A spoken interface based on the contextual modelling of smart homes

Pablo A. Haya and Germán Montoro

Escuela Politécnica Superior
Universidad Autónoma de Madrid
28049 Madrid, Spain
{Pablo.Haya, German.Montoro}@uam.es

1 Introduction

Ubiquitous computing, also-called pervasive computing, is an emerging technology that offers new opportunities and challenges [14]. We are especially interested in those that have an impact in home environments. In particular, we are focusing on how context information can enhance user interaction within a smart home environment.

We propose that the context gathered from the environment should be collected in a common model shared by every context-aware application [1]. This model should include the available resources and the relations among them. In this direction, we have implemented a middleware between the model of the smart home and the physical world in such a way that changes in the model are immediately reflected into the real world, and vice versa.

There are several groups researching in how to model the context as a web of relations among concepts, such as the Cobra project [4], Henricksen et al [8] and the Aire project [15]. Our proposal is specifically focused on home environments.

A laboratory has been converted into a real home environment to test our prototypes in a similar way as the Adaptive House [11], the Aware Home [9] and The Intelligent Room [10] projects.

Following sections describe a context information model for smart home environments. Every environment component is represented by a model's instance that contains information about its status and its relationships. This information is used by the home applications to react to the changes in the context. In particular, linguistic information is added to the representation of instances in order to support a contextual spoken dialogue interface.

2 Modelling the environment

We have devised a hierarchical classification of the relevant concepts for home environments. This section presents the ontology that entails these concepts. The following sections will explain how this ontology is employed by a spoken dialogue interface.

In the proposed ontology each concept is represented by a class name and a set of properties. Each property has a value that can be a literal or another concept. When the value of a property is a concept, a relation between the two concepts is established. This relation is considered as having an explicit “direction”, that is, in case it holds, the inverse relation must be explicitly asserted.

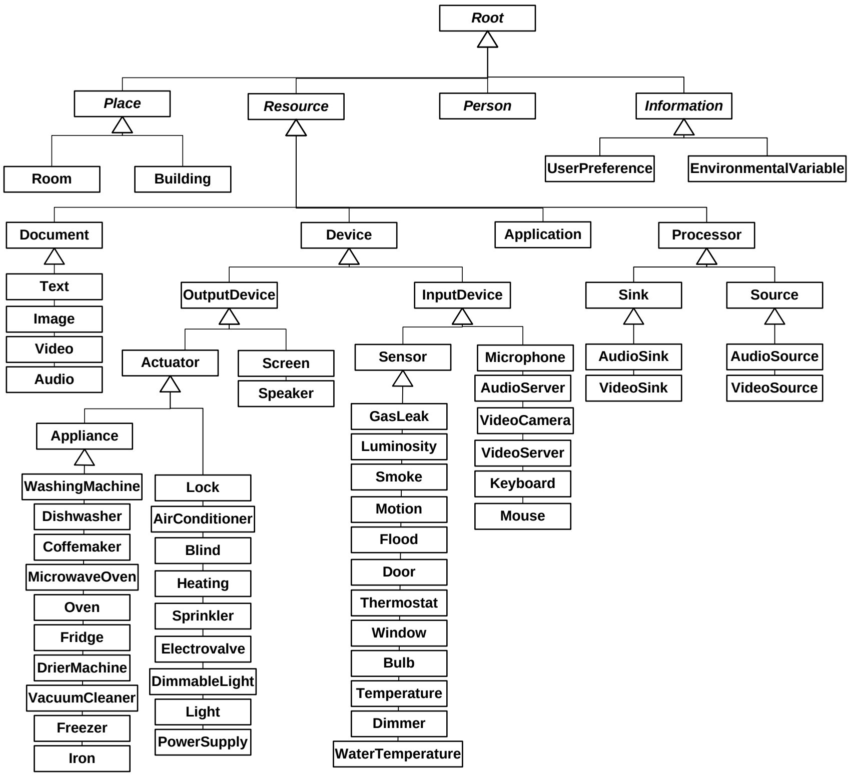

Fig. 1. Global view of the taxonomy for home environments

We have adopted Dey’s definition of context to develop our model for a home intelligent environment [6]. Figure 1 shows the complete hierarchical classification of concepts that we propose for home environments. The model starts with four main concepts: *Person*, *Place*, *Resource* and *Information*. A *Resource* is every component that can be used by a *Person* (or other *Resource*) located in some *Place*. *Information* can be a *User Preference* or an *Environmental Variable*, such as the current sound level or the luminosity. The description of a Resource always includes the following set of relations: *handles*, *is-handled-by*, *is-composed-by*, *allowed-user* and *is-located-in*. The *handles* relation establishes that a resource is being used by other resource. Reciprocally, the relation *is-handled-by* defines which *Person* or *Resource* is controlling a given *Resource*. The *composed-by* property allows to describe a *Resource* as composed of other resources. The *allowed-user* property defines the access policy. Finally, the *is-located-in* property represents where the resource is located.

The *Resource* concept is refined by the following four concepts: *Device*, *Document*, *Application*, and *Processor*. A *Device* represents a physical object (i.e. a microphone, a light bulb, a speaker). Each *Device* always includes the status property that, at least, indicates if the device is turned on or off. We have split the device category into *Output* and *Input device* depending on whether they produce or consume information. *Output devices* include video and audio consumers such as screens and speakers and mechanical actuators such as door locks, blinds, lights and home appliances. On the other side, *Input devices* comprise mice, keyboards, video and audio sources, such as microphone and video cameras, and physical sensors. Device classes showed at figure 1 represent simple devices. It is possible to define composite devices by means of the *is-composed-by* relation. Thus, a TV set is composed by an instance of Screen class and two or more instances of Speaker class. Finally, *Documents* and *Applications*, do not correspond to tangible objects. The first class represents digital files that store some information, while the second class represents computational services. Therefore, devices, documents and applications represent existing resources. On the other side, a Processor denotes a capability, something that can be performed by a resource. This allows, for example, distinguishing between the sensing capability and the sensor itself.

3 Working with the model

The ontological representation of the environment, including its instances, is written in an XML document. At startup, the system reads the document and automatically builds:

– A blackboard [7], which works as an interaction layer between the physical world and the spoken dialogue interface.
– A spoken dialogue interface that, by means of the blackboard, works as an interaction layer between the users and the environment.

This blackboard holds a representation of multiple characteristics of the environment. These characteristics correspond with instances of the previous ontology. Each instance is called an entity. Applications and interfaces can ask the blackboard to obtain information about the state of any entity or to change it. Entities can be added and removed to the blackboard in run-time, and the new information can be reused by the rest of applications. Applications and interfaces do not interact directly with the physical world or between them, but they only have access to the blackboard layer.

This blackboard layer isolates the applications from the real world. Physical world entity details are hidden to clients [13], making easier and more standard to develop context aware modules and interfaces.

Entities are associated to a concept. All the entities related to the same concept inheritance some general properties. This means that if we define a new entity its properties will come attached to it.

Some of the properties associated to the entities represent linguistic information. This information is formed by a verb part (the actions that can be taken with the entity), an object part (the name it can be given), a modifier part (the kind of object entity), a location part (where it is in the environment) and other parts. A set of these parts establishes one possible way a user may employ to interact with the entity. One entity has associated a collection of sets of parts, corresponding to all the possible ways to interact with the entity. A single part can be composed of one or more words, allowing the use of synonyms. Additionally, entities inheritance the name of its associated template grammar and the action method that has to be called after its linguistic information is completed. Action methods are specific for each type of entity and execute all the possible actions that may be requested by a user (for instance to turn on, turn off, dim up and dim down the light in an entity of type *dimmable_light*).

The linguistic information is transformed in specific grammars and dialogue nodes that support the spoken interaction process. Users manage and interact with the environment by means of the spoken dialogue interface and the interface employs the information represented in the blackboard to support the dialogue capabilities.

4 Dialogue representation

As it was said above, the spoken dialogue interface is composed of a set of grammars and a dialogue structure.

Grammars support the recognition process by specifying the possible sentences that can be uttered by the users, limiting the number of inputs expected by the recognizer [5]. This way, users will only be allowed to carry on dialogues related to the current configuration of the environment, not considering other possible utterances. The system creates a grammar for each concept. Grammars are based on the grammar template associated to the concept. In the interface creation process entities only have to fill in the corresponding grammar template with their collection of set parts.

The dialogue structure is based on a linguistic tree. Before creating the dialogue interface the tree only has an empty root node. Every set of linguistic parts is transformed in a tree path, with a node for each part. Nodes hang from parent nodes that represent previous parts of the same set. Nodes store the word corresponding to that part and the name of the entity where they belonged. Parts with more than one word (synonyms) will be transformed in different nodes and following parts of the same set will hang from every one of these synonym nodes.

Words are analyzed by a morphological parser [3] in order to get their number and gender. Repeated words are analyzed only the first time and this information is stored for later use at the generation process.

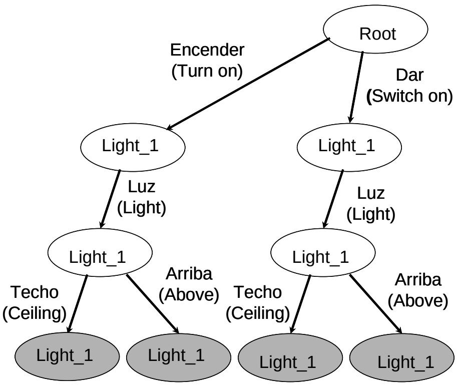

Fig. 2. Partial linguistic tree

As an example, let us suppose that the entity light_1 has the following set of parts: {“encender dar”, “luz”, “”, “techo arriba”, “”}, in English {“turn_on switch_on”, “light”, “”, “ceiling above”, “”}, where the first column corresponds to the verb part, the second one to the object part, the third column to the modifier part, the fourth one to the location part and the last column to the additional information part. “Turn on” and “switch on” are synonyms, the same as “ceiling” and “above”. Therefore this corresponds with four possible ways to interact with the entity light_1. Starting from an empty tree, the system would create the linguistic tree showed in figure 2.

Shadowed nodes correspond to action nodes. When the system reaches one of these nodes the system executes the action method associated to the entity where it belongs (in this case it would turn on the light_1).

Another set of parts may have a word part at the same level as a previous set. In this case the system will not create a new node for that part, but it will reuse that node and will append, if necessary, the name of the entity where it belonged. Let us suppose, for instance, that the entity light_1 has the following two sets of parts: {“apagar”, “luz”, “”, “techo arriba”, “”} and {“apagar”, “fluorescente”, “”, “”, “”}, in English {“turn off”, “light”, “”, “ceiling above”, “”} and {“turn off”, “fluorescent”, “”, “”, “”}, which correspond with three possible ways of interacting with the entity light_1. In this case, the word part “turn off” is at the same level in both sets of parts so that only one “turn off” node is created and “light” and “fluorescent” both hang from it. If now, we have a new entity called radio_1, with this linguistic set of parts: {"apagar", "radio", "", "", ""}, in English {“turn off”, “radio”, “”, “”, “”}, the system only has to append the name of the entity radio_1 to the “turn off” node. Next it adds a “radio” node as its child, at the same level as “light” and “fluorescent”. Starting from an empty tree, the system would create the linguistic tree showed in figure 3.

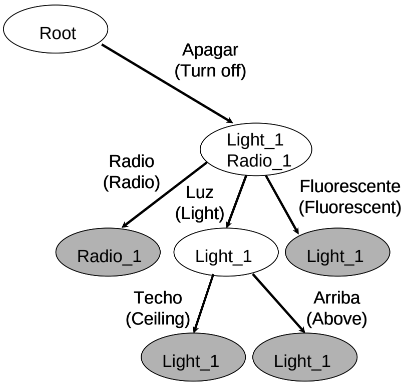

Fig. 3. Linguistic tree for light_1 and radio_1 entities

This automatic process is followed by all the collections of sets of parts of all the entities presented in the smart environment. Once the grammars and the linguistic tree are completed the system is provided with a spoken dialogue interface that supports all the possible ways of interaction for the current smart environment.

5 Conclusions and future work

Current home environments projects require a deployment of a heterogeneous set of technologies. The proliferation of communication networks and protocols complicates the seamless integration of environment devices. As a result, projects are usually developed from scratch, and much time is spent on integration tasks. We propose a standard context layer that defines a common vocabulary for agents who need to share a common context in an intelligent home environment.

Based on this layer and the dynamic composition of smart environments we have developed a spoken dialogue interface that adapts to heterogeneous smart environments. The interface and its behavior vary depending on the environment and its current state. Further information about the spoken dialogue interface can be found at [12].

The use of multimodal approaches can benefit the interface. A new face recognition module is going to be added to the system, in order to identify the people who are in the environment. This information can be used by several modules of the system, including the spoken dialogue interface, to improve their functionality. Following with this idea, the synchronization of speech and hand gestures can help to improve the interaction [2]. For this, a new gesture recognition module should be built. Other possible modal interaction can be produced by showing the information on a screen, instead of uttering a request. The user may answer either by speaking or by clicking on the selected choice.

Acknowledgments

This work has been sponsored by the Spanish Ministry of Science and Education, project number TIN2004-03140.

References

1. Alamán A, Cabello R, Gómez-Arriba F, Haya P, Martínez A, Martínez J, Montoro G (2003) Using context information to generate dynamic user interfaces. In Proceedings of HCI International 2003
2. Bourguet M, Ando A (1998) Synchronization of speech and hand gestures during multimodal human-computer interaction. In Proceedings of CHI’98 (Los Angeles, GA, April 18-23), 241-242
3. Carmona J, Atserias J, Cervell S, Márquez L, Martí MA, Padró L, Placer R, Rodríguez H, Taulé M, Turmo J (1998) An Environment for Morphosyntactic Processing of Unrestricted Spanish Text. Proceedings of LREC'98, Granada, Spain
4. Chen H, Finin T, Joshi A (2004) Semantic Web in the Context Broker Architecture. In Proceedings of PerCom 2004
5. Dahlbäck N, Jönsson A (1992) An empirically based computationally tractable dialogue model. Proceedings of COGSCI’92
6. Dey A (2001) Understanding and using context. Personal and Ubiquitous Computing. 5:1
7. Engelmore R, Mogan T (1988) Blackboard Systems. Addison-Wesley
8. Henricksen K, Indulska J, Rakotonirainy J (2002) Modeling Context Information in Pervasive Computing Systems. In Proceedings of Pervasive 2002, pp 167-180
9. Kidd CK, Orr R, Abowd GD, Atkenson CG, Essa, IA, MacIntyre B, Mynatt E, Starner TE, Newstetter W (1999) The Aware Home: A Living Laboratory for Ubiquitous Computing Research. In Proceedings of the Second International Workshop on Cooperative Buildings
10. Le Gal C, Martin J, Lux A, Crowley JL (2001) SmartOffice: Design of an Intelligent Environment. IEEE Intelligent Systems, 16:4, pp 60-66
11. Mozer M (1998) The neural network house: An environment that adapts to its inhabitants. In Proceedings of the AAAI Spring Symposium on Intelligent Environments
12. Montoro G, Haya PA, Alamán X (2004) Context adaptive interaction with an automatically created spoken interface for intelligent environments. In Proceedings of INTELLCOMM’04
13. Salber D, Abowd GD (1998) The design and use of a generic context server. In Proceedings of Perceptual User Interfaces (PUI’98)
14. Satyanarayanan M (2001) Pervasive Computing: Vision and Challenges. IEEE Personal Communications, 8:4, pp 10-17
15. Peters S, Shrobe H (2003) Using Semantic Network for Knowledge Representation in an Intelligent Environment. In Proceedings of PerCom'03