A Digital Secretary for Smart Offices Setup Up

João Laranjeira, Carlos Freitas, Goreti Marreiros,
Carlos Ramos, and João Carneiro

Abstract. In Ambient Intelligence paradigm, helping people in their daily routine tasks is a priority that has to be accomplished. People have less availability and the number of devices is growing, so some kind of autonomy must be given to the environment. In this paper we expose a system able to prepare meeting rooms for the execution of several kinds of events. Our proposal is based on an agents' community that performs in an autonomous way the operations required to setup the meeting room to a specific, including the configuration of software and hardware. The agents are autonomous and fulfilled with capabilities to prevent software and hardware failing. The proposed digital secretary was tested in LAID (Laboratory of Ambient Intelligence for Decision making) present in GECAD and it was able to promote the environment autonomy.

1 Introduction

Ambient Intelligence (AmI) deals with a new world where computing devices are spread everywhere (ubiquity), allowing human being to interact with physical world environments in an intelligent and unobtrusive way [9]. AmI can be introduced in several different environments and involves many different disciplines, like automation (sensors, control and actuators), human-machine interaction and computer graphics, communication, ubiquitous computing, embedded systems and, obviously, Artificial Intelligence. In the aim of Artificial Intelligence, research expects to include more intelligence in the AmI environments, allowing a better support to the human being and the access to the essential knowledge in order to make better decisions when interacting with these environments [9].

Carlos Freitas · Goreti Marreiros · Carlos Ramos
Institute of Engineering – Polytechnic of Porto

João Laranjeira · Carlos Freitas · Goreti Marreiros · Carlos Ramos · João Carneiro
GECAD – Knowledge Engineering and Decision Support Group
e-mail: jpcl@isep.ipp.pt, cff@isep.ipp.pt,
goreti@dei.isep.ipp.pt, csr@dei.isep.ipp.pt

P. Novais et al. (Eds.): Ambient Intelligence - Software and Applications, AISC 92, pp. 69–76.
springerlink.com

Group decision making is, by definition, an excellent area that demonstrates the potential of Smart Meeting Rooms (SMR) [6]. Considering group decision making, in which meetings will be distributed by various places and/or in rooms filled with devices and sensors, meetings places arrangement tend to be monotonous and time consuming. Smart Offices (SO) concepts, a subdiscipline of AmI, emerged with the promise to optimize/improve interactions with such tools and also to minimize the users' effort [6]. We will now analyse some of the definitions presented in literature. Le Gal C [5] defined a Smart Office as an environment that is able to help its inhabitants to perform everyday tasks by automating some of them and making the communication between user and machine simpler and effective. Marsa-Maestre et al [7] defined Smart Office concept as an environment that is able to adapt itself to the user needs, to release the users from routine tasks they should perform, to change the environment in order to suit their preferences and to access services available at each moment by customised interfaces. Ramos et al [10] defined Smart Office concept as an environment that is able to reduce the decision-cycle offering, for instance, connectivity wherever the user is, aggregating the knowledge and information sources.

There are several projects exploring SO concept and Active Badge [11] was the first approach. Monica Project [5] intends to anticipate user intentions and augment the environment to communicate useful information through user monitoring. In Intelligent Environment Laboratory of IGD Rostock [3] is intended to create an interactive environment based on multimodal interfaces and goal-oriented interaction differing from other systems that normally use a function-oriented interaction. Sensor-R-Us is placed in the University of Stuttgart [8] and this application is useful in order to know the position and status of persons and other information about the room, such as the temperature, its availability or the number of meetings that a person has in a specific day. EasyMeeting [1] explores the use of FIPA agent technologies, Semantic Web ontologies, logic reasoning, and security and privacy policies. Its goal is to create a smart meeting room that can facilitate typical user activities in an everyday meeting. Considering that routine tasks must be performed in an autonomous way we present an agent based on digital secretary that aims to prepare a SMR for the reception of events, for instance presentations, decision-making meetings using GDSS's. Such events require tasks like connecting servers and workstations, running applications that are used in process, connecting/adjusting audio and cameras. We aim to set up such routines in an autonomous way. With this digital secretary the SMR will automatically adapt itself to an event context. The user can perform such action by scheduling events or triggering them when he is inside of the room. A scalable approach, give to users the power to build new routines and increase the room autonomy were our main goals. In this paper we will demonstrate the system architecture and the users' interfaces. Then, in experiments section, the digital secretary that we propose in this paper is applied and tested at LAID, a SMR presented in GECAD. Conclusions and future work are given in the last section.

2 MeetingRoomSetup System

MeetingRoomSetup System aims to give meeting rooms the ability to prepare itself for the reception of events. Implemented events include presentations, meeting's assisted by GDSS, idea generation meeting and choose and selection of alternatives meeting. All mechanisms will be activated to perform the tasks required to trigger the event. These processes are performed by the agents' community that approach the problem in a cooperative way, in order to adapt the environment to the desired context in an autonomous way.

In order to achieve these goals we propose a model based in 3 concepts: *Tasks, Routine and Services*. *Tasks* represent all the possible events that can occur, such as presentations, idea generation meetings, choose and selection of alternatives meeting, etc. *Routines* can be seen as repetitive operations that must be performed, launching a software, hardware or alerts, run the same applications in several computers, connect some computers, send alerts to the administrator. Finally, *services* are atomic operations which have as a function the activation of software, hardware or sending an alert, such as, connecting this computer, running a specific application in a specific computer or sending email to administrator.

In terms of architecture MeetingRoomSetup is composed by two main components: MeetingRoomSetupWeb and a community of agents. There are three secondary components: a database, an application of management for agents and an application of BackOffice to the management of the system data.

In the following subsection we explore the proposed architecture, its components and the interactions among different components.

2.1 Architecture

Fig. 1 represents the architecture of the system. The architecture is composed by an agents' community implemented using Open Agent Architecture (OOA) framework, a relational database and several interfaces of access to different users.

In the agents community we have introduced 3 kinds of agents, including task agents that represent events that can occur in meeting rooms; routine agents who represent a group of tasks that need to be performed in order to achieve a task, which can connect computers, servers, run applications, etc; and service agents that represent small services, for example connect a computer, send email to system administrator, turn the sound on, etc. Such agents' structure enables the users to define new events that the environment can receive; the administrator can perform such task by reusing routines or even by personalizing new ones by selecting small operations (services) already existent. They are also able to develop new agents and introduce them in the environment and in the community. The agents' community is coordinated by the agent facilitator (1), responsible to deliver the messages to the right agent. Computers present in the environment are activated by agents inside the community, once started they have agents who run locally (agents service). The purpose of these agents is to execute operations that must be performed locally.

Agents' community interaction with the user is performed by the following modules: MeetingRoomSetupWeb, AgentsManagerApplication and OAAMonitor. The first one is in a webpage format. The second one is an application that aims to register agents in community of agents and to help the system administrator to test tasks activation. OAA Monitor application is an interface that allows monitoring community of agents.

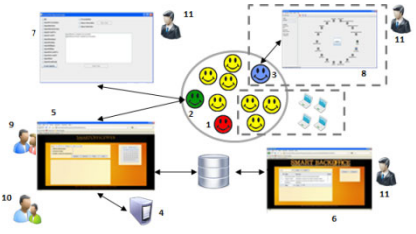

Fig. 1 Digital Secretary Architecture

All data circulating in the system is represented by MeetingRoomSetupDB component. It's a database which stores all data of the system. This data will be operated by some of the agents, web interface and SmartBackOffice. Users are directly involved with the system. They are divided in 3 groups: intern users, external users and administrator user. The first and second type have access to interface web with the aim to consult some data of the system, but the main goal is to benefit from the main functionality of system (tasks scheduling and tasks execution). The role of the administrator is to manage the community of agents and manipulate the data in the system. For that he will use the AgentsManagerApplication (community of agents' management) and SmartBackOffice (data manipulating).

2.2 Community of Agents

As we have seen in the previous section, the community of agents includes different types of agents: agent app (2), agent monitor (3), agent facilitator (1), agent task, agent routine and agent service. Agents' community was developed with the OAA framework and agents were implemented in JAVA. Agent App is an interface agent. It makes a bridge of communication between the agents' community and the interface component and vice versa. Interface components are AgentsManagerApplication and MeetingRoomSetupWeb. Agent Monitor is responsible for the control and management of all agents presented in the community of agents. This application will help the administrator of the system to control and manage the community of agents. Agent Facilitator is coordinating the community of agents. Each agent of the community of agents registers their abilities or features in the Agent Facilitator. When services are requested by an agent, instead of asking to a specific agent that performs this task, the agent simply make the request to the Agent Facilitator and this one decides which agents are available to that task. Agent task represents a task that the system can solve. At this moment

the digital secretary is able to perform 3 different tasks: presentation, idea generation and idea selection. For each task there is an agent task associated. They communicate with others agents with the purpose of these agents activate the system. Agents task knows the start date, end date and all resources needed to start the task. Agent Routine represents a routine in the system. The examples of routines are: connect computers, send alerts, run applications, among others. Each agent routine is responsible for one routine. For example, "agent alert" knows that we must alert 5 users, so he send 5 requests to "agent email" that is responsible by the service of sending emails. Agent Service represents a service in the system. Examples of services are: connect computer 12, run Firefox browser, send email to user admin, among others. Agent service receives a request of agent routine and executes this request with the purpose of preparing the room for an event. These agents are terminal agents because they don't send request to others agents - they execute services. There is an agent service that is special – the AgentSearchTasks. This agent constantly searches for scheduled tasks. When this agent finds a task, launches it in system then. All agents of this community are prepared to prevent software and hardware's failure that affect the proper operation of an event. For example, if a computer fails then the agent will connect other computer to prevent a wrong task operation. The main motivation to use agents in this system was to take advantage of the characteristics of persistence, autonomy and proactivity associated with the agents. Other components allow maintaining a scalable system that features can be extended by users. Together they endow the environment with the capability to resolve problems that may arise.

2.3 MeetingRoomSetupWeb

The aim of MeetingRoomSetupWeb (5) is to help the user to get all functionalities of the system. Users (9) inside the environment are able to access all functionalities of the system, whereas the external user (10) may only access scheduling functionalities, view tasks and cancel tasks. Through this application, administrators are able to control all tasks carried. External users must pass through an LDAP server (4) authentication. Beyond functionalities of external user, internal user may start or stop a specific task. Internal users are considered authenticated just by being in the environment and have access to MeetingRoomSetupWeb.

In Fig. 2 we can see the connection of the computer's process. In this example the problem that we want resolve is to prepare our test case environment, LAID, to a presentation. Before this process, a task agent (AgentPresentation) receives a request from the system to start a presentation. In the first step of this process, AgentConnectPCs (agent routine) requests to local agent (AgentConnectPC) to connect and this agent "wakes-up" the computer (PC22). After this, AgentConnectPCs (agent routine) verifies if the local agent (AgentConnectPC) is active. If the agent is unavailable then the AgentConnectPCs sends the request to another AgentConnectPC (agent service) that is available. This process is repeated until it finds a replacement computer or until it finishes the available resources.

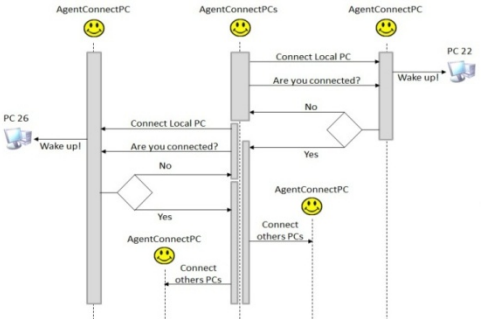

Fig. 2 Connection of computer process

2.4 SmartBackOffice and AgentsManagerApplication

SmartBackOffice application (6) allows the administrator to manipulate the data base of the system. It allows setting up several events, creating new ones by using the routines and services existent. It also allows adding of new services that included the addition of new agents with new kinds of services. This application allows scheduling an event to a certain date, so the system will be automatically activated on the chosen date. AgentsManagerApplication (7) is a very simple application that allows the administrator to run all agents required to execute all the tasks available. So this application will have a very simple interface where there are multiple options, for example, to create or delete agents. OAA Monitor (8) is an application that helps AgentsManagerApplication. OAA Monitor has the role of showing in a graphical way, all registered agents in the community of agents. The administrator can create new agents to add new tasks, routines and services to the system. For example, if a new device was added to the environment, the administrator could create a new agent to control the device.

3 Experiments

To evaluate this system, we are going to present a scenario and a problem that will be resolved. The goal is to prepare the LAID to an idea generation meeting. During this experiment, agent task will delegate procedures to agents routines that will execute them. The agent connectPCs (agent routine) will control the connection of computers. If any computer doesn't connect then this agent will contact and connect another one. To complete this task it is necessary to fulfil the following routines and services:

- Idea generation Meeting (Task)
  - Connect LAID's computers for 4 decision points (Routine);
    - Connect PC21, PC22, PC23 and PC25 (4 Services);
  - Run needed applications (Routine);
    - Run IGTAI [2] on all computers (4 Services);
  - Connect sound and video (Routine);
    - Connect all cameras and audio devices (2 Services);

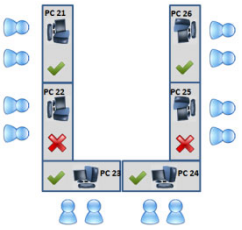

Fig. 3 Process of connecting LAID's computers

In Fig. 3 we can see the computers and users distribution in LAID. In this figure is represented the process of computers connection. PC22 and PC 25 are unavailable. AgentConnectPCs will detect this problem and connect two available computers, for example, PC 24 and PC26. Therefore the system adapts to the raised problems. To complete the task, AgentRunApplications will run applications on these computers. To evaluate this system, we compare it with iTalc [4] (Intelligently Teaching And Learning with Compute) and with a manual preparation of LAID. In Table 1 we can see the time needed to prepare the LAID for the scenario presented above. The results obtained with the MeetingRoomSetup System were considerably better compared to the other two options analysed.

Table 1 Time needed to prepare the LAID

	Time needed to prepare the LAID (min)
MeetingRoomSetup System	4
iTalc [4]	13
Manual Preparation	27

4 Conclusions and Future Work

In this paper was proposed a digital secretary endowed able to prepare a meeting room to receive different kinds of events. Realising the users from this hardware and software setup will contribute to the achievement of AmI characteristics in meeting rooms. The use of cooperative agents in the proposed architecture allows endowing the proposal with persistence, autonomy and proactivity. Giving hierarchy to agents and the ability to deploy them in different locations considering their goals allows us to build a scalable system that features can be extended by users.

As future work we want to develop the usage of semantic technologies to represent the state of the environment in real time, by forcing the agents to update a semantic representation of the environment state when they change it. We believe it will be possible to optimize agent's operations. Also, we intend to represent the users by means of agents in order to recognize automatically user's intentions, preferences and profile when they are in the meeting room.

Acknowledgments. The authors would like to acknowledge FCT, FEDER, POCTI, POSI, POCI, POSC, and COMPETE for their support to R&D Projects and GECAD.

References

[1] Chen, H., Perich, F., Chakraborty, D., Finin, T., Joshi, A.: Intelligent agents meet semantic web in a smart meeting room. In: 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems, New York, USA, pp. 854–861 (July 2005)

[2] Freitas, C., Marreiros, G., Ramos, C.: IGTAI- An Idea Generation Tool for Ambient Intelligence. In: 3rdIET International Conference on Intelligent Environments, Ulm, Germany (2007)

[3] Heider, T., Kirste, T.: Intelligent environment lab. Computer Graphics Topics 15(2), 8–10 (2002)

[4] iTalc - (2011), http://italc.sourceforge.net/

[5] Le Gal, C.: Smart Environments: Technology, Protocols and Applications. Wiley, chap Smart Offices (2005)

[6] Marreiros, G., Santos, R., Ramos, C., Neves, J., Novais, P., Machado, J., Bulas-Cruz, J.: Ambient Intelligence in Emotion Based Ubiquitous Decision Making. In: Proc. Artificial Intelligence Techniques for Ambient Intelligence, IJCAI 2007 – Twentieth International Joint Conference on Artificial Intelligence, Hyderabad, India (2007)

[7] Marsa-Maestre, I., Lopez-Carmona, M.A., Velasco, J.R., Navarro, A.: Mobile agents for service personalization in smart environments. Journal of Networks (JNW) 3(5), 30–41 (2008)

[8] Minder, D., Marrón, P.J., Lachenmann, A., Rothermel, K.: Experimental construction of a meeting model for smart office environments. In: Proceedings of the First Workshop on Real-World Wireless Sensor Networks (2005)

[9] Ramos, C.: Ambient Intelligence Environments", Encyclopedia of Aritificial Intelligence. In: Rabunãl, J., Dorado, J., Sierra, A. (eds.) Information Science Reference (2009)

[10] Ramos, C., Marreiros, G., Santos, R., Freitas, C.: SmartOffices and Intelligent Decision Rooms. In: Handbook of Ambient Intelligence and Smart Environments (AISE), pp. 851–880. Springer, Heidelberg (2009)

[11] Want, R., Hopper, A., Falcao, V., Gibbons, J.: The active badge location system. ACM Transactions Information Systems 10(1), 91–102 (1992)