ISSN 2278-3091
Volume 10, No.1, January - February 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse091012021.pdf
https://doi.org/10.30534/ijatcse/2021/091012021

Formal Specification Approach for Smart Space Development

Muhammad Waqar Aziz, Ali Sayyed
CECOS University of IT & Emerging Sciences,
Peshawar, 25000, Pakistan
waqar@cecos.edu.pk, alisayyed@cecos.edu.pk

ABSTRACT
Being an important application of the Internet of Things, smart spaces are increasingly developed throughout the world for different purposes ranging from home automation to smart grids. Despite the considerable focus given to the practical development of smart spaces, there are few attempts of utilizing formal methods in this domain. Especially, the requirements of developing a smart space have not yet been formally specified, to the best of the authors' knowledge. To fill this gap, a formal specification approach is presented in this paper for smart space development. The proposed approach first identifies the key components of a smart space, then it uses a state-based formal specification language – Z, to formally specify the requirements of these components. The requirements of developing a hypothetical smart space are considered for formal specification in this paper. This work does not only demonstrate how the components of complex systems, such as smart spaces, can elegantly be modeled using a Software Engineering formalism. But it can also be used as a step towards defining a holistic smart space development framework, along with requirement engineering and system design techniques.

Key words: Formal Specification, Smart Space development, Masjid Al-Haram.

1 INTRODUCTION

The development of a smart space is an important application of the Internet of Things (IoT). A smart space is a ubiquitous computing environment, which provides the best possible services by constantly detecting the context and activities [1]. As ongoing research on IoT, several smart spaces have been developed in different domains, such as home automation [2], energy optimization [3], smart grid [4], and parking assistance [5], etc. However, the development of smart spaces requires cross-disciplinary knowledge and hence stipulates a large set of requirements. Moreover, due to some run time properties such as reliability and security, smart spaces may be considered safety-critical. Due to these reasons, the use of formal methods in the development of a smart space becomes necessary. Formal methods can be used to remove ambiguity, inconsistency, and incompleteness from the development requirements and thus provide a foundation for designing reliable systems.

The formal specification is a formal method that is used in software engineering to mathematically model the components of a system [6]. As software requirements represented in natural language are ambiguous, formal specification is used to specify each requirement accurately before its implementation. In a software engineering process, formal specifications are used to produce clear and precise Software Requirements Specification, which is needed to develop reliable safety-critical systems. For this purpose, some formal specification languages are available, such as VDM, Z, etc. Z language [7] is the most widely-used formal specification language that is based on typed set theory. It is a model-oriented language that specifies the state and transitions of each component of a system. Therefore, the author believes that Z language is well suited to not only model the components of smart spaces, but to consequently clarify the development requirements.

In contrast to the existing smart space research, which is mostly focused on the technological aspects of development, very few attempts have been made to apply formal methods in this domain. Although formal specification for different software systems can be found in the literature, it has not been previously used to specify the requirements of developing a smart space. To clarify the requirements of developing a smart space and to remove ambiguities and inconsistencies from them, the requirements need to be formally specified. Besides, the use of formal specification is also essential to model the high-level behavior of smart space in terms of its components and their properties. To this end, a formal specification approach is presented in this paper for smart space development. The proposed approach first identifies the key components of a smart space and then elaborate their detailed formal specifications in terms of state and operational schemas.

This research work proposes the development of a smart space in Masjid Al-Haram, the sacred mosque in Makkah [8]. The details of the Proposed Smart Space (PSS) are provided in Section 2. The proposed approach and its steps are presented in Section 3. To validate the applicability of the proposed approach, five different key components of the PSS are identified using the UML Use Case models, which are developed in our previous paper [8]. Then, formal specification of these components is provided by identifying their states and events causing state transitions, in Section 4. By modeling each component of the smart space in this way, provides a clear understanding of the system's behavior and the smart space development requirements (as discussed in Section 5). Although the formal specification presented in this paper are specific to the PSS, the proposed approach is not confined to this one smart space case; instead, it is generalized enough to be applied in any smart space development.

2 MODEL OF A HYPOTHETICAL SMART SPACE

With the rapidly increasing growth of the number of pilgrims coming to the Holy Mosque, there is always a mounting need of proposing and implementing new solutions for crowd control. Although the conventional methods for controlling the crowd such as human interventions can be improved by using the fast-growing Information and Communication Technologies, they are often unsuccessful and uncomfortable for the pilgrims [9]. Controlling and managing the ever-increasing occurrences of crowd poses several challenges to the authorities. The most dominant among them is crowd control as the poor control may cause stampedes among the crowd or accidents leading to injuries or even death. To provide a solution, this research work proposes the development of a smart space in Masjid Al-Haram that would provide convenient and intelligent services both to the pilgrims and the authorities.

It is planned that the PSS would use environment sensing and crowd estimation. The ambient information provided by the environment sensing includes some of the effective physical properties of smart space, such as temperature, illumination, and air quality sensing that would allow smart control of electric lights, air conditioners, and ventilators, respectively. Whereas, the selected crowd estimation method would allow the smart control of doors and electronic display screen to provide crowd guidance. In this way, the PSS is supposed to provide a crowd control and management strategy that can inform, restrict, and route the pilgrims to less crowded areas. In our previous work [8], the smart space models, application scenarios, and descriptions are developed for Masjid Al-Haram. The identified actors, use cases, and their interaction are illustrated through UML Use Case models (an example is shown in Figure 1).

3 PROPOSED APPROACH FOR SMART SPACES

The proposed formal specification approach for smart space consists of two steps: identification of the key components of the smart space and their formal specification. These steps are explained in this section.

3.1 Components Identification

To elicit and capture the requirements of developing a smart space, this work utilizes the requirement engineering technique for smart spaces [8]. The presented technique [7] allows the requirement gathering during smart space development in a systematic way. First, the possible actions occurring in the smart space are identified, where each action is represented by a use case.Then, the related actions are grouped into different application scenarios based on common functionality.Finally, the elicited requirements are specified using the modified Use Case models [8]. The approach presented in this paper adopts these developed use case models and considers each action group as one component.

3.2 Formal Specification

Each component may have different states as well as associated events that cause transitions among these states. So, the states of each component and the associated events are documented in this step. Based on this, the formal specification models for the identified smart space components are developed. For each component, the model consists of the requirement(s) to be formalized, states of the component, and three types of schemas (i.e., state, initialization and operational).

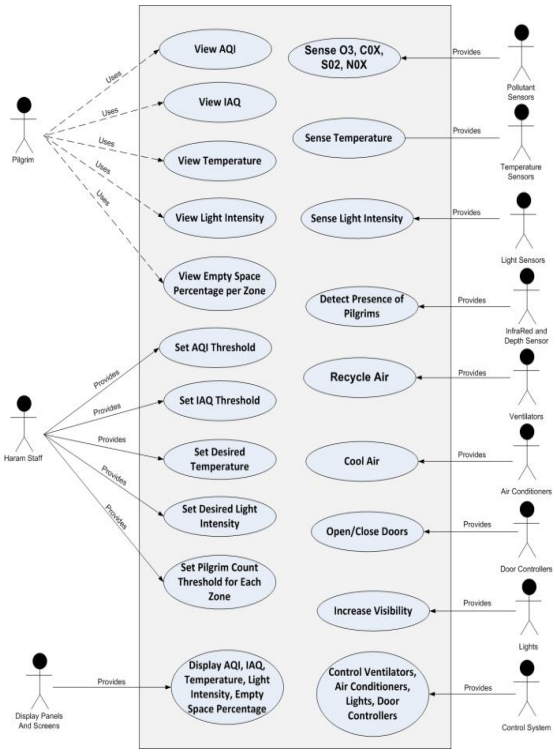

Figure1. System-level use case of PSS [8]

4 APPLICATION IN THE PSS

As a first step of the proposed approach, four different components of the PSS are identified: temperature control, air quality control, door control, and light control. In the second step, the formal specification models for the identified components are presented in the following. For each component, the model specifies the requirement and then defines its states. Then, state, initialization and operational schemas are defined. Table 1 provides a summary of states and events of each component.

Table 1. States and Events of the identified components of the PSS.

Components	States	Events
Temperature control	on, off	Manage temperature
Air quality control	low, high	Detect pollution
Light Control	on, off	Turn the light on/off
Door Control	open, close	Open/close door

4.1 Temperature Control

Requirement: *Based on the current temperature value of the smart space, turn on or off the cooling system.*

States: A free type "CSSTATUS" is used to represent on and off states of the cooling system.

$$CSSTATUS == \{on, off\}$$

State schema: The *Temperature* schema is defined that contains four variables: "cs" of type "CSSTATUS" and "min", "max" and "current" of type Natural numbers, to represent minimum, maximum and current temperatures respectively.

Temperature

$cs : CSSTATUS$

$min, max, current : \mathbb{N}$

In the *initialization* schema, the "min" and "max" variables are given initial values of 16 and 25, to set the minimum and maximum threshold values of the temperature respectively. The variable "cs" is given the value on, which means that initially, the cooling system would be on.

InitTemperature

Temperature

$cs = on$

$min = 16$

$max = 25$

Operational schema: One *operational* schema is used to manage the temperature control, based on the event of turning on or off the cooling system.

1. Turn on the cooling system, as the current temperature increases from the maximum temperature limit. Similarly, turn off the cooling system as the current temperature decreases from the minimum temperature limit.

ManageTemp

$\Delta Temperature$

$current? : \mathbb{N}$

$current? < min \Rightarrow cs = off$

$current? > max \Rightarrow cs = on$

4.2 Air Quality Control

Requirement: *Based on the current value of air pollution in the environment, turn on or off the ventilator.*

States: A free type "VENTILATORSTATUS" is defined to represent low and high states of the ventilator system.

$$VENTILATORSTATUS == \{low, high\}$$

State schema: The *Ventilator* schema is defined that contains two variables: "ven_status" of type "VENTILATORSTATUS" and "max_concen_level" of type Positive Natural numbers.

Ventilator

$ven\_status : VENTILATORSTATUS$

$max\_concen\_level : \mathbb{N}_1$

In the *initialization* schema, "ven_status" is assigned the initial values of low to represent that the ventilator system would be operating at a low speed initially. Whereas, "max_concen_level" is assigned the maximum allowable value for the concentration level of the harmful gases in the air.

InitVentilator

Ventilator

$ven\_status = low$

$max\_concen\_level = maximum\_allowable\_value$

Operational schema: The following *operational* schema represents the event of turning the speed of the ventilator system low or high.

1. Turn the speed of the ventilator system high when the current value of air quality index (aqi) becomes greater than the maximum allowable value for the concentration level of the harmful gases and vice versa.

PolutionDetection

$\Delta Ventilator$

$aqi? : \mathbb{R}$

$aqi? > max\_concen\_level? \Rightarrow ven\_status = high$

$aqi? < max\_concen\_level? \Rightarrow ven\_status = low$

4.3 Light Control

Requirement: *Whenever a person enters the smart space, the light automatically turns on, and when there is no person the light turns off.*

States: A free type "LIGHTSTATUS" is used to represent different states of light that are on and off.

$$LIGHTSTATUS == \{on, off\}$$

State schema: The *Light* schema is defined that contains two variables: "light_status" of type "LIGHTSTATUS" and "person_count" of type Natural numbers.

Light

$light\_status : LIGHTSTATUS$

$person\_count : \mathbb{N}$

In the *initialization* schema, these variables are given initial values of off and zero respectively. It means that initially the light would be off and there would be no person in the smart space.

InitLight

Light
light_status : *off*
person_count = 0

Operational schemas: There would be two *operational* schemas for light control based on events of turning the light on and off.

1. As a person enters the smart space, the light would on while the person count is incremented by one.

LightOn

Δ*Light*
person_count′ = *person_count* + 1
light_status = *on*

2. When a person leaves the smart space, the person count is decremented by one. If the person count reaches zero, it means that there is no person in the smart space, the light would off.

LightOff

Δ*Light*
person_count′ = *person_count* - 1
person_count′=0 ⇒ *light_status* = *off*

4.4 Door Control

Requirement: *When there are less than 100 persons in the smart space, the door will be open. However, it will close automatically whenever the persons in the smart space increase from this set value.*

States: A free type "DOORSTATUS" is used to represent the open and close states of the door.

$$DOORSTATUS == \{open, close\}$$

State schema: The Door schema is defined that contains two variables: "door_status" of type "DOORSTATUS" and "person_inside" of type Natural numbers.

Door

door_status : *DOORSTATUS*
person_inside : ℕ

In the *initialization* schema, these variables are given initial values of open and zero respectively. It means that the door would be open and there is no person in the smart space on initialization.

InitDoor

Door
door_status = *open*
person_inside = 0

Operational schemas: There would be two *operational* schemas for the door based on events of opening or closing the door.

1. When a person enters the smart space, the *person_inside* counter is incremented by one. The door would close when the persons inside the smart space would reach to 100.

PersonEnter

Δ *Door*
person_inside′ = *person_inside* + 1
person_inside′ ≥ 100 ⇒ *door_status* = *close*

2. Whenever a person goes out of the smart space, the *person_inside* counter is decremented by one. The door would open when the persons count inside the smart space would be less than 100.

DoorOpen

Δ*Door*
person_inside′ = *person_inside* - 1
person_inside′ < 100 ⇒ *door_status* = *open*

5 RELATED WORK AND DISCUSSION

Although the use of formal specification in software engineering is not new [6], it is still used to specify the requirements of software systems. For instance, the formal specification of an online food ordering system is presented in [10]. In the case of smart space development, there is a scarcity of applying formal specifications. However, individual efforts can be found that are confined to specific types of smart spaces. For example, a formal specification framework for smart grid components is presented in [11]. Since the framework [11] considers the components specific to the smart grid, it is difficult to generalize the framework [11] to other types of smart spaces. Likewise, the components and scenarios, defined in [12],are specific to smart traffic monitoring and control system. Moreover, the formal representation developed [12] is based on the UML sequence diagram. Similarly, the components modeled in [13] are specific to the logistic monitoring system, which are difficult to generalize for other types of smart spaces. Although the components somewhat similar to this paper are also considered in [14], the work is focused on user login and central system modules of a smart home monitoring system.

In contrast to the above efforts, this research work aims at developing a framework for smart space development. The focus of this work is on requirement engineering, as it is not easy to understand the requirements during smart space

development. To this end, previously we have presented the requirement engineering technique for smart spaces [8] to systematically identifying the requirements. As a next step, the approach presented in this paper uses the technique [8] to identify the components of the smart space. Then it provides a formal model of the identified components to lead towards abetter understanding of the requirements. Further, in contrast to most of the above-mentioned efforts, this work uses Z formal specification language, which the author believes is easy to understand. This is confirmed by the application of the proposed approach in the smart space case study. Moreover, the application of the proposed approach has demonstrated that by formally specifying the requirements, the drawbacks of a semi-formal specification or specifying the requirements in natural language can be removed.

6 CONCLUSION AND FUTURE WORK

With the aim of proposing a software engineering-based holistic framework for smart space development, a formal specification approach is presented in this paper. The proposed approach, as a first step, identifies the components of a smart space. In the second step, the identified components are modeled using the Z formal specification language. The applicability of the proposed approach in a smart space case study has evidently demonstrated the clarity, consistency, and completeness in the development of software requirements. The author would like to mention here that the proposed approach can be used in the future as a part of a holistic smart space development framework.

REFERENCES

1. S. Poslad, ***Ubiquitous computing: smart devices, environments and interactions***. Published by the John Wiley & Sons 2011.
2. S.Minghong, Y.Hongbing, andS.Mingying, "**Research on the Smart Home System Based on IOT"**,*Advanced Materials Research*, PP. 488 – 489, 2012.
3. G.Vogt et al, **Use Cases for SMARTSPACES services and systems**, Deliverable D1.2, Revision 1, available at http://www.smartspaces.eu/outputs/.
4. J. F. Martins et al,**Smart Homes and Smart Buildings**. in *Proc. of 13th Biennial Baltic Electronics Conference* BEC2012, Tallinn, Estonia, October 3-52012.
5. J.Chinrungrueng, U.Sunantachaikul, and S.Triamlumlerd, **Smart Parking: an Application of optical Wireless Sensor Network**,in *Proc. of the International Symposium on Applications and the Internet Workshops* SAINTW'07, 2007.
6. A. Hall, "**Seven myths of formal methods"**, *IEEE Software*, Vol. 7, Number 5, pp. 11–19, 1990.
7. J. M. Spivey, "**An Introduction to Z and Formal Specifications**", *Software Engineering Journal*, Jan 1989.
8. M. W.Aziz, A. A. Sheikh, E. A.Felemban, Requirement engineering technique for smart spaces,in *Proc.of the International Conference on Internet of things and Cloud Computing*, pp. 1-7. ACM, Cambridge,March,2016.
9. M. W.Aziz, et al,"**Automated Solutions for Crowd Size Estimation**",*Social Science Computer Review*, Vol. 36, Number 5, pp. 610-631, 2018.
10. P.Saratha, G. V. Uma, B.Santhosh, **Formal Specification for Online Food Ordering System Using Z Language**, in *Second IEEE International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM)*,pp. 343-348, 2017.
11. W.Akram, M. A.Niazi, "**A Formal Specification Framework for Smart Grid Components"**. *Complex Adapt System Model*, Vol. 6, Number 5, 2018.
12. U. N.Abbas,N. A. Zafar,F. Ullah, "**Formal Modeling and Verification of Smart Traffic Environment with Design Aided by UML"**, *International Journal of Advanced Computer Science and Applications,*Vol. 7, Number 12, pp. 165-172, 2016.
13. S.Saddiq, N. A. Zafar, F. Ullah, **Formal modeling of smart logistics monitoring**,in *Proc. of 1st IEEE International Conference on Electronics, Materials Engineering and Nano-Technology* (IEMENTech), (2017, April).
14. S.Farooq, N. A.Zafar,F. Ullah, **Formal modeling of smart home monitoring system**,in *Proc.of 1st IEEE International Conference on Electronics, Materials Engineering and Nano-Technology(IEMENTech)*, pp. 1-6 (2017, April).