# SPOKEN INTERACTION IN INTELLIGENT ENVIRONMENTS: A WORKING SYSTEM

Germán Montoro, Xavier Alamán and Pablo A. Haya[1]

*Abstract*
*Intelligent environments are ubiquitous computing systems that react to and interact with their inhabitants. They adapt to their necessities, assisting them in their every day life. We present a real intelligent environment that supports spoken interaction with its users. The spoken dialogue interface is automatically created according to the environment and the interpretation and generation vary depending on the physical environment context.*

## 1. Introduction

Since the appearance of the term ubiquitous computing in the early nineties [15] there is a higher research interest in the development of the, so called, pervasive environments or intelligent environments [2, 3]. These environments are context-aware spaces "in which a ubiquitous computing system has contextual awareness of its users and the ability to maintain consistent, coherent interaction across a number of heterogeneous smart devices" [14]. Context is relevant information of the interaction between users and applications, which can be provided to the applications [5].

This contextual information can be used in many different ways in order to assist to the diverse modules that are part of an intelligent environment. Thus, environments can employ the context to learn the user actions, preferences and requirements, so that they can predict and adapt to their necessities. This way, they react to the user intentions with an automatic behavior [4, 11]. Environments can also employ the contextual information to assist the automatic creation of a user interface adapted to them [9, 10], to support the interface in the interaction with the users [14] or to improve the performance of the sensors or recognizers [13].

We describe how we use the contextual information to create a plug and play spoken dialogue interface for an intelligent environment. Furthermore, once the interface is created, the system employs the context to support the interaction with the users and to improve the interface capabilities. Next section briefly introduces our intelligent environment and how it is represented. After that we explain the creation of the interface and its interaction with the users and, finally, we present the conclusions and future work.

[1] Escuela Politécnica Superior, Universidad Autónoma de Madrid, Spain, German.Montoro@uam.es

## 2. Interact: an intelligent environment

Interact is an intelligent environment that tries to assist its inhabitants in their activities, "augmenting" the space to provide new functionalities and ways of interaction. Several sensors and devices have been placed into the environment. These include five control devices for five different lights (one fluorescent, two floor lights and two dimmable lights), a door opening mechanism, several smart identity cards, a radio tuner, a TV set, microphones, speakers, an IP video camera and two flat screens. These sensors and devices are connected to an EIB (EIBA) network and to a backbone IP. The application that accesses to the physical layer is harmonized through a SMNP layer [8]. Besides, the environment runs diverse software modules, such as a voice recognizer and synthesizer, a multimedia streaming server, an email notification and a presentation and picture agent.

With these elements we have built a real living room environment that serves as a testbed for the study and development of intelligent environments. The environment is also provided with two different user interfaces: a graphical web-based user interface [1] and a spoken dialogue interface. Both interfaces are automatically created, being based on the environment contextual information, and the interaction with them produces physical changes in the environment (turns on a light, opens the door, selects a new radio station, etc.). The environment is also provided with some automatic behaviors, trying to adapt to its inhabitants necessities (it shows a person favorite paintings in a flat screen when this person gets into the room or it turns off all the lights when the environment is empty).

The environment is initially described in a XML document. It contains information of multiple characteristics of the environment, such as its distribution (in buildings and rooms), its active entities (including devices or meta-information related to them), their location, their state and relationships between them. With this document the system automatically builds a blackboard [6] (which works as a layer between the physical world and the applications) and the user interfaces adapted to that specific environment.

Once the system has created the blackboard environment representation, the interfaces applications and users can start interacting with the environment. Interaction is adapted to the environment active entities and their state. Therefore, entities can be added and removed from the environment in runtime or they can change their state, and applications and interfaces automatically adapt to this new situation.

## 3. Odisea: a spoken dialogue interface for intelligent environments

Odisea is a spoken dialogue system that allows interaction between users and intelligent environments. Odisea is composed of a set of grammars and a dialogue structure, based on a tree. These elements are automatically created and permit to carry on context-based spoken interaction between the environment and the users.

### 3.1. Automatic interface creation

The grammars and the dialogue tree structure are created on startup according to the environment description represented on the XML document. Entities stored on this

document can have associated some linguistic information. This information is formed by a set of linguistic parts and specifies the possible spoken interactions that can be established between the entity and the user. Entities of the same type share most of the linguistic information and only have to indicate some specific features (such as the location, size or color). For instance, two different lights in the same environment will share most of the linguistic parts regarding to the way of interacting with and controlling them, and it will only be necessary to specify their location or some other specific features. Besides this linguistic information, entities should store the name of their related template grammar (an action grammar, a yes/no grammar, etc.) and the name of the action method associated to that entity. Action methods execute physical actions related to the entity (turn on or off a light, send an e-mail, regard a new user, etc.).

With this linguistic information the system builds the grammars and the dialogue tree that support the spoken recognition and interaction. There will be one grammar for each type of entity (every entity of the same type shares the same grammar) and they will be based on their associated grammar template. The dialogue tree structure is initially empty and the sets of linguistic parts are transformed in nodes that form tree paths. Every node stores the word corresponding to that part and the entity where it belonged. Parts with more than one word (synonyms) are transformed in different nodes. For instance, largely simplifying the set of parts for each entity, two "light" entities, a "radio tuner" entity and a "send e-mail" entity would produce the tree structure showed in figure 1. Words in arches correspond to the word parts that must be addressed in order to perform an action with the entity. Words inside the circles correspond to the entities that are associated to that tree path. Shadowed circles correspond to action nodes (after reaching them, the system can execute the action method associated to its entity).

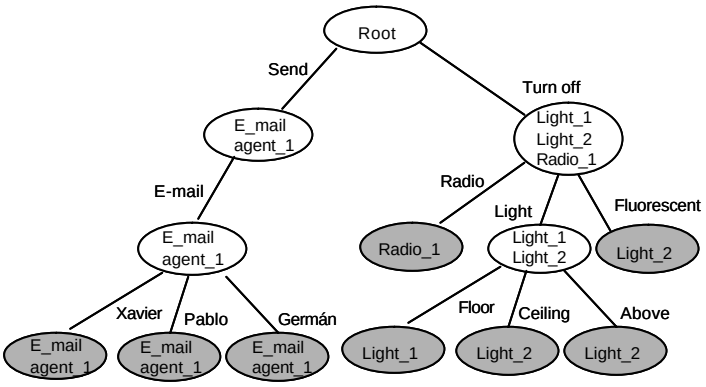


**Figure 1 Example of a simple linguistic tree**

### 3.2. Interacting with the environment

Odisea is managed by a dialogue supervisor, which is in charge of carrying on the dialogue interaction. This supervisor receives the semantic tags corresponding to the user utterances. Supported by the tree structure and the physical room context, it tries to correctly interpret the user intention and generate a response or execute an action.

Initially the supervisor is in a sleeping mode. After a user utters “odisea” it wakes up and answers “how may I help you?”. In that moment the user can start the interaction with the environment. After seven seconds, or seven seconds after the last dialogue interaction, the supervisor returns to the sleeping mode.

Once awake, when the supervisor receives a user utterance, it checks for matches with the linguistic tree, starting from the root. The supervisor only considers the matches with those nodes that contain some entity with a different state to the user requested state (interpreting that the user did not want to follow the other paths) and with nodes that do not correspond with synonyms of previous ones (avoiding repetitions). Following this method, every time it gets a match it descends to that node in the tree. This process is repeated until there are not more matches or until the supervisor reaches an action node.

If the supervisor reaches an action node it only has to call the action method associated to its entity. In many cases it will produce a physical change in the environment but it also can execute a program (for instance, an e-mail client) or generate a spoken message (to provide information).

If the supervisor does not stop in an action node it needs clarification. To do this it makes a tree search starting from the node where it stopped (considering only the nodes with some entity with a different state and avoiding the synonyms). This makes that the interpretation and clarification processes can adapt to the current physical context. As an example, let us consider two different scenarios for the dialogue represented in the figure 1. In the first one the *light_1* and *light_2* are both on. In the second one the *light_1* is on and the *light_2* is off. Then the user utters “could you turn off the light?”, so that the supervisor stops in the node “light”. In the first scenario the supervisor goes down to the nodes “floor” and “ceiling”, since their entities have a different state (on) to the user request (off), and avoids the node “above”, because it is considered a synonym of “ceiling”. With this, it produces the answer “Do you want to turn off the floor or the ceiling light?”. In the second scenario the supervisor goes down to the “floor” node but avoids both the “ceiling” node, because it does not have any entity with a different state, and the “above” node, because it is a synonym of the previous one. Given that it has only processed one action node (there is only one light on) it will execute its action method and will turn off the *light_1*. With this example we see that the supervisor has been able to interpret the user utterance successfully, although the user did not specify the complete name of the entity. Furthermore we see that the same interaction can produce different reactions depending on the current environment state. The number of entities can increase or decrease, or there can be many entities of the same type. The supervisor automatically adapts to these situations varying its interpretation and clarification behaviors.

The supervisor supports either to continue with a previous dialogue or to shift the dialogue task in the middle of a conversation. For this, after a clarification answer it starts checking for matches from the node where it stopped in the last interaction (following a previous dialogue) and from the root node (initiating a new one) and gets the best result.

Odisea is also provided with recovering capabilities for some recognition and interpretation errors. Besides start checking for matches from the root and previous

dialogue nodes, it goes over these nodes and also checks for matches for their children. Therefore the system recovers from misinterpretations (if the supervisor followed an erroneous path it can start from the root in the next iteration) and misrecognitions (if there is a recognition error skipping one node may lead the supervisor to the right path).

Odisea also supports pronominal anaphora resolution. For this, the supervisor creates some special anaphora resolution nodes and stores the last followed path. When a user employs a pronoun to refer to an object the supervisor stops in one of these nodes. Then, it goes to its corresponding regular node and follows the same path as the last uttered sentence. For instance, if a user utters "I would like you to turn on the radio" and after "could you turn it down?" the supervisor will stop in the anaphora resolution node "turn it". From there it goes to its corresponding node "turn" and after that it follows the last path (it goes down to the "radio" node). Finally, it continues with the initial sentence, this is, it goes to the "down" node. As a consequence it turns the radio down.

The generation is also adapted to the current physical context. When the system needs clarification and starts searching the child nodes, it also builds a possible answer sentence. This sentence will only refer to those nodes with a different state in some of their entities and will take into account if the node corresponds with a verb, an object, a location, if it has to offer multiple alternatives, etc. in order to build the sentence properly. Once the supervisor has finished the clarification and has built the answer sentence there can be three possible situations: (1) if there is only one possible action to offer it executes it, (2) if there are from one to three possible actions it utters the constructed sentence, offering the user all the possibilities and (3) if there are more than three possibilities it utters a more general sentence, so that the user do not have to retain too much information.

Finally, in some occasions, instead of a sentence Odisea utters light-weight audio signs [12]. These signs are used to avoid disturbing users if it is not necessary. The supervisor reproduces an audio sign when the system returns to the sleeping mode and another audio sign when it does not get any match after a user utterance. These situations can happen after a user finished its interaction with the system or after a recognition error. In both of them it is not necessary to distract her, although the system still offers some information

## 4. Conclusions

Interact is an intelligent environment that is able to interact with and react to its inhabitants. It is provided with a spoken interface, Odisea, where the dialogue creation and interaction are based on the context. It automatically adapts to the environment characteristics and entities. One of the main ideas of the project is that is built on a real intelligent environment. Users can interact with its elements and produce and see physical changes in the environment.

The system was evaluated in a general public fair. Novice users could interact with the environment and see the result through a webcam connected with it. New functionalities have been added since then and we are developing new user tests. These tests will allow us to determinate the accuracy of the recognizer, the task success and the elapsed time to complete a task, among other results that help to determinate the system performance [7].

We still have to modify or add some system capabilities. The system allows interacting with the entities but not asking for their state. We hope to support this functionality by creating a similar tree to the one explained here. We also have to study how to integrate diverse modalities. A new face recognition module is going to be added to the environment and it could be useful to employ a gesture recognition module in conjunction with the voice recognizer.

## 5. Acknowledgments

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

## 6. References

[1] ALAMÁN, X., CABELLO, R., GÓMEZ-ARRIBA, F., HAYA, P., MARTÍNEZ, A., MARTÍNEZ, J. and MONTORO, G. "Using context information to generate dynamic user interfaces". 10th International Conference on Human-Computer Interaction, HCI International 2003. Crete, Greece. June 22-27, 2003.

[2] BRUMITT, B., MEYERS, B., KRUMM, J., KERN, A. and SHAFER S. "EasyLiving: Technologies for Intelligent Environments" Proceedings of Handheld and Ubiquitous Computing, 2nd Intl. Symposium, September 2000, pp. 12-27.

[3] 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.

[4] COOPERSTOCK, J.R., TANIKOSHI, K., BEIRNE, G., NARINE, T. and BUXTON, W. "Evolution of a reactive environment". Proceedings of CHI'95 (Denver, CO, May 7-11).

[5] DEY, A.K., SALBER, D. and ABOWD, G.D. "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications". Human-Computer Interaction (HCI) Journal, 16 (2-4), 2001, 97-166.

[6] ENGELMORE, R. and MOGAN, T. "Blackboard Systems". Addison-Wesley, 1988.

[7] LITMAN, D.J. and PAN, S. "Empirically Evaluating an Adaptable Spoken Dialogue System". Proceedings of the 7th International Conference on User Modelling, 1999.

[8] 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.

[9] MILWARD, D. and BEVERIDGE, M. "Ontology-based dialogue systems". IJCAI WS on Knowledge and reasoning in practical dialogue systems, Acapulco, Mexico, August 10, 2003.

[10] MONTORO, G., ALAMÁN, X. and HAYA, P.A. "A plug and play spoken dialogue interface for smart environments". Proceedings of Fifth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing'04). Seoul, Korea. February 15-21, 2004.

[11] MOZER, M.C. "The neural network house: An environment that adapts to its inhabitants". Proceedings of the AAAI Spring Symposium on Intelligent Environments (AAAI98).

[12] MYNATT, E.D., BACK, M., WANT, R. and FREDERICK, R. "Audio Aura: Light-weight audio augmented reality". Proceedings of ACM UIST'97 (Banff, Canada), 211-212, 1997.

[13] NAGAO, K. and REKIMOTO J. "Ubiquitous talker: Spoken language interaction with real world objects". Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95), Vol. 2, 1284-1290, 1995.

[14] SHAFER, S.A.N., BRUMITT, B. and CADIZ, JJ. "Interaction Issues in Context-Aware Intelligent Environments". Human-Computer Interaction (HCI) Journal, 16, 2001.

[15] WEISER, M. "The computer of the 21st century". Scientific American, 265, 3, 66-75, 1991.