “Semantic web-based policy interaction detection method with rules in smart home for detecting interactions among user policies,” *IET Communications*, vol. 5, pp. 2451-2460, 2011.

[15] M. Shehata, A. Eberlein, and A. Fapojuwo, “Using semi-formal methods for detecting interactions among smart homes policies,” *Science of Computer Programming*, vol. 67, pp. 125-161, 2007.

[16] C. Hsu and L. Wang, “A smart home resource management system for multiple inhabitants by agent conceding negotiation,” in *Proc. IEEE International Conference on Systems, Man and Cybernetics*, 2008, pp. 607-612.

[17] P. Carreira, S. Resendes and A. Santos, "Towards automatic conflict detection in home and building automation systems," *Pervasive and Mobile Computing*, vol. 12, pp. 37-57, 6, 2014.

[18] M. Kolberg, E. H. Magill, and M. Wilson, “Compatibility issues between services supporting networked appliances,” *IEEE Communications Magazine*, vol. 41, pp. 136-147, 2003.

[19] H. Igaki and M. Nakamura, “Modeling and detecting feature interactions among integrated services of home network systems,” *IEICE Trans. Inf. Syst.*, vol. 93, pp. 822-833, 2010.

[20] N. D. Griffeth and H. Velthuijsen, “The negotiating agents approach to runtime feature interaction resolution,” *Feature Interactions in Telecommunications Systems*, pp. 217-235, 1994.

[21] M. Shehata, A. Eberlein, and A. O. Fapojuwo, “Managing policy interactions in KNX-based smart homes,” in *Proc. 31st Annual International Computer Software and Applications Conference*, 2007, pp. 367-378.

[22] M. Wilson, E. H. Magill, and M. Kolberg, “An online approach for the service interaction problem in home automation,” in *Proc. 2005 Second IEEE Consumer Communications and Networking Conference*, 2005, pp. 251-256.

[23] A. Classen, P. Heymans, and P. Schobbens, “What’s in a feature: A requirements engineering perspective,” in *Fundamental Approaches to Software Engineering*, F. José, Ed., Springer, 2008, pp. 16-30.

[24] Agent Working Group, “Agent technology green paper,” Object Management Group (OMG), 2000.

Ahmed S. Alfakeeh is a lecturer at faculty of computing and information technology system, King Abdulaziz University, Jeddah, Saudi Arabia. Currently, he is a PhD candidate in the final year at Software Technology Research Laboratory, a research institute established within De Montfort University, Leicester, UK. His research project involves feature interactions detection and resolution in context-aware systems, smart homes systems in particular. Ahmed graduated in computer science and education in 2005 from King Abdulaziz University, and then in 2010 he received MSc in IT from De Montfort University.

Ali H. Al-Bayatti is Senior Research Fellow and Head of Intelligent Transportation theme. He has received his PhD degree titled “Security Management in MANoN” in Computer Science at the Software Technology Research Laboratory, De Montfort University.

Dr. Al-Bayatti leads the Intelligent Transportation theme where his research deals with vehicular (e.g. Vehicular Ad hoc Networks) and smart technologies (e.g. Context-aware Systems) that promote collective intelligence. Applications range from promoting comfort, to enabling safety in critical scenarios. The goal of his research is to improve the effectiveness, efficiency, mobility and safety of transportation systems. His research appeared in major academic journals, IEEE vehicular technology, and Elsevier Journal of Networking and Computer Applications. Dr. Ali Al-Bayatti is currently teaching core module in undergraduate courses in computer networks (Multi-Service Networks). He is currently the programme leader for MSc Software Engineering, MSc Cyber Technology, MSc Professional Practice in Digital Forensic and Security and Forensic Computing for Practitioners.