LNAI 3661

Themis Panayiotopoulos
Jonathan Gratch
Ruth Aylett
Daniel Ballin
Patrick Olivier
Thomas Rist (Eds.)

Intelligent Virtual Agents

5th International Working Conference, IVA 2005
Kos, Greece, September 2005
Proceedings

Springer

Themis Panayiotopoulos   Jonathan Gratch
Ruth Aylett   Daniel Ballin
Patrick Olivier   Thomas Rist (Eds.)

Intelligent Virtual Agents

5th International Working Conference, IVA 2005
Kos, Greece, September 12-14, 2005
Proceedings

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Themis Panayiotopoulos
University of Piraeus, Department of Informatics
80 Karaoli & Dimitriou str., Piraeus, 18534, Greece
E-mail: themisp@unipi.gr

Jonathan Gratch
University of Southern California, Institute for Creative Technologies
13274 Fiji Way, Marina del Rey, CA 90292, USA
E-mail: gratch@ict.usc.edu

Ruth Aylett
Heriot-Watt University, Mathematics and Computer Science
Edinburgh EH14 4AS, UK
E-mail: ruth@macs.hw.ac.uk

Daniel Ballin
Chief Technology Office
BT Group, Room 66, B54, Adastral Park, Ipswich IP5 3RE, UK
E-mail: daniel.ballin@bt.com

Patrick Olivier
University of Newcastle upon Tyne, Informatics Research Institute
Newcastle upon Tyne NE1 7RU, UK
E-mail: p.l.olivier@ncl.ac.uk

Thomas Rist
University of Applied Sciences Augsburg
Friedberger Str. 2a, 86161 Augsburg, Germany
E-mail: tr@rz.fh-augsburg.de

Library of Congress Control Number: 2005931803

CR Subject Classification (1998): I.2.11, I.2, H.5, H.4, K.3

ISSN 0302-9743
ISBN-10 3-540-28738-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-28738-4 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11550617 06/3142 5 4 3 2 1 0

Preface

The origin of the Intelligent Virtual Agents conference dates from a successful workshop on Intelligent Virtual Environments held in Brighton at the 13th European Conference on Artificial Intelligence (ECAI'98). This workshop was followed by a second one held in Salford in Manchester in 1999. Subsequent events took place in Madrid, Spain in 2001 and Irsee, Germany in 2003 and attracted participants from both sides of the Atlantic as well as Asia.

This volume contains the proceedings of the 5th International Working Conference on Intelligent Virtual Agents, IVA 2005, held on Kos Island, Greece, September 12–14, 2005, which highlighted once again the importance and vigor of the research field. A half-day workshop under the title "Socially Competent IVA's: We are not alone in this (virtual) world!" also took place as part of this event. IVA 2005 received 69 submissions from Europe, North and South America, Africa and Asia. The papers published here are the 26 full papers and 14 short papers presented at the conference, as well as one-page descriptions of the 15 posters and the descriptions of the featured invited talks by Prof. Justine Cassell, of Northwestern University and Prof. Kerstin Dautenhahn, of the University of Hertfordshire.

We would like to thank a number of people that have contributed to the success of this conference. First of all, we thank the authors for their high-quality work and their willingness to share their ideas. We thank the Program Committee, consisting of the editors and 74 distinguished researchers, who worked hard to review the submissions and to select the best of them for presentation. A special thanks goes to the Local Organizing Committee for their efficient work on preparing and running the event. We would like to thank our sponsors for their financial support and, last but not least, we thank all those who attended the conference.

We invite readers to enjoy the papers in this book and look forward to the next Intelligent Virtual Agents conference.

July 2005

Themis Panayiotopoulos
Jonathan Gratch
Ruth Aylett
Daniel Ballin
Patrick Olivier
Thomas Rist

Committee Listings

Conference Chairs

Jonathan Gratch (USC Institute for Creative Technologies, USA)
Ruth Aylett (Heriot-Watt University, UK)
Daniel Ballin (Chief Technology Office, BT Group, UK)
Patrick Olivier (University of Newcastle Upon Tyne, UK)
Thomas Rist (University of Applied Sciences Augsburg, Germany)

Local Conference Chair

Themis Panayiotopoulos (Department of Informatics, University of Piraeus, Greece)

Organizing Committee

Nikos Avradinis (University of Piraeus, Greece)
George Anastassakis (University of Piraeus, Greece)
Spyros Vosinakis (University of Piraeus, Greece)
Aris Belesiotis (University of Piraeus, Greece)
Aris Koutsiamanis (University of Piraeus, Greece)
John Giannakas (University of Piraeus, Greece)
Kostas Tilelis (University of Piraeus, Greece)

Invited Speakers

Justine Cassell (Northwestern University, USA)
Kerstin Dautenhahn (University of Hertfordshire, UK)

Program Committee

Jan Albeck
George Anastassakis
Elisabeth André
Norman Badler
Paulo Barthelmess
Josep Blat
Joanna Bryson
Lola Canamero
Justine Cassell
Marc Cavazza
Nicolas Ech Chafai
Elizabeth Churchill
Toni Conde
Kerstin Dautenhahn
Betsy van Dijk
Patrick Doyle
Angelica de Antonio
Fiorella de Rosis
Patrick Gebhard
Marco Gillies
Erdan Gu
Bruno Herbelin

Randy Hill
Adrian Hilton
Kristina Hook
Xiao Huang
Katherine Isbister
Mitsuru Ishizuka
Ido Iurgel
Sverker Janson
Lewis Johnson
Natasa Jovanovic
Martin Klesen
Stefan Kopp
John Laird
James Lester
Craig Lindley
Zhen Liu
Brian Loyall
Nadia Magnenat-Thalmann
Andrew Marriot
Stacy Marsella
Michael Mateus
Alexander Nareyek
Anton Nijholt
Magalie Ochs
Gregory O'Hare
Sharon Oviatt
Martijn van Otterlo
Ana Paiva
Zhigeng Pan
Catherine Pelachaud
Chris Pennock
Paolo Petta
Tony Polichroniadis
Helmut Prendinger
Jaime Ramírez
Matthias Rehm
Daniela Romano
Alejandra Garcia Rojas
Zsofia Ruttkay
Wei Shao
Anthony Steed
Matthew Stone
Paul Tepper
Daniel Thalmann
Kris Thorisson
Demetri Terzopoulos
David Traum
Ana Vaz
Hannes Vilhjalmsson
John Vince
Spyros Vosinakis
Michael Young

Sponsoring Institutions

University of Piraeus Research Center
AgentLink
Hellenic Artificial Intelligence Society
Medicon SA
Focus Magazine

Table of Contents

IVAs and Education

Conversational Agents/NLP

IVA Architectures

Cognition, Reasoning and Behaviour

NonVerbal Communication

Storytelling/Interactive Narrative

Social Intelligence

Emotions/Affect/Personality

Evaluation and Methodology

Poster Session

Training Agents: An Architecture for Reusability

Gonzalo Mendez and Angelica de Antonio

Computer Science School,
Technical University of Madrid,
Campus de Montegancedo s/n, 28660 Boadilla del Monte (Madrid), Spain
gonzalo@gordini.ls.fi.upm.es, angelica@fi.upm.es

Abstract. During the last years, Intelligent Virtual Environments for Training have become a quite popular application of computer science to education. These systems involve very different technologies, ranging from computer graphics to artificial intelligence. However, little attention has been paid to software engineering issues, and most of these systems are developed in an ad-hoc way that does not allow the reuse of their components or even an easy modification of the application. We describe an agent-based software architecture that is intended to be easily extended and modified. Also, some experiments to test the suitability of the architecture are shown.

1 Introduction

Many of the advances in the application of intelligent agents to the field of Intelligent Virtual Environments for Training (IVET) have come from the Artificial Intelligence community, such as Herman the Bug [1], Cosmo [2] or Steve [3,4].

However, little effort has been devoted to software engineering issues, and in the few cases where some attention has been paid to design methods, such as in Jacob [5], they have focused in object oriented design rather than agent oriented design.

The MAEVIF (*Model for the Application of Intelligent Virtual Environments to Education*) project is the result of several experiences integrating virtual environments and intelligent tutors [6,7] that served to point out the problems that commonly arise in such integrations. The objective of the MAEVIF project was to define a model for the application of intelligent virtual environments to education and training, which involved:

- The definition of a generic model for intelligent learning environments based on the use of virtual worlds.
- The definition of an open and flexible agent-based software architecture to support the generic model of an IVET.
- The design and implementation of a prototype authoring tool that simplifies the development of IVETs, based on the defined architecture.
- The definition of a set of methodological recommendations for the development of IVETs.

T. Panayiotopoulos et al. (Eds.): IVA 2005, LNAI 3661, pp. 1–14, 2005.

In the remainder of this paper it will be described how the traditional architecture of Intelligent Tutoring Systems (ITS) [8,9] has been extended to support Virtual Environments (section 2) and how it has been transformed into an agent-based architecture (section 3). In section 4, an explanation of the functionality of the authoring tool will be given. Section 5 will present a discussion of the results that have been achieved with the MAEVIF project. Then, the basic functioning of the system will be described (section 6), and finally, in section 7, some future work lines will be shown.

2 An Extension to the Architecture of Intelligent Tutoring Systems

The development of three dimensional Virtual Environments (VEs) has a quite short history, dating from the beginning of the 90s. The youth of the field, together with the complexity and variety of the technologies involved, have led to a situation in which neither the architectures nor the development processes have been standardized yet. Therefore, almost every new system is developed from scratch, in an ad-hoc way, with very specific solutions and monolithic architectures, and in many cases forgetting the principles and techniques of the Software Engineering discipline [10]. Some of the proposed architectures deal only partially with the problem, since they are centered on a specific aspect like the visualization of the VE [11,12] or the interaction devices and hardware [13].

Our approach to the definition of an architecture for IVETs is based on the agent paradigm. The rationale behind this choice is our belief that the design of highly interactive IVETs populated by intelligent and autonomous or semi-autonomous entities, in addition to one or more avatars controlled by users, requires higher level software abstractions. Objects and components are passive software entities which are not able to exhibit the kind of proactivity and reactivity that is required in highly interactive environments. Agents, moreover, are less dependent on other components than objects. An agent that provides a given service can be replaced by any other agent providing the same service, or they can even coexist. New agents can be added dynamically providing new functionalities. Extensibility is one of the most powerful features of agent-based systems. The way in which agents are designed make them also easier to be reused than objects.

Since an IVET can be seen as a special kind of ITS, and the pedagogical agent in an IVET can be seen as an embodiment of the tutoring module of an ITS, our first approach towards defining an standard architecture for IVETs was to define an agent for each of the four modules of the generic architecture of an ITS [9] (see Fig. 1).

The ITS architecture, however, does not fit well with the requirements of IVETs in several aspects:

- IVETs are usually populated by more than one student, and they are frequently used for team training. An ITS is intended to adapt the teaching

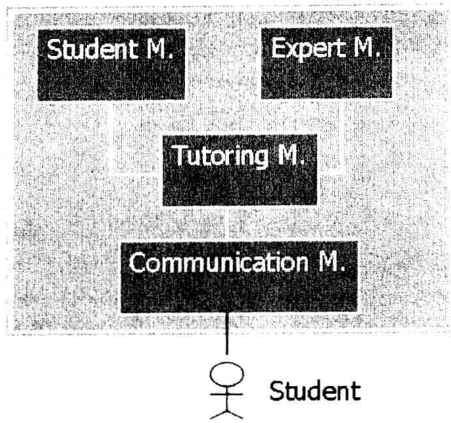

Fig. 1. Architecture of an ITS

and learning process to the needs of every individual student, but they are supposed to interact with the system one at a time. However, in a multi-student IVET, the system has to adapt both to the characteristics of each individual student and to the characteristics of the team. Consequently, the student module should model the knowledge of each individual student but also the collective knowledge of the team.

- The student is not really out of the limits of the ITS, but immersed in it. The student interacts with the IVET by manipulating an avatar within the IVET, possibly using complex virtual reality devices. Furthermore, each student has a different view of the VE depending on their location within it.
- The communication module in an ITS is usually realized by means of a GUI or a natural language interface that allows the student to communicate with the system. It would be quite intuitive to consider that the 3D graphical model is the communication module of an IVET. However, there is a fundamental difference among them: in an IVET, the learning goals are directly related to the manipulation and interaction with the 3D environment, while the communication module of a classical ITS is just a means, not an end. Therefore, the ITS needs to have explicit knowledge about the 3D VE, its state, and the possibilities of interaction within it.

As a first step we decided to modify and extend the ITS architecture by considering some additional modules. First of all, we split the communication module into a set of different views for all the students with a particular communication thread for each student, and a centralized communication module to integrate the different communication threads. Then, we added a world module, which contains geometrical and semantic information about the 3D graphical representation of the VE and its inhabitants, as well as information about the interaction possibilities. The tutoring module is unique to be able to make decisions that affect all the students, as well as specific tutoring decisions for a certain student. The expert module contains all the necessary data and inference rules to maintain a simulation of the behavior of the system that is represented through the VE (e.g. the behavior of a nuclear power plant). The student module, finally, maintains an individual model for each student as well as a model of the team.

3 An Agent-Based Architecture for IVETs

Taking the extended architecture described in the previous section as a starting point, the next step is to decide which software agents are necessary to transform this component-oriented architecture into an agent-oriented architecture, which has been designed using the GAIA methodology [14]. In this methodology, the authors suggest the use of the *organizational metaphor* to design the system architecture, which basically consists of analyzing the real-world organization in order to emulate its structure. It is mentioned that this approach does not always work (depending on particular organization conditions), but in this case, considering the extended architecture of an ITS as the real world, it seems quite appropriate to imitate its structure to develop the system architecture.

Figure 2 shows how the extended ITS architecture is transformed, from a modular point of view, into an agent-based architecture. It has five agents corresponding to the five key modules of the extended ITS architecture:

- A Communication Agent
- A Student Modelling Agent
- A World Agent
- An Expert Agent
- A Tutoring Agent

Analyzing the responsibilities of these agents, some additional roles can be identified that point to the creation of new, subordinate agents that can carry them out, subsequently giving rise to a hierarchical multi-agent architecture.

3.1 Central Communication Agent

The Central Communication Agent is responsible for the communication between the Virtual Environment and the Tutoring System. It delegates part of its responsibilities to a set of Individual Communication Agents dedicated to each student. There is also a Connection Manager Agent, which is responsible for coordinating the connections of the students to the system, and a set of Device Agents in charge of managing the data provided by the devices the students use to interact with the Virtual Environment.

3.2 Student Modelling Agent

This agent is in charge of maintaining a model of each student, including personal information, their actions in training sessions, and a model of the students' knowledge.

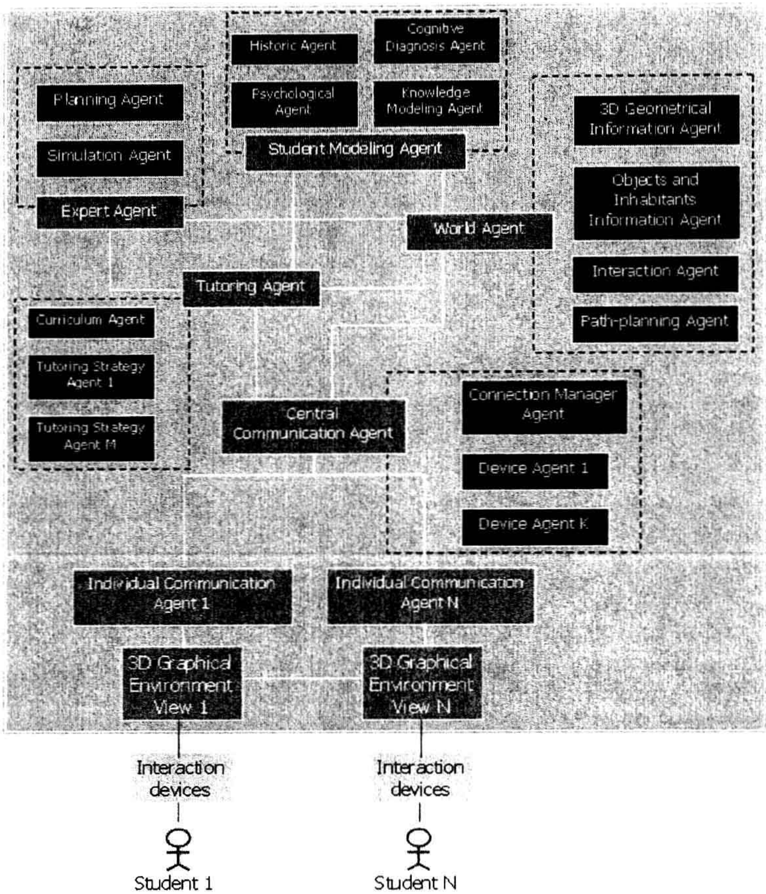

Fig. 2. Agent-based architecture

Figuring out the student's abilities and beliefs/knowledge is usually not a trivial issue. To better individualize training and appropriately understand the student's behavior, a representation of some of its personal features (personality traits, mood, attitudes,...) is defined and maintained. To do this, the Student Modelling Agent is assisted by:

- A Historic Agent, which is responsible for registering the history of interactions among the students and the system.
- A Psychological Agent, which is responsible for building a psychological profile of each student including their learning style, attentiveness, and other personality traits, moods and emotions that may be interesting for adapting the teaching process.
- A Knowledge Modelling Agent, which is responsible for building a model of the student's current knowledge and its evolution.

- A Cognitive Diagnostic Agent, which is responsible for trying to determine the causes of the student's mistakes.

3.3 World Agent

The World Agent is in charge of maintaining a coherent model of the VE, so that all the agents and students have the same information about the state of the world.

The World Agent is related to:

- The 3D Geometrical Information Agent which has geometrical information on the objects and the inhabitants of the world. Among other responsibilities, this agent will answer questions about the location of the objects.
- The Objects and Inhabitants Information Agent, which has semantic knowledge about the objects and the inhabitants of the world. This agent will be able to answer questions about the utility of the objects or the objects being carried by a student.
- The Interaction Agent, which has knowledge about the possible actions that the students can perform in the environment and the effects of these actions. It will be able to answer questions like "What will it happen if I push this button?"
- The Path-Planning Agent, which is capable of finding paths to reach a destination point in the environment avoiding collisions with other inhabitants and objects. For the purpose of finding these paths, the A* algorithm will be applied to a graph model of the environment.

3.4 Expert Agent

The expert agent contains the expert knowledge about the environment that is being simulated, as well as the expert knowledge necessary to solve the problems posed to the student and to reach the desired goals. Most of the activities to be executed by the students consist of finding an appropriate sequence of actions, or plan, to go from an initial state of the environment to a desired final state. These actions have to be executed by the team of students. The Expert Agent delegates some of its responsibilities to a Simulation Agent, that contains the knowledge about the simulated system, and a Planning Agent, that is able to find the best sequence of actions to solve different activities.

The plan for an activity is worked out by the Planning Agent with the collaboration of three other agents:

- The Path-Planning Agent can determine whether there is a trajectory from a certain point of the world to another one.
- The Interaction Agent provides information about the actions that a student can directly execute in the environment.