Using Magentix2 in Smart-Home Environments

S. Valero, E. del Val, J. Alemany and V. Botti

Abstract In this paper, we present the application of a multi-agent platform Magentix2 for the development of MAS in smart-homes. Specificallly, the use of Magentix2 (http://gti-ia.upv.es/sma/tools/magentix2/index.php) platform facilitates the management of the multiple occupancy in smart living spaces. Virtual organizations provide the possibility of defining a set of norms and roles that facilitate the regulation and control of the actions that can be carried out by the internal and external agents depending on their profile. We illustrate the applicability of our proposal with a set of scenarios.

1 Introduction

Currently, there is a number of social and economic drivers behind smart homes that do specially important the research in this area. In particular, an increasing number of old people in Europe that already live alone or prefer living independently and needs assistance services. Smart home technology can be coupled with works in other areas such as tele-health in order to improve the daily living of users. Moreover the demand for increased security, energy saving, comfort and improved quality of life in the home environment is increasingly taking more importance. Thus the value of building efficient and self-adaptive smart home systems is of primordial relevancy.

S. Valero (✉) · E. del Val · J. Alemany · V. Botti
Departamento de Sistemas Informáticos y Computación,
Universitat Politècnica de València, 46020 Valencia, Spain
e-mail: svalero@dsic.upv.es

E. del Val
e-mail: edelval@dsic.upv.es

J. Alemany
e-mail: jalemany@dsic.upv.es

V. Botti
e-mail: vbotti@dsic.upv.es

© Springer International Publishing Switzerland 2015

Á. Herrero et al. (eds.), *10th International Conference on Soft Computing Models in Industrial and Environmental Applications*, Advances in Intelligent Systems and Computing 368, DOI 10.1007/978-3-319-19719-7_3

The multi-agent system paradigm (MAS) is envisioned as a strong solution of challenges in the context of smart homes. MAS are one of the most representative instances among artificial intelligent systems dealing with complexity and distribution. In the context of smart homes, agents represent the entities in the environment and can be considered autonomous, adaptive, context-aware and capable of making decisions about actions (behaviors) based on their observations. They can be the software interface of the ubiquitous devices that offer their services or the software interface from the user side that demands such services. In other terms, agents are called soft-sensors in smart environments and will be designed to learn from previous experiences and to reason about their local information in order to improve their decisions and achieve their goals. In this way, a smart home (or environment) can be naturally viewed as a multi-agent system with distributed intelligence.

In this paper, we present the application of the platform Magentix2 to the context of Smart-Homes. The use of Magentix2 helps as to define a MAS with normative context that allow us to: (i) define different user profiles (roles) and living spaces (organization units); (ii) control the access to the system and to the services offered by agents that are part of the smart-home in presence of several inhabitant profiles (norms); and to trace the behavior of the inhabitants.

2 Related Work

Traditionally, smart home environments have been seen as centralized systems oriented to a single-user where home appliances are connected to the home network. Advances in ambient intelligence, ubiquitous computing, and autonomic computing and their application into smart homes have added intelligence to our housing and have facilitated the decentralization of the system information and functionality. These features provide flexibility and adaptability to the smart home system. Through the incorporation of intelligent remote control systems that supervise the home appliances and devices it has been possible to improve the quality of living in several aspects: comfort, health-care, safety, security, and energy efficiency. The research in the area of smart homes has tackled different technical issues such as heterogeneity in devices and technologies, context awareness, and security in order to facilitate the implementation of intelligent environments.

Although several technical challenges have been achieved, there are still open issues. On one hand, the major part of research on smart homes until now has focused on technical challenges required to provide ubiquitous and context-aware services considering that there is a single user in the living space [2, 14]. However, usually more than one resident occupy living spaces. The main challenges related to the multi-occupancy can be classified in three groups: activity tracking [15], profiling activities and behaviors [4], and conflicts in input preferences from multiple residents [3, 6].

There are previous works in the area of MAS that deal with some of these multiple occupancy problems. Rodriguez et al. [10] propose a MAS architecture based on virtual organizations that combines data obtained by multiple sensors in order to identify and localize the position of the inhabitants in residential home for the elderly. Sun et al. [12] propose a multi-agent design framework for smart house and home automation applications. They propose a BDI model for agent individual behavior design and a regulation policy-based method for multi-agent group behavior design. Also, a Petri-net based method is developed for system evaluation and analysis. Loseto et al. [8] propose a flexible multi-agent approach for smart-environments based on semantic resource discovery and orchestration. They also include negotiation techniques between user agents and the agents that represent the devices. However, as in the majority of approaches, the user agents are assumed that all of them play the same role. Bajo et al. [1, 9] present a previous version of the THOMAS architecture applied to a similar scenario used in this paper. Our proposal presents a set of THOMAS [5] extensions that give support to the dynamic management of virtual organizations. These extensions are implemented and provided by Magentix2 [11]. Moreover, Magentix2 includes a trace functionality in order to keep track of the activities inside organizations, which is also used in our proposal.

In this work, we propose a MAS based on organizations where different profiles for inhabitants can be defined. The proposed MAS is based on the agent platform Magentix2. Magentix2 provides concepts such as virtual organizations and norms that facilitate the management of multiple occupancy in smart-home environments. The virtual organizations provide the possibility of defining a set of norms that facilitates the regulation and control of the actions that can be carried out depending on the type of inhabitant.

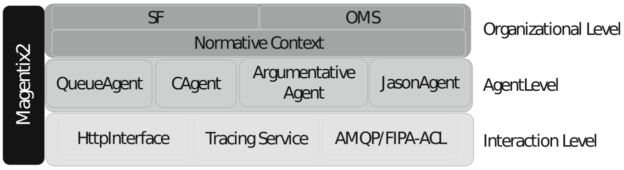

Fig. 1 Magentix 2 platform: components and services

3 Agent Platform: Magentix2

Magentix2[1] is an agent platform for open MAS which provides support at three levels (see Fig. 1):

Organizational level. It integrates the THOMAS framework [13] to provide a complete support for virtual organizations and SOA-like services.THOMAS offers a set of modular services provided by two main components, the Service Facilitator (SF) and Organization Manager Service (OMS). The SF offers a yellow/green

[1] http://www.gti-ia.upv.es/sma/tools/magentix2/index.php.

page service. The OMS is mainly responsible of the management of the organizations. Different types of virtual organizations are supported. Each organization can contain others organizations. Furthermore, it is possible to define diverse roles related to each organization. These roles are characterized by some attributes (position, accessibility and visibility), that can restrict the access to the services offered by the OMS. Also, it is possible to define a normative context to further restrict (or to permit) the access to the THOMAS services.

Interaction level. The platform supports flexible interaction protocols and conversations, indirect communication by means of a tracing service and interactions among agent organizations.

Agent level. Developers have different classes of agents available. For example, in the smart home context, it should be suitable to use two of them: CAgent (automatic creation of simultaneous conversations) and JasonAgent (BDI agents which can participate in simultaneous conversations).

3.1 Normative Context of the Organizational Level

There are some predefined norms which control the access to OMS services. Some OMS services are only available if agents which request them play a certain role inside the organization (Table 1). In this way, users can add PERMITTED or FORBIDDEN norms to relax or restrict the access control to the OMS services in an organization by means of the *registerNorm* service of the OMS. The OMS agent is responsible for verifying if a norm applies before provide a service. In particular, the order in which the restrictions and norms are checked before providing a service of the OMS is as follows: (i) PERMITTED norms, if one is fulfilled, the service is provided with no restrictions; (ii) FORBIDDEN norms, if one is satisfied, the service is not provided; and finally (iii) STRUCTURAL NORMS, that is, all predefined norms of the service are checked before providing the service. If no one is satisfied, the service is provided as usual.

Table 1 OMS service access of some of its available services taking into account the role position played by the requesting agent

	Position			
OMS services	Creator	Member	Supervisor	Subordinate
RegisterUnit	x	-	-	-
RegisterNorm	x	x	x	-
AcquiereRole	x	x	x	x (1)
AllocateRole	x	x	x	-
DeallocateRole	x	x	x	-
InformAgentRole	x	x	x	x

(1) The agent could acquire another role with posistio = creator or position = supervisor in its organization

Norm Description Norms are registered into an organizational unit with a unique name inside that organization. They should be written following the syntax of the THOMAS normative language (see [7] for a detailed explanation), which is based on AgentSpeak language. Concretely, the appearance of a norm is as follows:

$$@normName[Deontic, Target, Action, Activation, Expiration]$$

where:

- *Deontic* $\in \{f, p\}$, where f is used for forbidden norms, which restricts the access to services; and p is used for permitted norms, which relaxes the access to services.
- *Target* $=< type, id >$ where *type* $\in \{agentName, roleName, positionName\}$, whereas *id* is the associated value (for example, the position value *creator*) or the anonymous variable "_" (when any value is accepted). The field *target* allows users to determine which agents will be affected by a norm.
- *Action* is the name of the service. THOMAS only manages norms which their action corresponds to an OMS service.
- *Activation* is a well-formed formula expressed by means of first-order predicates which indicates the conditions to fulfill a norm. Users should add predicates related to data known in the THOMAS world (role details, played roles, organization structure, agent names, etc.). The *Activation* could be empty ("_"), in that case the norm is always fulfilled.
- *Expiration* is a well-formed formula expressed by means of first-order predicates which indicates when a norm expires. So, if the expiration of a norm is satisfied, the norm is not applied although the activation is fulfilled. Users should add predicates related to known data in the THOMAS world. The *Expiration* could be empty ("_"), in that case the norm never expires.

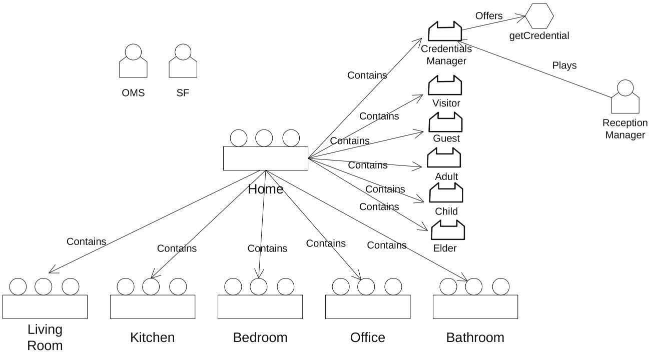

Fig. 2 Organizational view of a smart home

4 Applying Magentix2 to Smart-Homes

In order to illustrate the use of Magentix2 in a smart environment, we have considered smart-home an scenario where the smart-home is modeled as a service-oriented MAS based on virtual organizations. In the following sections, we describe the organizational view, the normative context, and the system dynamics of our proposal.

4.1 Organizational View

The organizational view describes the components of the system and their relationships. In this view, we define the agents, the organizational units, and the roles defined inside these organizational units. The organizational view of the MAS for smart-homes consists of the following entities (see Fig. 2):

- Agents: there is an agent that plays the organizational manager role *OMS* and an agent that plays the service facilitator role *SF*.
- Organizational Units (OU): we consider an organization unit called *Home*. Inside *Home OU*, we define a organizational unit for each of the rooms of the physical home.
- Roles: at the *Home OU* there is defined a set of organizational roles. These roles define the profiles that can be played by the agents inside the *Home OU*. We have considered a role for each type of inhabitant of the living space. The roles *Adult*, *Child*, and *Elder* represent the usual inhabitants of our smart home scenario. *Visitor* and *Guest* represent eventual inhabitants. Finally, the role *Credentials manager* represent the profile of an agent that manages who enters in the *Home OU* and which role must acquire an external agent to be a member of the organization.

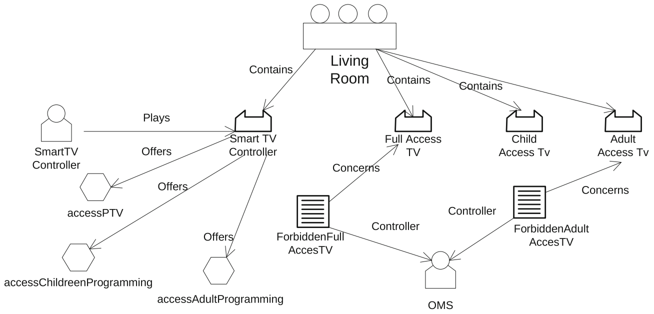

Fig. 3 Organizational view of the living room

Figure 3 describes with detail the *LivingRoom OU*. Inside the *LivingRoom OU* there is a set of roles defined. The *smartTVcontroller* role offers a set of services related to available TV programs in the TV. The roles *FullAccess* and *ChildAccessTv* are defined in order to control the access to the TV programs.

4.2 Normative Context

There is a set of norms that controls the acquirement of the required roles to access to the different services offered by the smart home. As an example, Fig. 4 shows the norm called *forbiddenAcquiereRole*. This norm is registered in the *Home OU* and prevents agents from acquiring roles directly. Thus, they only can acquire roles (credentials) through the *credendialsManager* agent. This norm is described using the Magentix2 normative language.

```
@forbiddenAquireRole[f, <agentName:_>, acquireRole(_,home,_),_,_]
```

Fig. 4 Forbidden norm registered in the *Home OU* that prevents agents from acquiring roles directly

In a similar way, Fig. 3 shows a set of norms that controls which agents can play the roles defined in the *Living Room OU*. Specifically, the norm *forbiddenFullAccessTV* (Fig. 5) avoids that agents which play *child* and *visitor* roles in *Home OU* buy films, because they cannot obtain the role required (Fig. 6).

```
forbiddenFullAccessTV[f,<agentName:Agent>,
acquireRole(fullAccessTV,livingRoom,Agent),isAgent(Agent)&
not playsRole(adult,home,Agent),_]
```

Fig. 5 Norms of the Living Room

4.3 System Dynamics

In order to illustrate the MAS functionality and how agents interact in the system, we present different scenarios that may occur in our smart-home context.

(a) Access to the Smart Home. An external agent a_i wants to get into the smart home system. The entry point is the *Home OU*. To be a member of the organization, the agent a_i must play at least one of the roles defined inside of *Home OU*. In order

to acquire one of these roles, the external agent contacts with a *ReceptionManager* agent that is inside the organization and plays the role *CredentialsManager*. The *ReceptionManager* agent offers a service (i.e., *getCredentials*) that allocates roles to newcomers based on the information contained in their profiles. The *ReceptionManager* interacts with the OMS agent to register members in the organization. After that, the newcomer agent is part of the *Home OU*.

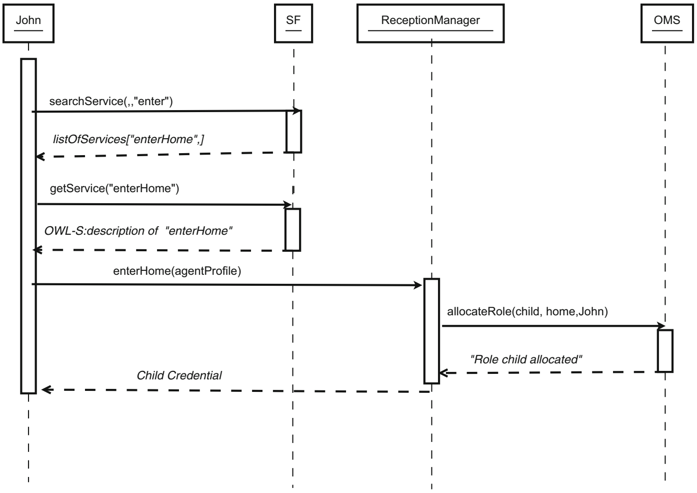

Fig. 6 Enter Home Scenario: a child named John gets his corresponding credential

(b) Asking for a service that you are not allowed to ask. An agent a_i that plays the role *Child* in *Home OU* enters in the organization *LivingRoom OU*. The agent wants to request a service in order to buy a film using the smart TV. The agent a_i asks the SF agent for a service of films and gets the information about services that offer pay-per-view movies and which are the provider agents. The agent a_i selects the Smart TV provider agent and asks it for the service. In order to provide the service to the agent a_i, the smart TV agent checks if a_i plays the required role. This verification is done by asking the OMS about the role that plays the requester. Based on this information the smart TV agent offers or not the service (Fig. 7).

(c) Asking for a service that you are allowed to ask. An agent a_i that plays the role *Adult* in *Home OU* enters in the *LivingRoom OU*. The agent a_i wants to request a service in order to buy a film. Therefore, it asks the SF agent for a service of films and then it gets the information about the available services and the agents that provide them. The agent a_i analyzes the information about the service profile of the provider agents and notices that to ask for the service, it is necessary to play the role *FullAccessTV*. Therefore, the agent a_i interacts with the OMS agent in order to acquire this role. Once the agent a_i plays this role, it asks the Smart TV agent for the service.

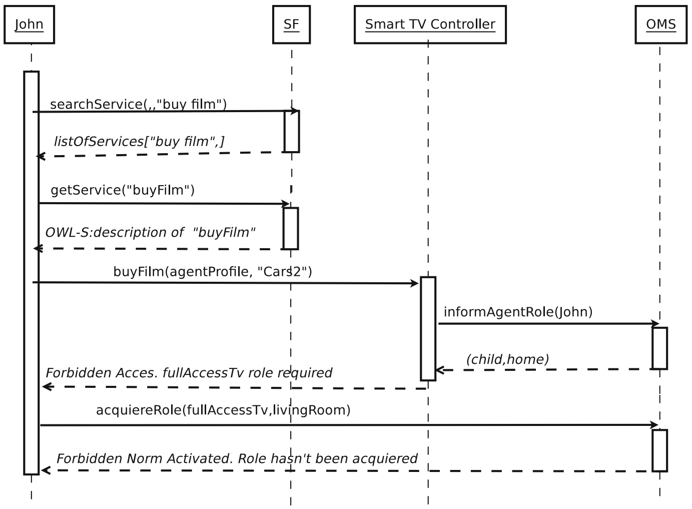
John
SF
Smart TV Controller
OMS
searchService(,,"buy film")
listOfServices["buy film",]
getService("buyFilm")
OWL-S:description of "buyFilm"
buyFilm(agentProfile, "Cars2")
informAgentRole(John)
(child,home)
Forbidden Acces. fullAccessTv role required
acquiereRole(fullAccessTv,livingRoom)
Forbidden Norm Activated. Role hasn't been acquiered

Fig. 7 Not Allowed Service Scenario: a child named John tries to access to the buyfilm service

(d) Access of a visitant (medical assistant). An external agent a_i, that represents a medical assistant, wants to get into the smart home system in order to control the health of one of the inhabitants. The entry point is the *Home OU*. To be a member of the organization, a_i must play at least one of the roles defined inside the *Home OU*. In order to acquire one of these roles, the external agent a_i contacts with the *ReceptionManager* agent that is inside the *Home OU*. As the external agent a_i has in its profile its occupation, the controller agent allocates the role *Visitor* and the role *HealthStaff*. The *ReceptionManager* agent interacts with the OMS agent for the registration process of the new member of the organization. Once the agent a_i is part of the organization and plays the role *HealthStaff* and *Visitor*, it asks the SF about services related to medical control (blood pressure, diabetic information, insurance information,...). With this information, the agent a_i can interact with the agents that provide the services in order to create an inform about the current situation of a home inhabitant (Fig. 8).

(e) Trace of events in the Smart Home An agent would be interested in tracing some activities performed by other inhabitants. For instance, the agent a_i that plays the role *Adult* can ask for tracing the services associated to the activity of other agent that plays the role *Elder* inside the *Home OU*. With this trace, the agent a_i controls if the medical assistant arrived at home. The trace can be also useful for controlling the activity of other inhabitants. For instance, if an agent that plays the role *Child* has tried to ask for a service that it is not allowed for agents that play the *Child* role.

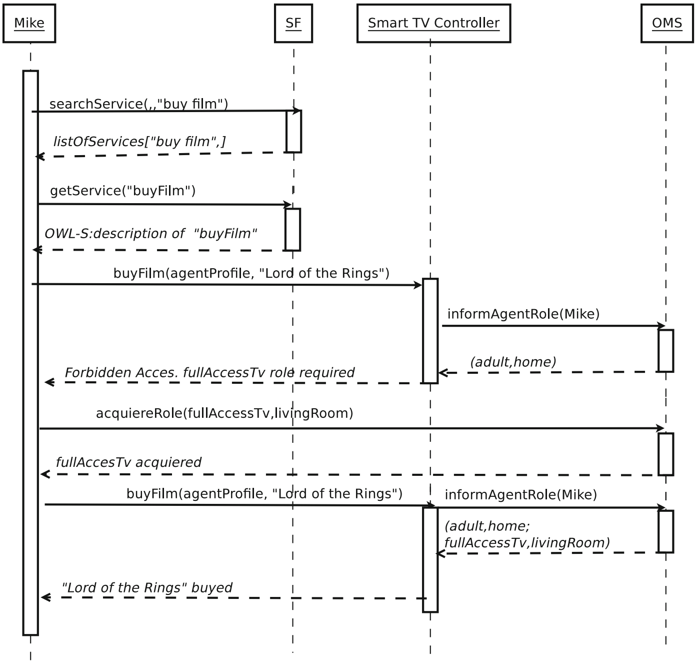
Mike
SF
Smart TV Controller
OMS
searchService(,,"buy film")
listOfServices["buy film",]
getService("buyFilm")
OWL-S:description of "buyFilm"
buyFilm(agentProfile, "Lord of the Rings")
informAgentRole(Mike)
(adult,home)
Forbidden Acces. fullAccessTv role required
acquiereRole(fullAccessTv,livingRoom)
fullAccesTv acquiered
buyFilm(agentProfile, "Lord of the Rings")
informAgentRole(Mike)
(adult,home;
fullAccessTv,livingRoom)
"Lord of the Rings" buyed

Fig. 8 Allowed Service Scenario: an adult named Mike asks for the buyfilm service

5 Conclusions

In this paper, we have presented a MAS for smart-home environments based on the Magentix2 platform. Our approach facilitates the management of multiple inhabitants in a living space. Through the use of organizational concepts, it is possible to control the activities of internal and external agents in an efficient way. We illustrate the applicability of our proposal through a set of scenarios in the context of smart-homes. As future work, we will extend Magentix2 to provide security services and consider norms in the Service Facilitator services. Moreover, we will compare empirical results of our proposal with other agent platforms that allow the definition and execution of norms.

Acknowledgments This work is supported by the Spanish government grants CONSOLIDER INGENIO 2010 CSD2007-00022, MINECO/FEDER TIN2012-36586-C03-01, TIN2011-27652-C03-01, and SP2014800.

References

1. Bajo J, Fraile JA, Pérez-Lancho B, Corchado JM (2010) The thomas architecture in home care scenarios: a case study. Expert Syst Appl 37(5):3986–3999
2. Cetina C, Giner P, Fons J, Pelechano V (2009) Autonomic computing through reuse of variability models at runtime: The case of smart homes. Computer 42(10):37–43
3. Cook DJ (2009) Multi-agent smart environments. J Ambient Intell Smart Environ 1(1):51–55
4. Crandall AS, Cook DJ (2010) Using a hidden markov model for resident identification. In: 6th international conference on intelligent environments, pp 74–79. IEEE
5. Criado N, Argente E, Botti V (2013) THOMAS: an agent platform for supporting normative multi-agent systems. J Logic Comput 23(2):309–333
6. Davidoff S, Lee MK, Zimmerman J, Dey A (2006) Socially-aware requirements for a smart home. In: Proceedings of the international symposium on intelligent, environments, pp 41–44
7. Grupo de Tecnología Informática e Inteligencia Artificial (GTI-IA) (2015). http://www.gti-ia.upv.es/sma/tools/magentix2/archivos/Magentix2UserManualv2.1.0.pdf. Magentix2 User's Manual v2.0
8. Loseto G, Scioscia F, Ruta M, di Sciascio E (2012) Semantic-based smart homes: a multi-agent approach. In: 13th Workshop on objects and agents (WOA 2012), vol 892, pp 49–55
9. Rodriguez S, Julián V, Bajo J, Carrascosa C, Botti V, Corchado JM (2011) Agent-based virtual organization architecture. Eng Appl Artif Intell 24(5):895–910
10. Rodríguez S, Paz JFD, Villarrubia G, Zato C, Bajo J, Corchado JM (2015) Multi-agent information fusion system to manage data from a WSN in a residential home. Inf Fusion 23:43–57
11. Such JM, Garca-Fornes A, Espinosa A, Bellver J (2012) Magentix2: a Privacy-enhancing Agent Platform. Eng Appl Artif Intell
12. Sun Q, Yu W, Kochurov N, Hao Q, Hu F (2013) A multi-agent-based intelligent sensor and actuator network design for smart house and home automation. J Sens Actuator Netw 2(3):557–588
13. Val E, Criado N, Rebollo M, Argente E, Julian V (2009) Service-oriented framework for virtual organizations. 1:108–114
14. Wu C-L, Liao C-F, Fu L-C (2007) Service-oriented smart-home architecture based on osgi and mobile-agent technology. IEEE Trans Syst Man Cybern Part C Appl Rev 37(2):193–205
15. Yin J, Yang Q, Shen D, Li Z-N (2008) Activity recognition via user-trace segmentation. ACM Trans Sens Netw (TOSN) 4(4):19