An approximation to the features of smart home prediction algorithms

Fernández-Montes, A.; Cruz, M.D.; Álvarez, J.A.; Ortega, J.A.; Torres, J.; Velasco, F.; González, L.
Departamento de lenguajes y sistemas informáticos.
ETSI Informática
Avda. Reina Mercedes s/n, Sevilla
{alejandro.fdez, manuel3086}@gmail.com, {juan, ortega, jtorres}@lsi.us.es, {velasco, luisgon}@us.es

Abstract

This paper reviews the goals of the Domoweb project and the solutions adopted to achieve them. As a result we enjoy a great support to develop smart home techniques and solutions. As a consequence of the acquired experiences a Smart home model is proposed as a division of four main categories. In relation with the smart home model, we show the essential features a smart environment prediction algorithm should satisfy and a procedure to select relevant information based in a window of state from the model to achieve artificial intelligence based solutions. Finally we propose a technique of situation recognition in a smart home environment.

1 Introduction

Smart home technologies are often included as a part of ubiquitous computing. Mark Weiser [Weiser, 1991] outlined some principles to describe Ubiquitous Computing (ubicomp) from which we emphasize that the purpose of a computer is to help you do something else.

Home technologies have tried to help home inhabitants since its creation. Nowadays, due to the popularization of computational devices, ubiquitous computing is called to be the revolution to develop smart systems with artificial intelligence techniques.

Domoweb [Álvarez et al., 2005] is a research project which was originally developed as a residential gateway implementation over the OSGi (Open Services Gateway Initiative) service platform. Domoweb implements the standard services any residential gateway must have like http server, web interfaces, device manager, user manager and other basic services.

Nowadays Domoweb conform a great platform where researchers from different disciplines converges and where we can deploy, develop and test smart home related solutions, due to the component based model, and the service oriented architecture that Domoweb and OSGi supports.

This article focuses on modeling smart homes and the features the prediction algorithms should implement. A good smart environment model must represent the properties, states, attributes and any other characteristic that could be useful in building smart environments solutions as is proposed in section 3.

Artificial intelligent methods can be supported by this model like prediction algorithms where is centered this article. The features these algorithms must be aware of are defined in section 4. In section 5 we examine a procedure to discriminate significant information with a window of states. Finally in section 6 a technique for recognition of situations is proposed as a start point of this line of research.

2 Domoweb

The main goal of the Domoweb project was to design and develop a smart home system controlled via web and open source based. This objective means that the home system must be able to manage typical smart home elements like alarms, comfort control, users and other system by means of necessary sensors and appliances.

One of the features of the system is its capacity of being managed through adaptable interfaces so users can control the smart home from a huge variety of devices. This control usually involves interactions with different technologies like Bluetooth, USB, or X10 so a special effort was necessary to communicate homogeneously with different technologies. Next we discuss different design and implementation decisions.

2.1 OSGi Service Platform implementation

Domoweb is based on open source implementations so we decided to use any of the available implementations. In particular we used two reference implementations: OSCAR[1] (currently known as Apache-Felix) and Knopflerfish[2]. Both platforms are component and services oriented and show a GUI where developers can deploy and manage components in an independent way.

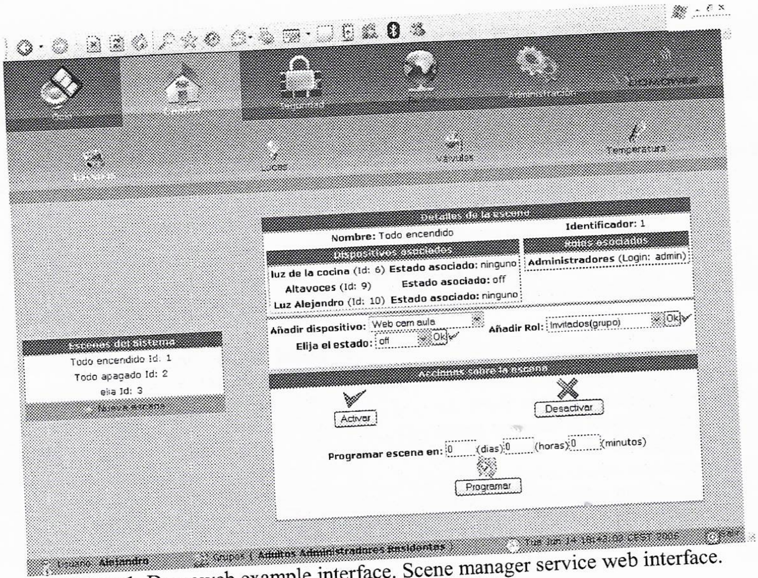
2.2 Infrastructure services

OSGi Service Platform proposes specifications of some basic services any gateway should implement. User administration, configuration administration and http server are good examples of these essential services.

---

[1] http://oscar.objectweb.org/

[2] http://www.knopflerfish.org/



Nombre: Todo encendido
Identificador: 1

luz de la cocina (Id: 6) Estado asociado: ninguno
Altavoces (Id: 9) Estado asociado: off
Luz Alejandro (Id: 10) Estado asociado: ninguno

Administradores (Login: admin)

Añadir dispositivo: Web cam aula
Elija el estado: off
Añadir Rol: Invitados(grupo)

Todo encendido Id: 1
Todo apagado Id: 2
elia Id: 3

Programar escena en: 0 (dias) 0 (horas) 0 (minutos)

Usuario: Alejandro
Grupos: { Adultos Administradores Residentes }
Tue Jun 14 18:43:02 CEST 2005

Figure 1. Domoweb example interface. Scene manager service web interface.

Other infrastructure services have been implemented to achieve Domoweb project goals. On the next listing we expound the most relevant infrastructure services:

- **System Information Service**. Help other applications to abstract from the database layer.
- **Device Manager Service**. Offers an independent of the technology and easy-to-use service to interact with installed devices
- **Start-up service**. Manages installation of all the services in a coordinated way.

2.3 Application services

Application services are defined as those services which offer any functionality to the end user. These services were also supported by some infrastructure services like those shown before and are usually implemented as a final web interface for infrastructure services and often implement typical smart home applications like:

- **Scene manager** service which offers an intuitive way to build predefined status for devices as shown in figure 1.
- **Home Configuration manager** a service where you can report a description of your home and devices layout using a XML language called *Lenguaje de Descripción de Casas* (LDC) [Álvarez et al., 2005].
- **Security** services like intrusion detection with cameras and specific appliances, presence simulation useful when nobody is at home.
- **Events register**. This service is the responsible for detection and storage of the events occurred at the smart home environment. This has a close relationship with learning techniques and prediction algorithms as we discuss in sections 4 and 5.

All these experiences have been useful to develop the necessary skills to face more complex services and ambient intelligence applications.

3 Smart home model

Artificial intelligence algorithms need a solid base of knowledge to work. This fact demands us a great effort for building a model of the smart home and its environment. Other projects have helped us to compose the model [Cook et al, 2006; Li et al., 2006; Das et al, 2005] which has been arranged in four main categories explained in next sections and shown in table 1.

3.1 Device Related

This category is the most obvious one so is related with the main elements in a smart home environment. Ambient intelligence algorithms should be aware of next main fields:

- **Status**. Algorithms must know the current states of devices installed on the smart home. Obviously this is essential for these algorithms, and one of the most important domains to build future predictions.
- **Location**. Devices usually occupy a location for a long time and this location may be useful for ambient intelligence algorithms. The model must be able to consider non-still devices as well, like motorized cleaner robots and others.

Categories	Fields	Description	Example
Device related	Status	Current state the devices are.	Sensor temperature is at 25°C
	Location	Where the devices are.	Cleaner robot is in Living room.
Inhabitants related	Personal data	Name, Age, sex.	Diane is 45.
	Location	Where the inhabitants are.	Mark is at bedroom.
	Physical state	Illness, injuries, and others states inhabitants are.	Roy has a cold.
	Mental state	Psychological state the inhabitants are.	David is depressed.
Environment related	Date, time	Temporary information	Current time is 13:36.
	Environmental conditions	the phenomena that currently occur in the atmosphere	It's rainy
Home background	Inert entities location	Where these entities are.	Sofa is at living room.
	Home limits properties	The properties of the home structure and limits.	Bedroom window opacity is 70%

3.2 Inhabitants related

Smart home algorithms must be aware of the inhabitants' status to offer appropriate predictions for any user or for the whole group of inhabitants. On this line we discuss some of the necessary fields to infer inhabitants-aware predictions:

- **Personal data**. This field includes all the data concerning to a particular person like name, age, sex and so on.
- **Location**. Inhabitants can move over the home rooms, so smart home systems have to know where each inhabitant is, and should be able to identify them.
- **Physical state**. This field is related with the illness or injuries that an inhabitant can suffer during his life. Smart home technologies must adapt to this situations and offer appropriate replies.
- **Mental state**. The state of mind of a person can be defined as a temporary psychological state. A depressed inhabitant behaviour usually differs from a euphoric one, so smart home must be consistent with these circumstances.

3.3 Environment related

This category probably is the most diffused due to it covers heterogeneous and difficult-to-limit fields as we discuss in the following list:

- **Date, time, season**. Obviously smart home behavior should be different in different temporal conditions and it may depend in these temporal factors as the air condition policies will differ between summer and winter.
- **Environmental conditions**. This field comprises current environmental conditions (sunny, cloudy, rainy and others). Smart home should make a request for a weather forecast as well, which could be significant to assess future decisions.

3.4 Home background

This category must comprise all the relevant things concerned to inert entities and its properties or qualities. This could be the less relevant category discussed, but anyway could be significant in some concrete applications. We propose a couple of fields related in next listing:

- **Furniture location and position**. Furniture occupies space at home and can be moved. Location (room where the furniture is) and position (place where the furniture is placed in a room) should be known by the smart home systems due to could be useful by concrete applications like robot movement related algorithms, or presence detection related algorithms.
- **Home limits properties**. The texture of a floor, the color of a wall, or the opacity of the windows could be significant in several cases such as temperature adjustment applications.

4. Features of prediction algorithms

In this section we present the features that we consider a smart home system must implement, specially related with the prediction algorithms the systems may have. The article doesn't focuses on artificial intelligence techniques like the studies of [Das et al, 2005; Leake et al, 2006; Hagras et al, 2004; Jonghwa et al, 2005; Nirmalya et al, 2006; Yamazaki, 2006; Jiang et al, 2004] do, but on what are the most important and indispensable features that must be considered to develop prediction algorithms. Much of the ideas presented below could be useful to implement others smart home algorithms.

4.1 Prediction supported by last events and states

Prediction algorithms should consider as input data two main aspects. First it should analyze the last events occurred in the home's performance field which have changed the home status in any manner.

Second it should analyze current state and previous states as well. This way the algorithms should determine last changes in the home status, and which events have been involved in these changes.

In section 5 we discuss further about these aspects and propose a way to consider states and events.

4.2 Predictable by the inhabitants

Smart homes should learn inhabitant behaviours and habits, and build predictions, but it has to be predictable by the inhabitants to achieve no unexpected behaviours.

4.3 Understandable decisions

Prediction algorithms have to offer an explanation of their predictions and/or actions. This way the inhabitants will be more trusted with the decisions the system took. This will improve the user acceptance of the smart home predictions.

4.4 Wrong decisions detection and related improvement

If the smart home executes a wrong decision it should be aware of this failure and be able to learn from its errors. A possible scenario should be when someone arrives home at night and the system orders the hall lights to switch on, but immediately the user performs the opposite action (which should be switch off the lights). The system has to notice this failure and extract some knowledge from this experience to face future similar situations with guarantees of success.

4.5 Anomalies detection

In some scenarios, the smart home should consider that a wrong decision executed as a consequence of a wrong prediction could be produced due to an anomaly. We can consider this scenario, an inhabitant wake up all working days at 7.00 a.m., so smart home switch on the coffee-maker some minutes before. But when the wake-up alarm goes off, the smart home detects no movement so it could be desirable to request an emergency service with a standard phone call or other mechanism.

4.6 Quick response when required

Some situations require the smart home prediction algorithms to return a response with time limits. These algorithms are responsible for detection of this situations and they must be able to adapt to these circumstances in order to provide a quick response.

4.7 General policies and user adaptation

Smart homes technologies should implement the mechanisms to support some home policies like security, energy or comfort. These policies can be collected in different levels. The inhabitants could define some general policies which could be customizable by concrete inhabitant preferences. We can discuss the following scenario; inhabitant A gives preference to energy consumption over comfort. To satisfy these preferences smart home system should try to minimize the energy consumption produced as a result of its predictions. On the other hand, inhabitant B gives preference to comfort over energy consumption, so algorithm has to adapt to this user preferences policy.

This way if the prediction algorithm determines that inhabitant A arrives home, it will switch on air conditioner system only at the arrival of the inhabitant, however if the inhabitant B is going to arrive home, the smart home should switch on the air conditioner system sometime before the arrival of the inhabitant.

5 Window of states

Smart home prediction algorithms usually make use of last states and events occurred at home environment. A state can be defined as the whole set of pairs field-value according to the model presented in section 3. We could also distinguish subsets of these states for each category so we shape the related sub-states. This way we can consider sub-states to include devices related, inhabitants related, environment related and background related information.

Events can be defined as something that happen at a given place and time. In smart home contexts, we can add to this definition that the event must cause a state change of the smart home. Events that do not cause a state change shouldn't be considered by smart home algorithms as something significant.

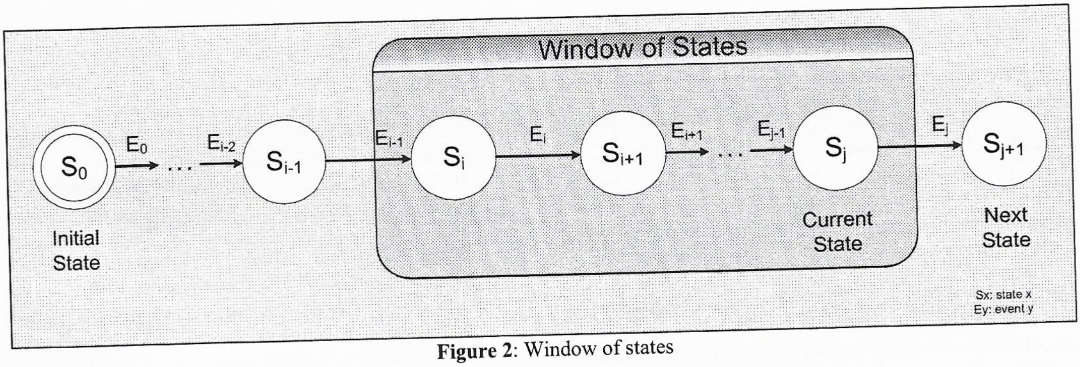


Figure 2: Window of states

When building a smart home solution we can discuss what states and events should be considered. To resolve this situation we propose the use of a window which envelops the states (and events) that are going to take part in the giving response as shown in figure 2.

States are represented as circumferences labelled from S_0 to S_{j+1}, transitions between states as arrows labelled with the event which caused the state change.

Reader should notice that states from S_0 (initial state) to S_{j-1} are past time states, S_j is the current state and S_{j+1} is the next state which is currently unknown.

The size of the window ***n*** equals to the number of states and events to be processed. This size is deduced from the equation ***n = j-i+1***.

5.1 Dynamic vs. Static window.

Two approaches can be considered related with the size of the window. Algorithms designer could prefer the use of a static and predefined size of this window to build smart predictions. This approach can be useful in non-complex solutions, so it's easy to implement. The size could be estimated by means of empirical techniques and the own designer experience.

However dynamic window size is a more powerful approach for determining the size in real time. Algorithms could extend and reduce the size of the window depending on the requisites of current application like time of processing, hardware capabilities, solution accuracy and others.

6 Situation recognition

A situation is defined as a set of states which have similar values. This set can be defined by:

- **a canonical state** as a representative state of all the states in the set
- **a distance function** which define how to measure the difference between two states
- **a threshold** which define the maximum distance between the canonical state and the states of the set.
- **a weight for each element of a state** between [0,1]. A weight of *zero* represent that an element must not be considered.

The following example illustrates a definition of a situation. Let x_1 be the variable which represents the state of a light, and let x_2 be the variable which represent the state of another light. These variables are discrete, and its range is {0.1} which represent the states *off* and *on* respectively.

The following figure shows all the states that these variables can constitute.

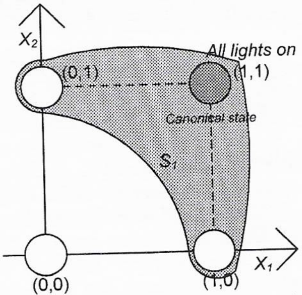


Figure 3: States for variables x_1 and x_2

We would like to define the situation S_1 where *any of the lights is on*. The canonical state of S_1 is {1,1}(all lights on) for the variables x_1 and x_2 respectively. A typical distance function could be defined as:

$$d([x_1,x_2],[y_1,y_2],[w_1,w_2]) = \\ = \sqrt{(x_1-y_1)^2 * w_1 + (x_2-y_2)^2 * w_2}$$

Where $[x_1,x_2]$ represent a state, $[y_1,y_2]$ represent the canonical state and $[w_1,w_2]$ represent the weights of x_1 and x_2 respectively which should be [1,1] for the situation S_1 *any of the lights is on* due to both lights should be considered. The threshold can be set to 1.

The table 2 shows the set of states that belong to the situation. If the distance is less or equal to the threshold, the state belongs to the situation. In other cases we can conclude the state will not belong to the situation

state		**canonical state**		**weights**		**d**	**threshold**	**$\subset S_1$**
x_1	x_2	y_1	y_2	p_1	p_2			
0	0	1	1	1	1	$\sqrt{2}$	1	no
0	1	1	1	1	1	1	1	yes
1	0	1	1	1	1	1	1	yes
1	1	1	1	1	1	0	1	yes

Table 2: states that belongs a situation

7 Conclusions and future work

In previous sections have been discussed some essential ideas in order to get an approach to real smart home solutions. The model proposed in section 3 could be a good starting point to reach this aim. This model is useful to build solid knowledge bases which help us developing prediction algorithms.

Proposed ideas shape a general research in smart home solutions. Future work must be centred in concrete, simple and real cases where we will be able to test and check the behaviour of the smart home model applied within prediction algorithms and its knowledge bases.

We are interested in expanding the situation recognition section including a deeper research in data homogenisation proposing different options like typification or normalization.

Another topic where we want to study in depth is the creation of the predefined rules where prediction algorithms can be supported.

Acknowledgments

This research is supported by the MEC I+D project In-Care. Ref: TSI2006-13390-C02-02 and by the project CUBICO Ref: TIC2141.

References

[Weiser, 1991] Mark Weiser. The computer for the 21st century. *Scientific American*. 1991.

[Álvarez et al., 2005] J.A. Álvarez, M.D. Cruz, A. Fernández, J.A. Ortega, J. Torres. Experiencias en entornos de computación ubica mediante arquitecturas orientadas a servicios. In *JSWEB 2005 (W3C).*

[Cook et al, 2006] Diane J. Cook, Michael Youngblood, and Sajal K. Das. A Multi-agent Approach to Controlling a Smart Environment. *Designing Smart Homes*, 165–182, 2006.

[Li et al., 2006] Jun Li, Yingyi Bu, Shaxun Chen, Xianping Tao and Jian Lu. FollowMe: On Research of Pluggable Infrastructure for Context-Awareness. *Proceedings of the 20th International Conference on Advanced Information Networking and Applications IEEE* 2006.

[Das et al, 2005] Sajal K. Das and Diane J. Cook. Designing Smart Environments: A Paradigm Based on Learning and Prediction. *PReMI 2005, LNCS*, 2005.

[Leake et al, 2006] David Leake, Ana Maguitman, and Thomas Reichherzer. Cases, Context, and Comfort: Opportunities for Case-Based Reasoning in Smart Homes. *Designing Smart Homes*, 109-131. 2006

[Hagras et al, 2004] Hagras, H., Callaghan, V., Colley, M., Clarke, G., Pounds-Cornish, A., Duman, H. Creating an ambient-intelligence environment using embedded agents. *IEEE Intelligent Systems* 2004

[Jonghwa et al, 2005] Jonghwa Choi, Dongkyoo Shin, and Dongil Shin. Research on Design and Implementation of the Artificial Intelligence Agent for Smart Home Based on Support Vector Machine. *ICNC 2005, LNCS.* 2005.

[Nirmalya et al, 2006] Nirmalya Roy, Abhishek Roy, Sajal K. Das. Context-Aware Resource Management in Multi-Inhabitant Smart Homes:A Nash H-Learning based Approach. *IEEE PERCOM'06. 2006.*

[Yamazaki, 2006] Tatsuya Yamazaki. Beyond the Smart Home. *ICHIT'06*. 2006.

[Jiang et al, 2004] Li Jiang, Da-you Llu, Bo Yang. Smart home research. *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, Shanghai, 26-29. 2004