Computational Linguistics and Intelligent Text Processing: 5th International Conference, CICLing 2004 Seoul, Korea, February 15-21, 2004. Proceedings. Lecture Notes in Computer Science, Volumen 2945. Springer, 2004. 360-370.

DOI: http://dx.doi.org/10.1007/978-3-540-24630-5_44

A plug and play spoken dialogue interface for smart environments

Germán Montoro, Xavier Alamán and Pablo A. Haya

Universidad Autónoma de Madrid
Departamento de Ingeniería Informática
Ctra. de Colmenar Km. 15. Madrid 28049 Spain
{German.Montoro, Xavier.Alaman, Pablo.Haya}@ii.uam.es

Abstract. In this paper we present a plug and play dialogue system for smart environments. The environment description and its state are stored on a domain ontology. This ontology is formed by entities that represent real world contextual information and abstract concepts. This information is complemented with linguistic parts that allow to automatically create a spoken interface for the environment. The spoken interface is based on multiple dialogues, related to every ontology entity with linguistic information. Firstly, the dialogue system creates appropriate grammars for the dialogues. Secondly, it creates the dialogue parts, employing a tree structure. Grammars support the recognition process and the dialogue tree supports the interpretation and generation processes. The system is being tested with a prototype formed by a living room. Users may interact with and modify the physical state of this living room environment by means of the spoken dialogue interface.

1 Introduction

In the last years, computational services have abandoned a centralized structure, based on the desktop metaphor, to be omnipresent in our environments. Following these changes, interfaces are transforming very fast to adapt to the new user necessities [1]. Leaving behind some old command line and graphical interfaces, systems have changed to provide new user-friendly approaches. The new interfaces have to adapt to the users, so that users may communicate with the system as naturally as possible.

Among these interfaces, spoken dialogue systems offer a wide range of possibilities, but also new challenges [2]. Dialogue interfaces (spoken or not) are frequently tied to a specific domain and, even more, are specially designed for the tasks they have to deal with. They are usually based on hand crafted dialogue designs. On the one hand, this makes harder to build a dialogue interface, increasing considerably its cost, on the other hand, changes in the system may necessarily imply modifications in the interface.

An approach to solve this problem is the automatic dialogue generation, based on ontological domain knowledge [3]. These systems provide plug and play spoken interfaces, avoiding the necessity of creating them from scratch and making them more easily reconfigurable.

This solution becomes much more effective in the case of smart environments. The configuration of these environments is highly dynamic and it may change considerably from one to another. New entities may be added, removed or temporally stopped, and the spoken interface should be aware of these changes.

In this paper we present a plug and play dialogue system for smart environments. Our system automatically creates and manages the dialogues, being based on the information stored on the environment ontology.

Section two presents a description of our system, section three describes the ontology; section four shows how the plug and play spoken dialogues are created; section five describes the implementation and evaluation of the system; and; finally, in section six we discuss the conclusions and future work.

2 System description

Our work is related to the research area known as smart environments. Smart environments are based on the concept of ubiquitous computing, originally defined by [4].

Ubiquitous computing systems provide access to computational services but make the computational devices invisible to the users. Users should not be aware of their presence and therefore the system should have a similar behavior to a human being [2]. Users are not in charge of finding the interface, but the system has the responsibility of serving the users [5].

Following the ideas of ubiquitous computing, a smart environment is a "highly embedded, interactive space that brings computation into the real, physical world". It allows computers "to participate in activities that have never previously involved computation" and people "to interact with computational systems the way they would with other people: via gesture, voice, movement, and context" [6].

Different and highly heterogeneous technologies may be found inside a smart environment, from hardware devices, such as sensors, switches, appliances, web cams, etc. to legacy software, such as voice recognizers, multimedia streaming servers, mail agents, etc. On the one hand, all of these entities have to be seamlessly integrated and controlled using the same user interface. For instance, users have to be able to start a broadcasting music server as easily as to turn off the lights. On the other hand, user interaction has to be kept as flexible as possible. It should be based on multiple and distinct modalities, such as web, voice, touch… so that user preferences and capabilities can be considered.

Bearing in mind these conditions we have developed a working prototype based on a real environment. This prototype includes an ontology, which provides a simple mechanism to represent the environment and communicate its state, and two different plug and play user interfaces (a web-based user interface and a spoken dialogue user interface), which interact with and control the elements of a real environment. These interfaces are automatically created and managed, being based on the information extracted from the ontology.

This real environment consists of a laboratory furnished as a living room, with several devices. There are two kinds of devices: control and multimedia. Control

devices are lighting controls, a door opening mechanism, a presence detector, smart-cards, etc. Multimedia devices, such as speakers, microphones, a TV and an IP video-camera are accessible through a backbone IP. Control devices are connected to an EIB (EIBA) network and a gateway joins the two networks. The blackboard that accesses to the physical layer is harmonized through a SMNP (Simple Management Network Protocol) layer [7].

3 Environment ontology

The ontology is implemented in a middleware layer, which is the glue between the user interface and the environment. The interaction between them is based on an event-driven protocol. The environment layout and its state are stored on a common repository, called blackboard [8].

This blackboard holds a representation of multiple characteristics of the environment. These include the distribution of the environment (buildings and rooms), the environment active entities, their location, their state, the possible relationships between them and the flows of information. The nature of an entity can range from a physical device to an abstract concept, such as the number of persons in a room or the list of persons allowed to get into it.

The ontological environment representation is written in an XML document. This document is parsed to obtain a repository, the blackboard, which is used as a proxy information server. Interfaces and applications may ask the blackboard to obtain information about the state of any entity or to change it. Entity descriptions can be added or 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 (see figure 1).

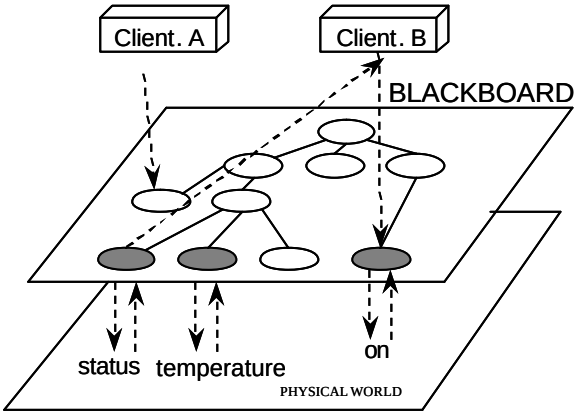

Fig. 1. Blackboard interaction

All the entities of the same type share a set of common properties that describe universal accepted features of the entity. Specific properties represent custom

application information. In order to automatically create dialogues associated to the entities, common linguistic information is attached to the entities. This linguistic information comprises most of the possible ways a user may employ to interact with the entity. Basically it 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 an additional information part (additional information about the entity). These linguistic parts permit the use of synonyms, to allow a more natural interaction with the system. Additionally, entities store the name of the action method that has to be called after its linguistic information is completed and the name of its associated template grammar. Figure 2 shows a simplified example of the linguistic information presented on an entity.

```
<entity name="Lamp_1" id="1" type="0">
 <property name="Status">
   <paramSet name="jeoffrey" id="2"> . . . </paramSet>
   <paramSet name="Odisea" id="3">
     <paramSet name="sentence" id ="4" action="EncenderApagarLuz">
           <param name="verbPart">encender dar</param>
           <param name="objectPart">luz</param>
           <param name="modifierPart"></param>
           <param name="locationPart">techo arriba</param>
           <param name="aditionalInformationPart"></param>
     </paramSet>
     <paramSet name="sentence" id="5"> … </paramSet>
</property>
</entity>
```

Fig. 2. Ontology dialogue information

The employ of these common linguistic properties makes that, in most cases, when a user defines a new entity its spoken dialogue interface will automatically come attached to it. Only in a few cases, some changes have to be done to adapt the location part for a particular entity or, eventually, to refine other parts.

4 Dialogue creation

As it was said above, the system employs the linguistic information presented on the ontology to automatically create the spoken dialogue interface.

At startup, the dialogue system reads every entity from the ontology. When an entity has associated dialogue information, the system adds the linguistic information to the appropriate grammar and adds a new dialogue part to the spoken dialogue interface.

4.1 Grammar creation

Most of the spoken dialogue systems employ specialized grammars to increase the recognition accuracy. Grammars support the recognition process by specifying the possible sentences that may be uttered by the users, limiting the number of possible inputs expected by the recognizer [9].

Our spoken dialogue system creates a grammar for every different type of entity and modifies it according to the linguistic information attached to the entity. As an example, an entity of type *light*, that represents a light presented in a room, has associated an action grammar. This action grammar allows to utter a wide range of sentences oriented to access to and control general environment devices.

In addition to the entity dialogues, the system may create new dialogues that do not depend on any entity. That is the case, for instance, of a *politeness dialogue*, which user and system may employ to thank, say hello, good bye, etc. This dialogue has associated a plain grammar. Plain grammars do not have any verb, questioning or location parts.

At the dialogue creation process the system reads all the entities from the ontology. For every entity the system checks if it has associated linguistic information. If so, it gets the name of its corresponding grammar template and checks if a previous grammar was already created by an entity of that type (as it was said above, all the entities of the same type share the same grammar). If that type of entity does not have a grammar yet, the system creates a new grammar based on the entity grammar template. For this grammar (or for a previous grammar created by other entity of the same type) it adds each linguistic part to the corresponding position in the grammar.

All the grammar templates have the same rules. They only differ on the possible sentences that are supported. The interface is based on Spanish, and therefore the nouns, adjectives and articles have number and gender. For instance, for the object part they all have the <singular female noun>, <singular male noun>, <plural female noun>, <plural male noun> and <invariant common noun> rules. Some of these rules will be filled in by different entities of the same type and some others will be left empty.

Given that the linguistic parts of an entity come in one form of the word (infinitive for verbs or singular male for nouns) we employ a tagged lexicon and a syntactic parser [10] to obtain the most appropriate rule and the rest of the forms of the words for that rule.

The new linguistic parts append new words to the rules, preserving the words previously added for other entities of the same type. If a rule already contained that word, the new word is not added again.

As an example of this process we may consider the entities fluorescent, lamp (both of type light) and front door (of type door) and a system dialogue for yes and no. The three entities have associated an action grammar and the system dialogue has associated a plain grammar. The fluorescent entity will create a new grammar based on the action grammar template and will add its parts to this grammar. Given that the lamp entity is of the same type as the fluorescent entity (type light) it will employ the grammar already created by the fluorescent entity to append its linguistic parts. The front door entity will create a new grammar (since there is not a previous entity of the type door) also based on the action grammar template and it will append its linguistic

parts. Finally the yes/no system dialogue will create a new grammar based on the plain grammar template. The grammar set will be formed by three grammars, two of them based on the action grammar template and the last one based on the plain grammar template.

Once the grammars are created, the system may employ them for recognition, and activate and deactivate them at its convenience.

4.2 Dialogue part creation

Besides the grammar creation process, for each entity the system creates the dialogue parts that will be employed later to provide interpretation and generation capabilities.

Dialogue parts are added as dialogue nodes to a tree structure. This structure establishes a new linguistic ontology that will carry on the process of interaction with the user.

At the beginning of the process the dialogue tree is formed by an empty root node. For each entity with linguistic information, the system adds new nodes or provides new information to existing nodes, according to the linguistic parts associated to the entity. Nodes are formed by a word and a list of the entities that contained that word.

To add a new part the system always starts from the root. The first linguistic part added to the tree is the verb part. If the root node does not have any child node containing the verb word, the system will create a new node containing the word and the entity where it belonged. If the root node already had a child node containing the verb word, the system adds the name of the entity to the list of entities of that node. The system follows the same process for all the verb synonyms. After the verb part addition is concluded it will continue with the object part. The object part nodes will be children of all the verb part nodes (all the verb synonyms) previously created. If the dialogue part did not have a verb part, the object part would hang from the root. This means that parts of the same type may have different depth in the tree. After the object part is added to the tree, this is completed with the modifier, location and additional information parts.

Let us see an example taken from our real smart environment (here simplified to improve its comprehension). The room has a fluorescent light and a radio. The fluorescent entity is called light_1 and the radio entity is called radio_1. The light_1 entity has associated the following linguistic information:

("encender dar", "luz", "", "techo arriba", "")
("apagar", "luz", "", "techo arriba", "")
("encender", "fluorescente", "", "", "")
("apagar", "fluorescente", "", "", "")

The first column corresponds with the verb part, the second column with the object part, the third one with the modifier part, the forth one with the location part and the last one with the additional information part. These sentence skeletons establish all the possible ways to interact with and control this fluorescent light. As it was said above, the use of grammars will transform these four skeleton sentences in multiple sentences that will provide a natural and intuitive way to interact with the system. Notice that some parts are empty and that other parts contain more than one word (what denotes the use of synonyms).

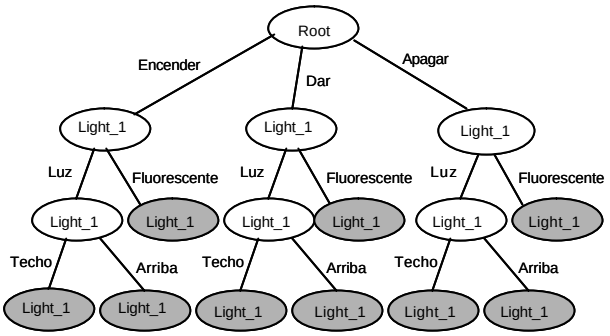

Fig. 3. Partial linguistic tree

Figure 3 shows the linguistic tree that is created with the linguistic information specified above. Shadowed nodes correspond with action nodes. Every time the dialogue system reaches one of these nodes it executes the action associated to the entity specified in the node. In this case, it would execute the action associated to the entity light_1, what would turn the fluorescent on or off, depending on the node.

Following with the creation of the linguistic tree, the radio entity has the next linguistic information (to simplify, we only show the turn off information, not considering the linguistic information to select a specific radio station or change the radio station or the volume, which is present on the real scenario):

("apagar", "radio", "", "", "")

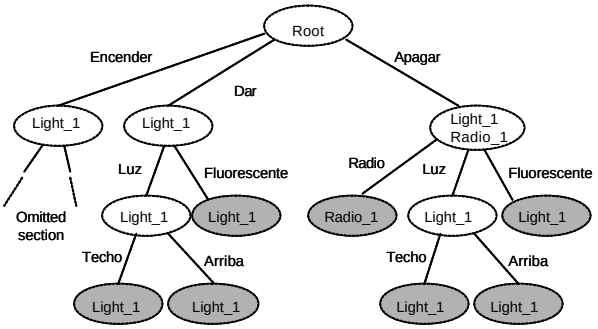

Fig. 4. New linguistic tree

Figure 4 shows how the tree is completed with this new linguistic information, adding new nodes and modifying some of the already existing ones. In this case, if the dialogue system reaches the shadowed node named radio_1, it would turn off the radio (or answer back to the user if the radio was already off), as it is set in the action method associated to the radio_1 entity. Notice that the node under the arrow "apagar" contains two entities. This means that there are two entities in the environment that may be turned off ("apagar"). This node information will be used later at the interpretation and generation processes .

4.3 Interpretation and generation

Once the linguistic tree is created, the system is ready to interact with the user by means of the spoken dialogue interface. This interaction will be based on the information presented on the linguistic tree. Although the generation and interpretation processes are quite complex and therefore they are out of the scope of this paper, we will give a brief overview of them for a better understanding of the tree generation process.

After a user utterance the system will go down the tree to interpret the sentence. If it reaches an action node, it executes the associated action method. Its execution usually implies some change in the environment, although it may also initiate a system question or answer utterance. If it does not reach an action node, the system will try to find out if it may take some action, being based on the environment current state and the nodes under the node where it stopped. In any case, the system will either take an action or prompt the user for more information by generating a sentence based on the information gathered from the environment and the tree.

To go down through the tree the system will take into consideration if a dialogue was completed in the last interaction (if an action was executed), if it is in the middle of a dialogue, the exact sentence uttered by the user and the environment state.

As an example, based on the tree described above, let us suppose that the user utters the sentence *I would like to turn off …*, where … corresponds with noise. The system starts on the root node, goes down the arrow "apagar" and stops there. Once there it may take several actions. The system checks the nodes under the node where it stopped and realizes that there are two entities that may be turned off: "radio" and "luz techo" (notice that the system recognizes that "luz arriba" and "fluorescent" refer to the same entity as "luz techo" so it only considers the first linguistic information encountered). Next, it gets the state of the fluorescent and the radio. If only one of them is on, it may turn it off. If they are both on, it will generate a sentence asking which one of them the user wants to turn off. If they are both off, the system will inform the user in this respect.

As it can be seen in this example, the dialogue interpretation and generation processes are related to the domain ontology and the linguistic information presented on the tree. Not only the dialogue creation process varies depending on the environment but the interpretation and generation do not follow a fixed script and they also adapt to the environment circumstances.

5 Evaluation

The system has been implemented and tested employing the real environment already mentioned. This environment consists of a laboratory furnished as a living room. It is equipped with a television, a DVD player, a radio tuner, an IP Web cam, a flat screen, an electrical door lock, an alphanumeric screen, personalized smart cards, two fluorescent lights, two halogen floor lights, two hi-fi speakers and one wireless microphone (see figure 5).

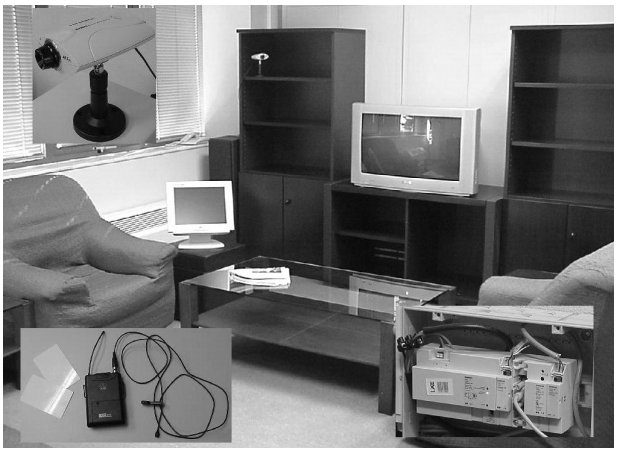

Fig. 5. Snapshot of the living room

Test dialogues interfaces have been created employing different room elements in each test. Combinations of elements of the same and different types were made so that we could check the correctness and accuracy of the grammar and dialogue tree creation processes. In any case the grammar and dialogue tree construction were fully satisfactory.

Let us explain one simple developed test to show the system performance. A spoken dialogue interface was automatically created for a room with a fluorescent light, two floor lights (each with a dimmable and a reading light), a radio tuner and a door lock. This interface has to support dialogues that allow to turn on and off the fluorescent or any of the two reading lights, dim up or down any of the two dimmable lights, open the door, turn on or off the radio and change the radio station. An additional *politeness dialogue* allows users and system to thank, say hello, good bye, etc. In this case the system created a new grammar (based on the action template grammar) for the fluorescent light, adding the information required to turn on, turn off and refer to a fluorescent light. After that it continued with the first floor lamp, composed by a reading and a dimmable light. It appended the information needed to refer to a reading light (the information employed to turn on or turn off a reading light was not added, since it was exactly the same as the fluorescent light information). After that it added the information necessary to dim up or down a dimmable light, to refer to it and, given that there is more than one floor light, it also added the location information employed to distinguish between both lights. Next, the system processed the second floor light. In this case it only had to append the location information for the new light, given that the rest of the linguistic information was shared with the first floor light. This concluded with the creation of the lighting grammar. Other three grammars were also constructed for the radio and door control and the *politeness* capabilities. Concurrently with the grammars creation, the system built the linguistic tree. The final tree was formed by dozens of nodes (many of them shared by several entities) and numerous action nodes. The automatic linguistic tree generation is not described in detail because it would imply a long explanation, but it followed the steps illustrated in section 4.2. These two processes completed the spoken dialogue

interface creation. After that, the grammar and dialogue trees were examined to make sure that they were properly created and that they supported all the possible known dialogues. Finally, we tested the system with real users to check the accuracy, efficiency and naturalness of the interface (this final evaluation test corresponds with the interpretation and generation processes).

In every case we found that the system created desired and accurate interfaces for the proposed environments and that the interface construction efficiently adapted to the different environment domains. Interfaces were always produced being based on real elements of our smart environment. As a result we got interfaces that could be employed by users to interact with real devices and rooms, making easier to evaluate their creation process and final performance.

6 Conclusions and future work

In this paper we have presented a dialogue system that automatically creates a spoken dialogue interface for a smart environment.

Smart environments are resulting in an increasing interest in the research and industry fields. They provide an *augmented* space that offers new services and ways of interaction [11]. These services have lots of applications for homes, working environments, elderly people, etc. improving considerably the quality of life of their inhabitants. Nevertheless, it is absolutely necessary to provide them with easy-to-use and intuitive interfaces.

To get this, the interface must be created with the minimum effort, making possible the fast development and implantation of these environments. According with these ideas we have developed a real smart environment. Users may interact with it thanks to a plug and play, automatically created, spoken interface. The environment is placed in a laboratory where it is being tested with trained and untrained people of different skills, in order to get performance results and improve its functioning.

Currently we keep working to improve and add new functionalities to the dialogue system.

The linguistic ontology will contain a new part, the question part, which will allow users to make questions about the environment. Current dialogue interpretation and generation processes will have to suffer some changes to support questions, but the dialogue creation process will remain as explained, except by the fact that the linguistic parts will be added to the tree in different order.

We are also trying to integrate the spoken modal interface with other modal interfaces. For instance, in some situations the system may require information to the user by uttering a sentence or by showing the information on a screen. The user may answer either by speaking or by clicking on the selected choice.

Dialogues are added to the system when entities with linguistic information are inserted in the domain ontology. Nevertheless they are not efficiently removed from the tree or stopped when the entity is eliminated from the ontology or stopped.

This plug and play spoken dialogue interface is the result of the continuous process of evaluation tests carried out during its development and current state. New tests,

with more and bigger rooms, will also refine the automatic creation methods so that we can test its performance in large scale environments.

Acknowledgments

This paper has been sponsored by the Spanish Ministry of Science and Technology, project number TIC2000-0464.

References

1. Weiser, M. “The world is not a desktop”. ACM Interactions, 1, 1, 7-8, 1994.
2. Yankelovich, N. “How do users know what to say?” ACM Interactions, 3, 6, December, 1996.
3. Milward, D. and Beveridge, M. “Ontology-based dialogue systems”. IJCAI WS on Knowledge and reasoning in practical dialogue systems, Acapulco, Mexico, August 10, 2003.
4. Weiser, M. “The computer of the 21st century”. Scientific American, 265, 3, 66-75, 1991.
5. Abowd, G.D. “Software design issues for ubiquitous computing”. IEEE CS Annual Workshop on VLSI: System Level Design (IWV '98), Orlando, FL, April 16-17, 1998.
6. Coen, M.H. “Design Principles for Intelligent Environments”. Proceedings of the AAAI Spring Symposium on Intelligent Environments (AAAI98). Stanford University in Palo Alto, California, 1998.
7. Martínez, A.E., Cabello, R., Gómez, F. J. and Martínez, J. “INTERACT-DM. A Solution For The Integration Of Domestic Devices On Network Management Platforms”. IFIP/IEEE International Symposium on Integrated Network Management. Colorado Springs, Colorado, USA, 2003.
8. Engelmore, R. And Mogan, T. Blackboard Systems. Addison-Wesley, 1988.
9. Dahlbäck, N. and Jönsson, A. “An empirically based computationally tractable dialogue model”. Proceedings of the 14th Annual Conference of the Cognitive Science Society (COGSCI’92), July 1992.
10. Carmona, J.; Cervell, S.; Atserias, J.; Cervell S.; Márquez, L.; Martí, M.A.; Padró, L.; Placer, R.; Rodríguez, H.; Taulé, M. and Turmo, J. “An Environment for Morphosyntactic Processing of Unrestricted Spanish Text”. Proceedings of 1st International Conferenceon Language Resources and Evaluation (LREC'98), Granada, Spain, 1998.
11. Wellner, P. “Interacting with paper on the digital desk”. Communications of the ACM, 36, 7, 86-96, 1993.