Exploiting the JaCaMo Framework for Realising an Adaptive Room Governance Application{DEMO}

Alexandru Sorici
”Politehnica” University of Bucharest,
Department of Computer Science and
Engineering, 313 Splaiul Independentei,
Bucharest, Romania
alex.sorici@gmail.com

Olivier Boissier, Gauthier Picard
ISCOD/LSTI, ENS Mines Saint-Etienne
158 Cours Fauriel
42024 Saint-Etienne, France
{boissier,picard}@emse.fr

Andrea Santi
DEIS, Alma Mater Studiorum Universita
di Bologna, Via Venezia 52, 47521
Cesena (FC), Italy
a.santi@unibo.it

Abstract

Ambient Intelligent (AmI) applications are often meant to be used in complex and highly dynamic environments and are characterized by features such as context-awareness, personalization, adaptivity and anticipation of user's desires. In this demo we focus on how the use of the high level of abstraction provided by multi agent-oriented technologies and related programming languages – and in particular of the ones rooted on a strong notion of agency – can ease the conceiving and realisation of AmI applications exhibiting such features. For doing this we present here an agile governance application, realised using the JaCaMo framework, for the dynamic management and allocation of rooms in the context of a smart co-working space.

Categories and Subject Descriptors I.2.11 [*Artificial Intelligence*]: Distributed Artificial Intelligence–Intelligent agents, Multiagent systems, Languages and structures

General Terms Design, Languages

Keywords Ambient Intelligence, Multi-Agent System programming, JaCaMo

1. Introduction

Ambient Intelligent (AmI) applications are characterized by features such as context-awareness, personalization, adaptivity and anticipation of user's desires [5]. Such properties are catered for by technologies involving dynamic and massively distributed sensors and actuators device networks, human-centric computer interfaces, dependable and secure systems, etc. Following what is stated in [5], a relevant research issue concerns how to concretely program non-intrusive AmI applications exhibiting good adaptation capabilities.

Agent-oriented abstractions and multi-agent systems are well known in literature as a programming paradigm for the realisation of complex and dynamic systems [4]. Accordingly, in this demo we exploit a newborn multi agent-oriented programming framework named JaCaMo[1], for realising an adaptive room management application [7] in the context of a smart co-working space – e.g. a school, an office building. In particular we will focus on MAS organizations [1] and related techniques to achieve a precise and efficient level of management in the considered scenario.

2. Adaptive Room Management: Scenario Description

In this demo we consider a fairly common AmI scenario: the management of a room allocation problem in the context of a smart co-working space – e.g. a school, an office building, etc. – where people can book and use rooms according to their needs and to the current occupancy schedule.

Rooms are equipped with different items – projectors, whiteboards, TV sets, etc. – and are tagged by different usage categories – meeting room, teaching room, etc. – on the base of user-defined room requirements (e.g. number of seats, equipment, etc.). Each room is augmented with a proper set of Arduino[2]-based sensors and effectors for managing: the room temperature, the lights level, the presence of equipment and people, etc.

The application has to set an autonomous and adaptive room management behaviour in accordance with: *(i)* the events that are currently held – e.g. regulating the room temperature in accordance with the number of the event's participants, automatically turning off the lights for teaching events involving a projector, etc. – *(ii)* and also on the base of (re)allocation of the rooms according to the user requests. Users – professors, engineers, etc. – can: *(i)* demand the scheduling of new events in the building, *(ii)* modify or cancel scheduled events, *(iii)* register themselves as event's participant, and *(iv)* inquiry information about scheduled events. According to the details of the request – made in term of room category, format, number of seats and required equipments – the event will be either allotted to the appropriate room – if any is available – or discarded.

The frequent changes in the activity of the people – approaching deadlines, end/start of school courses, etc. – coming in the co-working space, make it is possible to have spikes of request types – lectures, brainstorming sessions, etc. – that must be correctly handled by the system – (re)allocate rooms to host more constrained event types – at the best of its capacity.

3. The JaCaMo Framework

JaCaMo is a newborn multi-agent oriented programming framework allowing the development of MAS taking into account three different programming dimensions, namely the agent, environment,

[1] `http://www.jacamo.sourceforge.net`

[2] `http://www.arduino.cc`

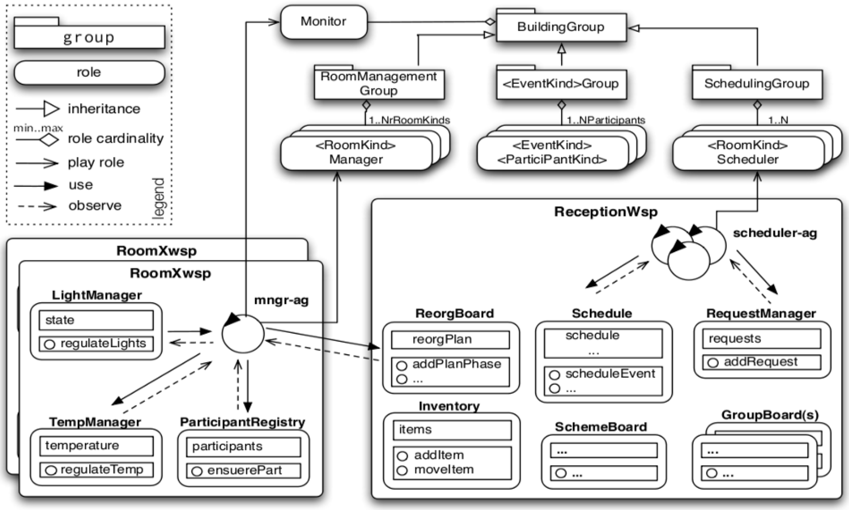

Figure 1. Architecture of the JaCaMo application for the governance of room allocation in a smart co-working space.

and organisation levels. JaCaMo is built upon the synergistic integration of three existing agent-based technologies: *(i) Jason* [2], *(ii)* $\mathcal{M}$OISE [3], and *(iii)* CArtAgO [6]. A JaCaMo multi-agent system (i.e., a software system programmed in JaCaMo) is given by an agent organisation programmed in $\mathcal{M}$OISE, organizing autonomous agents programmed in *Jason*, working in shared distributed artifact-based environments programmed in CArtAgO.

JaCaMo integrates these three platforms by defining a semantic link among concepts of the different programming dimensions at the meta-model and programming levels, in order to obtain a uniform and consistent programming model aimed at simplifying the combination of those dimensions when programming multi-agent systems.

4. The JaCaMo-based Governance Application

Figure 1 provides an overview of the realised JaCaMo application which is the subject of this demo. Organisation programming is exploited here to specify and manage the overall management strategy of the system, involving different roles (from individual room managers to schedulers), their coordination, and in particular to handle re-organisation processes required in such dynamic contexts. Adaptation is performed in an endogeneous way, whereby system agents employ an additional cooperation scheme – playing monitoring and design roles – which helps them to construct the most appropriate reorganization plan.

The artifact-based distributed environment is exploited to model and interface with the physical devices in the rooms (lights, temperature controllers, etc.), to model and represent high-level shared data structures with related operations (such as registers keeping track of room participants and schedules), besides typical coordination artifacts (such as blackboards).

Agents obviously encapsulate the control and decision-making part of the application, in this case related to monitoring and controlling facilities in rooms as well as deciding upon strategies to use for dynamic room allocation.

5. Results

In order to evaluate the developed governance application subject of this demo, we designed a simple simulator which allows to stress the adaptation capability of the realised application. The simulator allows to establish a set of settings and the desired scenario dynamics that could trigger a need for reorganization.

By carefully setting values for parameters that determine the request dynamics – i.e. distribution of requests for different event types during a given period of time – the simulation creates a form of inverse situations, whereby the demands for two compatible event types lie in opposition—e.g. a strong increase of brainstorm sessions in a given day is followed by strong increase of meetings in the next one. This simulation context drives a dynamic shift of agents' roles from the managing of one event type to the other, which constitutes an adequate test for the entire reorganization functionality implemented. The results are given under the form of charts and logs which reveal the actions undertaken by the agents of the presented MAS, their distribution to room management roles and the total number of issued and denied requests.

Acknowledgments

Part of this work is funded by the CMIRA Programme of the Rhone-Alpes Region.

References

[1] O. Boissier, J. F. Hübner, and J. S. Sichman. Organization oriented programming: from closed to open organizations. In *Proceedings of the 7th international conference on Engineering societies in the agents world VII*, ESAW'06, pages 86–105, Berlin, Heidelberg, 2007. Springer-Verlag.

[2] R. Bordini, J. Hübner, and M. Wooldridge. *Programming Multi-Agent Systems in AgentSpeak Using Jason*. John Wiley & Sons, Ltd, 2007.

[3] J. F. Hübner, J. S. Sichman, and O. Boissier. Developing Organised Multi-Agent Systems Using the MOISE+ Model: Programming Issues at the System and Agent Levels. *Agent-Oriented Software Engineering*, 1(3/4):370–395, 2007. ISSN 1746-1375.

[4] N. R. Jennings. An agent-based approach for building complex software systems. *Commun. ACM*, 44(4):35–41, 2001.

[5] P. Remagnino and G. L. Foresti. Ambient intelligence: A new multidisciplinary paradigm. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 35(1):1–6, 2005.

[6] A. Ricci, M. Piunti, M. Viroli, and A. Omicini. Environment programming in CArtAgO. In R. H. Bordini, M. Dastani, J. Dix, and A. El Fallah-Seghrouchni, editors, *Multi-Agent Programming: Languages, Platforms and Applications, Vol. 2*. Springer, 2009.

[7] A. Sorici. Agile Governance in an Ambient Intelligence Environment. Master's thesis, University "Politehnica" of Bucharest, September 2011.