A Proposal for an XML Definition of a Dynamic Spoken Interface for Ambient Intelligence

Germán Montoro[1], Pablo A. Haya[1], Xavier Alamán[1],
Ramón López-Cózar[2], and Zoraida Callejas[2]

[1] Dept. of Computer Science and Engineering, Universidad Autónoma de Madrid, Spain
{German.Montoro, Pablo.Haya, Xavier.Alaman}@uam.es
[2] Dept. of Languages and Computer Systems, Universidad de Granada, Spain
{rlopezc, zoraida}@ugr.es

Abstract. Environments based on ambient intelligence require new interfaces that allow a natural interaction. The development of these interfaces has to be done in a standard way, considering the dynamic characteristics of these environments. In this paper we present the results in the development of an intelligent environment and a description language for the automatic generation of a spoken dialogue interface, which adapts to the characteristics of every environment.

1 Introduction

Ambient intelligence and intelligent environments [1] have appeared as a new field of research in the area of user interfaces. One of the aims of the ambient intelligence is to provide a natural interaction between an environment and its inhabitants.

These environments provide new possibilities of interaction [2], offering new challenges to the designers of the interfaces [3]. The environment must help people in their everyday life, offering non-intrusive ways of communication. Therefore classrooms, offices, laboratories and homes should be provided with their own entity and be capable of assisting their occupants in their tasks. Moreover, this interaction must be adapted to the task, the environment, its occupants, and the available devices [4, 5].

Furthermore, considering the heterogeneous and continuously changing characteristics of the intelligent environments, the spoken interface should be created automatically, adapting dynamically to the specific peculiarities of each environment.

In this paper we present the results in the process of definition and implementation of a spoken dialogue interface for intelligent environments that adapts to every domain. Dialogues are automatically constructed and it allows interaction with the environment, controlling the devices by means of a natural language spoken interface.

The generated dialogue system is based on a tree structure, where linguistic parts are represented by nodes. In addition, it contains a set of grammars that are also automatically created and define the possible ways of interaction with the environment. This tree is employed in the processes of comprehension and interaction with the users, that is, to interpret and generate sentences [6].

D.-S. Huang, K. Li, and G.W. Irwin (Eds.): ICIC 2006, LNAI 4114, pp. 711 – 716, 2006.

```
<class name="name" extends="type">
    <property name="propiety_name">
        [set_of_properties, ...]
    </property>
    [, <property name="propiety_name">
        [set_of_properties, ...]
       </property>
    ] ...
</class>
set_of_properties:
<paramSet name="set_name">
    <param name="parameter_name">value</param>
    [, <param name="parameter_name">value</param> ] ...
</paramSet>
```

Fig. 1. Syntax of a Document of Definition of Classes of Entity (DDCE)

The interaction interface is defined by an XML user interface definition language (XML-UIDL). This language has similar characteristics to other UIDLs such as UIML [7] or XIML [8], but provides new functionalities that adapt to the peculiarities of the intelligent environments.

To carry out our research we have built a real intelligent environment. This environment is a laboratory furnished as a living room, provided with several devices. Among them, we can find lights and switches, an electronic lock mechanism, speakers, microphones, a radio tuner, a TV set and RFID cards.

2 Definition of the Entities of an Intelligent Environment

In our system, the possible types of entities that can be found in an intelligent environment are defined in one or several Documents of Definition of Classes of Entities (DDCE). Each class of entity defines the universal characteristics that all the entities of that type must have. Instances of that class of entity inherit the common characteristics and may, if necessary, add new specific characteristics to the instance (see section 3).

Therefore, when a new type of physical device or application is designed, it comes with a DDCE, corresponding to the XML description of the general characteristics of that type of entity.

Each class is defined by its properties (specified by the label *property*) and, optionally, a set of common parameters (represented by the labels *paramSet* and *param*). Parameters can be associated with a specific property or a whole class. Besides, each class inherits the properties of its parent class (using the attribute *extends*). Figure 1 shows the syntax of a DDCE.

There are some universal types of classes, such as, *room, device* or *person*. Every class has to inherit from one of these types or from a class that has inherited from them. The definition of a universal type contains the basic characteristics that can be shared by all the classes of that type.

3 Interface Definition

The definition of the linguistic information associated with the entities is established in a set of documents called Documents of Parameterization of Classes of Entity (DPCE). This information is related to the classes of entity. This way, instances of these types automatically inherit all the defined properties. These instances are described in the Documents of Definition of the Entities of the Environment (DDEE).

The definition of the interface is done in two different steps, which can be carried out by different people at different times:

- Firstly, it is necessary to define the information related to each new class of entity in the DPCE. This is the linguistic information for the interaction, the methods employed for the automation of the system and, optionally, new grammar templates. This information is defined once and shared by all the entities of the same type.
- Secondly, the entities that can be found in the environment and their type are defined in the DDEE. The linguistic information of each entity of the environment is provided by the definition of the classes of entities at the DPCE. This way, in many cases, to create a spoken dialogue interface it is only necessary to define in the DDEE which elements are present in the environment. It is not required to modify or add any kind of additional linguistic information. Even so, the information inherited by each entity can also be parameterized, adapting it to the special characteristics of the environment.

3.1 Definition of the Linguistic Information Associated to the Classes of Entities

As mentioned above, each class of entity has its own linguistic information that establishes all the possible interactions that can be accomplished with the entities of that type. This information is classified in seven possible linguistic parts:

- Verb part (VP). It describes the actions that can be taken with the entity.
- Object part (OP). It establishes the possible names that the entities can take.
- Location part (LP). It describes the physical situation of the entity in the environment.
- Indirect object part (IOP). It specifies who receives the specified action.
- Modal part (MODALP). It describes how the action can be carried out.
- Quantification part (QP). It defines a value or amount that is applied to the action.
- Modifier part (MP). It adds new information to some of the previous parts.

The linguistic information associated with the classes of entities is defined in the DPCE. Later, this information will be shared by all the entities of that type defined in the DDEE. Therefore two entities of the same type inherit the same linguistic information and can be later parameterized depending on the specific characteristics of the entity in the environment.

In order to add the information related to the oral interface it is necessary to attach a set of parameters under a *dialogue* class label. Next, there will be an *action* parameter for every action that can be taken with the entity and, for each of them, one

```
<class name="name">
    <property name="property">
        <paramSet name="dialogue">
            <param name="action1"> action_name </param>
            <param name="skeleton1"> Part Word [, Part Word ]...
            </param>
            [, <param name="skeleton2"> Part Word
               [, Part Word ]...
               </param> ] ...
            [, <param name="skeletonm"> Part Word
               [, Part Word ]...
               </param> ] ...
        </paramSet>
    </property>
</class>
Part:
VP | OP | LP | IOP | MODALP | QP | MP
```

Fig. 2. Syntax of the linguistic information attached to the DPCE

or several *skeleton* parameters that define the possible skeletons of sentences that can be employed to invoke the action. The syntax is shown in figure 2.

The *action* parameters identify a new action and contain the description of the action that can be realized. The skeletons of sentences are identified by the *skeleton* parameter and hold the keywords employed to build the sentences that will invoke the associated action. Each keyword must be preceded by the linguistic part that it represents (*VP, OP, LP, IOP, MODALP, QP* or *MP*). It is possible to specify synonyms in a linguistic part writing two or more consecutive words. Sentence skeletons can begin with any linguistic part and be repeated as many times as it is necessary.

This document can be edited and modified to adapt to different oral ways of communication. For instance, some synonyms can be added or removed to adapt the interaction to the dialect of different regions. The interaction is easily definable and reconfigurable and can change, grow or adapt to new necessities in a straightforward way. Therefore, it is not necessary to modify the implementation of the dialogue interface, rather it is sufficient to edit and modify the definitions in the DPCE to make the changes available in the new interface.

3.2 Definition of the Instances of Entity

With the definition of the linguistic information associated with the classes of entities, and considering that all the entities of the same type share the same possibilities of interaction, most of the time it is only necessary to define which entities can be found in an environment to automatically obtain a customized speech interface.

This definition of the elements that can be found in an environment is written in the DDEE following the syntax showed in Figure 3.

Nevertheless, sometimes it will be necessary to specify linguistic information related to the specific environment in which the interface is created. This is the case, for instance, of an environment in which there are several entities of the same type, making necessary the addition of new information to differentiate between them.

```
<instances>
    instante_definition
    [, instante_definition ] ...
</instances>

instante_definition:
<entity name="name" type="entity_class"/>
[, <entity name="name" type="entity_class"/> ] ...
```

Fig. 3. Syntax of the Document of Definition of the Entities of the Environment (DDEE)

A similar situation arises with specific characteristics of the entities in a concrete environment, such as color, size, position, etc.

To solve this problem, it is possible to define a Document of Parameterization of the Entities of the Environment (DPEE), which allows the stipulation of new linguistic information related to a specific entity. To do this, it is required to append the parameter *add* to the properties which will receive the new information. Next, the attribute can be followed by a number that points out the number of the sentence skeleton where the new information must be attached. It may also be followed by the word *all*. Tshis specifies that the information should be added to every sentence skeleton in that property. Figure 4 shows the syntax of the linguistic parameterization of a property in an entity.

Once again, this is done to permit the configuration, description, modification and adaptation of the linguistic interactions in a homogeneous and straightforward way.

After the linguistic information associated with the classes of entities is defined in the DPCE, the designer of the environment interface only has to declare which entities are present in the environment, employing the DDEE. This designer or others will be able to modify these definitions, employing the DPEE and adapting the interface to a specific environment.

This information is gathered and merged in the Document of Description of the Environment (DDE), which is employed for the automatic generation of the spoken dialogue interface.

The definition of the linguistic information is carried out in a similar fashion as the definition of the entities, their properties and possible new interfaces. This way, the system provides a homogeneous and standard mechanism for the definition of the environment and its characteristics, including the interfaces.

```
<entity name="name">
    <property name="property">
        <paramSet name="dialogue">

            <param name="add_(1, all)">value</param>
            [ , <param name="add_2">value</param> ] ...

        </paramSet>
    </property>
</entity>
```

Fig. 4. Syntax of the linguistic parameterization in a DPEE

4 Automatic Generation of the Spoken Dialogue Interface

The linguistic information obtained from the DDE allows the automatic creation of a spoken dialogue interface which is adapted to the characteristics of the environment.

The creation process is comprised of two parallel steps. Firstly, the system creates a set of proper grammar for the interaction with the system. Secondly, it builds a linguistic tree employed for the interpretation and generation of sentences.

In order to automatically create the interface, the system reads the information stored in the DDE. For each one of the entities represented in the document (that is, for each of the entities of the environment), it obtains the linguistic information associated to the class of entity and, if necessary, it adds the new specific linguistic information for that entity.

At the same time, the system builds the linguistic tree. This process begins with an empty root node. Each of the parts of the sentence skeletons associated with the entities is transformed into information that is later attached to the tree. This can be done in the form of a new node or adjoining information to an existing one.

When the system finds a part with several synonyms it creates a node for each of them. The following children of that sentence skeleton will hang from each of them.

Acknowledgments

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

References

1. Remagnino, P., Foresti, G.L., Ellis, T. (Eds.): Ambient Intelligence: A Novel Paradigm. Springer Verlag (2005)
2. Weiser, M.: The World Is Not a Desktop. ACM Interactions. 1 (1994) 7-8
3. Shafer, S., Brumitt, B., Cadiz, J.J.: Interaction Issues in Context-Aware Intelligent Environments. Human-Computer Interaction. 16 (2001) 363-378
4. Paternò, F., Santoro, C.: One Model, Many Interfaces. In Proceedings of CADUI (2002)
5. Rayner, M., Lewin, I., Gorrell, G., Boye, J.: Plug and Play Speech Understanding. 2nd SIGdial Workshop on Discourse and Dialogue (2001)
6. Germán Montoro, Pablo A. Haya, Xavier Alamán.: Context Adaptive Interaction with an Automatically Created Spoken Interface for Intelligent Environments. INTELLCOMM 04. Bangkok, Thailand. November (2004)
7. Ali, M.A., Pérez-Quiñones, M.A., Abrams, M., Shell, E.: Building Multi-Platform User Interfaces with UIML. In Proceedings of CADUI. (2002)
8. Puerta, A., Eisenstein, J.: XIML: A Universal Language for User Interfaces. White paper. Available at http://www.ximl.org/Docs.asp. (2001)