Corroborating Emotion Theory with Role Theory and Agent Technology: a Framework for Designing Emotional Agents as Motivational Tutoring Entities

Von der Fakultät Ingenieurwissenschaften der

Universität Duisburg-Essen

zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigte Dissertation

von

Bogdan – Florin Marin

aus

Craiova, Rumänien

Referent: Prof. Dr.-Ing. Axel Hunger

Korreferent Prof. Dr.-Ing. Hans-Dieter Kochs

Tag der mündlichen Prüfung: 17.12.2007

Acknowledgements

Undertaking a study that leads to the award of PhD is much like a journey of exploration and discovery. Although you may have some idea of where it is you wish to end up, the many rich experiences and pitfalls along the way are largely unforeseen. It is certainly an experience that I would recommend to anyone who believes they are capable. That is not to say, however, that it is a study suitable for everyone. The road to travel is long and tough, and many fall by the wayside.

My own journey has been one of academic learning and self-discovery. During my study, I have enjoyed incredibly the process of scaling new heights of knowledge, of cutting a trail where others may have never been, of using and pushing my mind to attack and answer the big questions. During this journey my mind has been refined to a sharpness and focus hitherto unforeseen to me, and I feel I am now able to wield my mind as a tool, in all situations. This has allowed me to look within, and understand exactly whom I am. In addition, my character has grown and expanded with a wealth of new experiences that have served to polish it.

I feel lucky to have undertaken my research in a relatively wide field, where the boundaries and rules have not been standardized yet. This has afforded me an academic freedom that many students do not enjoy, and allowed me to follow an academic path out of the reach of many. This type of work cannot be done alone in isolation though, and I would like to take this opportunity to thank those who have made it possible for me to get this far.

Firstly I would like to thank my family, for their continued support throughout all my years of study. Without their help, I would have been unable to complete my work. I hope my completion goes some way to repaying their love, trust and support.

To study for a PhD requires a suitable environment and support in which to do so. Most important in providing this has been my supervisors. They deserve special credit for having the patience to guide a determined and unconventional student, even when many of the proposed ideas were contrary to their own philosophies. I am sure the experience must have been challenging, but I believe that we all have learnt greatly from it.

In addition, I would like to offer my thanks to all my colleagues – the other members of the Technische Informatik Institute – for providing a stimulating social environment, and for their input or influence over the course of my study.

One cannot work on anything exclusively for so long and so hard, without the need for respite. I have many friends to who I owe thanks, who have allowed me to relax, rage, or lose myself, away from grindstone. Some deserve special mention: Rodica and Dominic Draia, Alex Grosu, Ionut Mitrache, Aurel Neata, Magdalena Dima, Claudiu Popescu.

Lastly, but most importantly I would like to thank my girlfriend, Mariana, whose unswerving love, support and companionship have allowed me to concentrate my efforts on achieving my goals. She truly is a wonderful person and I count myself extremely lucky to be with her.

Abstract

Nowadays, more and more applications require systems that can interact with humans. *Agents* can be perceived as computing services that humans, or even other agents, can request in order to accomplish their tasks. Some services may be simple and others rather complex. A way to determine the best agents (services) to be implemented is to identify who the actors are in the object of study, which roles they play, and (if possible) what kind of knowledge they use.

Socially Intelligent Agents (SIAs) are agent systems that are able to connect and interface with humans, i.e. robotic or computational systems that show aspects of human-style social intelligence. In addition to their relevance in application areas such as e-commerce and entertainment, building artefacts in software and hardware has been recognized as a powerful tool for establishing a science of social minds which is a constructive approach toward understanding social intelligence in humans and other animals.

Social intelligence in humans and other animals has a number of fascinating facets and implications for the design of SIAs. Human beings are biological agents that are embodied members of a social environment and are autobiographic agents who have a unique personality. They are situated in time and space and interpret new experiences based on reconstructions of previous experiences. Due to their physical embodiment, they have a unique perspective on the world and a unique history: an autobiography. Also, humans are able to express and recognize emotions, that are important in regulating individual survival and problem-solving as well as social interactions.

Like artificial intelligence research trend, SIA research trend can be pursued with different goals in mind. A deep AI approach seeks to simulate real social intelligence and processes. A shallow AI approach, which will be highlighted also within this thesis, aims to create artefacts that are not socially intelligent *per se*, but rather appear socially intelligent to a given user. The shallow approach does not seek to create social intelligence unless it is meaningful social intelligence vis-à-vis some user situation

In order to develop believable SIAs we do not have to know how beliefs-desires and intentions actually relate to each other in the real minds of the people. If one wants to create the impression of an artificial social agent driven by beliefs and desires, it is enough to draw on investigations on how people with different cultural background, develop and use theories of mind to understand the behaviours of others. Therefore, SIA

technology needs to model the folk-theory reasoning rather than the real thing. To a shallow AI approach, a model of mind based on folk-psychology is as valid as one based on cognitive theory.

Distance education is understood as online learning that is technology-based training which encompasses both computer-assisted and Web-based training. These systems, which appear to offer something for everyone at any time, in any place, do not always live up to the great promise they offer.

The usage of social intelligent agents in online learning environments can enable the design of “enhanced-learning environments” that allow for the development and the assessment of *social competences* as well as the common *professional competences*.

Within this thesis it is shown how to corroborate affective theory with role theory with agent technology in a synchronous virtual environment in order to overcome several inconveniences of distance education systems. This research embraces also the shallow approach of SIA and aims to provide the first steps of a method for creating a believable life-like tutor agent which can partially replace human-teachers and assist the students in the process of learning. The starting point for this research came from the fact: anxious, angry or depressed students do not learn; people in these conditions do not absorb information efficiently, consequentially it is an illusion to think that learning environments that do not consider motivational and emotional factors are adequate.

Table of Contents

List of Figures .... *v*

List of Tables .... *vii*

List of Abbreviations .... *viii*

1. **INTRODUCTION 1**

1.1. Motivation for Research..1

1.2. Challenges and Research Opportunities...3

1.3. Other Research Areas 5

1.3.1.*CSCW* 5

1.3.2.*Intelligent Tutoring Systems*...6

1.4. Research Contributions.. …7

1.5. Thesis Outline...8

2. INTRODUCING PEDAGOGICAL AGENTS…….……………………………10

2.1. Introducing Software Agents' Technology ………………………………...…..10

2.1.1. *Software Agents: Definitions and Terminology* …………………10

2.1.2. *Reasoning for Agents: Agents as Intentional Systems* …………..13

2.1.3. *Reasoning for Agents: BDI Foundations* ………………………..17

2.2. Background of Intelligent Tutoring Systems…………………………...…18

2.2.1. *The Key Components of ITS* ……………………………………..21

2.2.2. *Review of ITSs Technology Research* …………………………...22

2.3. Why Software Agents as Tutors?.…………………………………………24

2.4. Agent-based Intelligent Tutoring Systems: A Review………………………….28

3. MODELING SOFWTARE AGENTS: A NEW APPROACH………………32

3.1. Agent oriented Methodologies: State of the Art………………………………..32

3.1.1. *GAIA*……………………………………………………………..33
3.1.2. *AAII*……………………………………………………………....33
3.1.3. *SODA*…………………………………………………………….34
3.1.4. *Tropos*……………………………………………………………35
3.1.5. *OperA*…………………………………………………………..…36
3.1.6. *MaSE*……………………………………………………………..37
3.2. Issues in Modeling Agent Systems…………………………………………...…38
3.3. A New Approach for Modeling Pedagogical Agents……………………….....…39

4. MODELING PEDAGOGICAL AGENTS: REQUIEREMENTS GATHERING ……………………………………………………………..42

4.1. Requirements for a Complete Model of Collaborative Virtual Learning Environments …………………………………………………………...……42
4.1.1. *A Model for Collaborative Virtual Environments* ………………43
4.1.2. *A Metamodel of Human Collaboration* …………………………44
4.1.3. *Requirements of Synchronous Collaborative Learning Environments*………………………………………………………………48
4.2. Basic Requirements for a Pedagogical Agent in a CVE ………………...……...51
4.2.1. *Requirements for autonomous agents*……………………………51
4.2.2. *Requirements for agents as ITS*…………………………………52
4.2.3. *Requirements for believability*………………………...………54
4.2.4. *Requirements for cultured embedded agents*……………….………56

5. REQUIEREMENTS ANALYSIS PHASE……………………………………60

5.1. A Role Model…………………………………………………………...…60
5.1.1. *Roles and Role Models: Background Theory*……………………61
5.1.2. *A Role Metamodel for Pedagogical Agents* ……………………62
5.1.3. *Model of Follow-up of a Collaborative Activity*………………...65
5.1.4. *Modeling Tutoring Agents using AORML*…………………….……69
5.2. Analyzing the Social Model of Tutors…………………………………......…72
5.2.1. *Introducing Social Intelligence*…………………………………..76

5.2.2. *Understanding Social Intelligence*……………………………….78
5.2.2.1.Culture ……………………………………………………………….83
5.2.2.2.Emotion Theory……………………………………………………...86
5.2.2.3.Millon’s Personality Theory……………………………………...….88

6. MODELING PEDAGOGICAL AGENTS: DESIGN PHASE………………91

6.1. Principles of an Emotional/Motivational Framework for Pedagogical Agents...92
6.2. An Analytical Approach for Emotion Modelling: OCC Theory………………..97
6.2.1.*Adapting OCC Theory*…………………………………………………...99
6.3. Modelling Emotions using Dynamic Belief Networks………………………...102
6.3.1.*Emotion Triggering with Dynamic Belief Networks*……………………104

7. ARCHITECTURAL MODEL OF THE PEDAGOGICAL AGENT…………109

7.1. Control Process for Pedagogical Agents………………………………………114
7.2. Student Model…………………………………………………………………115
7.2.1. *Introducing Fuzzy Cognitive Maps*……………………………116
7.2.2. *Modeling Student Profile using Fuzzy Cognitive Maps*……..…118
7.2.3. *State of the Art: Overview of Symbolic Machine Learning*….…121

8. PUTTING THEORIES TO PRACTICE: PASSENGER AGENT TUTOR ……………………………………………………………………………………124

8.1. Synchronous Groupware PASSENGER…………………………………….124
8.2. Requirements for an Agent-based Tutor for the PASSENGER Groupware…..126
8.3. A MAS for the PASSENGER Groupware…………………………………….128
8.3.1. *Tutor Agent*……………………………………………………………..129
8.3.2. *Mobile Agent*……………………………………………………………131
8.3.3. *Coordinator Agent*……………………………………………………...133
8.4. Experimental Results: A Passenger Learning Scenario………………………..134
8.5. Evaluation…………………………………………….………………………..136

9. CONCLUSIONS……………………………………………………………..138

9.1. Main Contributions of this Research……………………………………..138

9.2. Future Work…………………………………………………………………141

9.3. Final Remarks……………………………………………………………….143

10. LIST of REFERENCES ……………………………………………………….144

11. APPENDIX 1: LIST of PUBLICATIONS ………………………………………161

List of Figures

Figure 2-1 *BDI Interpreter from [Rao95]* *18*

Figure 2-2 *Tree Structure of a CAI system* *19*

Figure 2-3 *Major ITS Components* *21*

Figure 3-1 *Software development life cycle* *40*

Figure 4-1 *A Metamodel of Collaborative Processes from [Haw02]* *46*

Figure 4-2 *Generic architecture for a synthetic agent model* *59*

Figure 5-1 *Model of an agent in a group* *60*

Figure 5-2 *BNF specification of Role class* *63*

Figure 5-3 *Decomposition of the follow-up of learners engaged in a collective activity* *66*

Figure 5-4 *Follow up of Collective Activity from [Lap04]* *67*

Figure5-5 *Functional clover of collaborative learning environments from [Ell94]* *68*

Figure5-6 *The core elements of AOR Modeling* *70*

Figure5-7 *Scenario Solution A* *70*

Figure5-8 *Scenario Solution B* *71*

Figure5-9 *Emotional Structure [Gol95]* *75*

Figure5-10 *Gestures for: a) "thinking", b) "wait to take turn", c) "big difference", d) "explaining"* *79*

Figure 6-1 *Emotionally Oriented Intelligent Agent Architecture* *91*

Figure 6-2 *Animations for confuse and deny* *95*

Figure 6-3 *Feelings recognition module* *96*

Figure 6-4 *The transfer function (transfers emotions potential to intensities) [Pic97]* *99*

Figure 6-5 *Emotions' display: distress and joy* *102*

Figure 6-6 *Generic Structure for a typical DBN [Nic94]* *104*

Figure 6-7 *DBN in monitoring system [Nic94]* *105*

Figure 6-8 *Mental Model of Agent* ... 105

Figure 6-9 *DBN for Happy-for and Sorry-for* ... 107

Figure 7-1 *Architecture for an emotional pedagogical agent* ... 110

Figure 7-2 *Sample activity tree* ... 112

Figure 7-3 *Control process of the pedagogical agent* ... 115

Figure 7-4 *Map for detecting motivation in learner [Geo03]* ... 119

Figure 8-1 *Passenger Client* ... 125

Figure 8-2 *Knowledge Levels [Mar04]* ... 127

Figure 8-3 *Structure of Passenger System* ... 128

Figure 8-4 *Passenger Agent Tutor* ... 129

Figure 8-5 *Mobile Agent UMLDiagram* ... 132

Figure 8-6 *How the mobile agent travels for answers* ... 133

Figure 8-7 *Coordinator Agent* ... 134

Figure 8-8 *Task partial solution: Context Diagram* ... 135

Figure 8-9 *Acceptance of Tutor Agent* ... 136

List of Tables

Table 4-1 *Primary services necessary in a synchronous collaborative learning environment......50*

Table 7-1 *Middle layer of basic learner [Kol84] 117*

Table 7-2 *The compression of some published student modeling systems 122*

List of Abbreviations

3APL	An Abstract Agent Programming Language (triple a-p-l)
ACC	Agent Coordination Context
AOR	Agent-Object-Relationship
AORML	Agent-Object-Relationship Modeling Language
BDI	Belief Desire Intention
BNF	Backus-Naur Form
CAG	Coordination Agent
CAI	Computer Assisted Instruction
CSCW	Computer Supported Cooperative Work
CVE	Collaborative Virtual Environment
DBN	Dynamic Belief Network
DLE	Distance Learning Environment
ER	Entity Relationship
ECA	Embodied Conversational Agent
FCM	Fuzzy Cognitive Map
FIPA	Foundation for Intelligent Physical Agents
FSA	Finite State Automat
GA	Global Agent
GSS	Group Supporting System

HCI	Human Computer Interaction
ICT	Information and Communication Technology
IE	Intelligent Entity
IERS	Intelligent Entity in a Running State
ITC	Instructional Technology Council
ITS	Intelligent tutoring System
LAF	Learning Activity Factor
LC	Learner Characteristic
LP	Learner Profile
MAS	Multi Agent System
MaSE	Multiagent Systems Engineering Methodology
NFSA	Nondeterministic Finite State Automat
OCC	Ortony – Clore – Collins
PAT	Passenger Agent Tutor
PFC	Passenger Floor Control
SIA	Socially Intelligent Agent
SML	Supervised Machine Learning
UDE	University of Duisburg-Essen
UML	Unified Modeling Language
WWW	World Wide Web

Chapter 1: Introduction

1.1. Motivation for Research

The rapid progress in Information and Communication Technology (ICT) brings significant influences as well as a lot of new opportunities in higher education systems. As a new means to provide educational contents to students anytime and anywhere, distance education system is studied from all aspects in [Fis98], [Haz98]. Nowadays, many distance education systems have been implemented and are used on the basis of various individual purposes in universities, schools and other educational facilities.

The Instructional Technology Council [ITC] defines distance education as "*the process of extending learning, or delivering instructional resource-sharing opportunities, to locations away from a classroom, building or site, to another classroom, building or site by using video, audio, computer, multimedia communications, or some combination of these with other traditional delivery methods*." Shih claims that the term distance learning may be used interchangeably with distance education or may refer to "the desired outcome" of distance education [Shi03]. Distance education and e-learning are identified as similar terms for a trend of modern education.

Hentea [Hen03] believes that distance learning and traditional class-room learning are not mutually exclusive. Each has *pros* that can be maximized and *cons* that can be minimized by combining them, an approach known as blended learning. The term blended learning or hybrid learning refers to learning environments that combines aspects of online and face-to-face instruction. Ideally, hybrid learning combines the best of both worlds: the social support of classroom learning and the flexibility of distance learning outside the classroom. The elements of distance education include policy, people, and technologies. The technologies used in distance education can be classified as the following:

- Communication technologies that include computer and network infrastructure (hardware and software), broadband, wireless, multimedia, distributed systems, and mobile systems.

- Intelligent technologies that include intelligent tutoring, artificial neural networks for behavior analysis, authentication mechanisms, soft computing, and visual computing.

- Educational technologies that include practical and new learning models, automatic assessment methods, effective and efficient authoring systems.

According to Hentea [Hen03], distance education is understood as online learning that is technology-based training which encompasses both computer-assisted and Web-based training. These systems, which appear to offer something for everyone at any time, in any place, do not always live up to the great promise they offer.

Current distance and open learning devices attempt to mitigate the difficulties encountered by learners when they follow a distance course. Then, it is necessary to take account of these difficulties when distance learning is set up, avoiding insulation and a lost of motivation by learners that are the cause of many giving up [Ren03].

Two major inconveniences can be noticed in distance education: the first one is laboratory experimentation/ practice. Usually, during these experimentations students have to be physically present in the university laboratories. A solution to avoid this disadvantage is virtual experimentation: the experiments are simulated and visualized by means of virtual reality [Sch99].

In local laboratory experiments, students usually work together in groups of two or more. This learning paradigm is often called collaborative learning. One solution for this problem is usage of virtual collaborative environments which bring together users who are geographically distributed but connected via a network. Therefore the students can be trained using the virtual lab concept to work in spatially distributed teams. The virtual laboratory concept is quite general encompassing a range of technologies and human factors that are necessary for operation in any remote environment, whether remote in distance, time or scale. The development and the setup of such a lab require resources as follows:

- Technologically mediated communication channel
- Shared workspace for a group
- Personal workspace
- Learning materials/ learning tools

The second issue in distance education is tutor's difficulties when he follows a distance collaborative learning process, and in particular those participants who cannot keep up progress with their group mates. A solution for this inconvenient can be intelligent software agents which can partially replace a human-tutor in the collaborative learning process of distributed student-teams.

Agents can be perceived as computing services that humans, or even other agents, can request in order to accomplish their tasks. Some services may be simple and others rather complex. A way to determine the best agents (services) to be implemented is to identify who the actors are in the object of study, which roles they play, and (if possible) what kind of knowledge they use.

We believe that the use of intelligent agents applied to online learning environments can enable the design of “enhanced-learning environments” that allow for the development and the assessment of *social competences* as well as the common *professional competences*. Examples of *social competences* include presenting ideas in a workgroup, providing and receiving criticism, cooperating with others, and behaving ethically in one’s working life.

Thus, when designing such an environment, the developers should consider the agents as integrating three kinds of services:

1. Helping people to perform innovative activities (i.e., educators need to create groups, projects, assessment portfolios; students have to relate the solutions they create to the problems proposed, to negotiate with other students, to collaborate with them, and to criticize or judge their peers’ work)

2. Stimulating social behavior within students (i.e., if the system determines that two students are working on similar issues, it can inform the students and give them information about how to contact each other)

3. Offering the educators clear and objective information about the students’ performances (i.e., which students are more creative, who effectively produces what, which students cannot collaborate, which students have to improve their reasoning skills)

We consider the use of intelligent agents as being a good approach for building collaborative online learning environments, because these agents can collect huge amounts of data regarding students’ interactions and present these data in a way that allows students and teachers to visualize what is going on and plan what to do. Students can plan their contributions for their learning session in which they are participating, and educators can plan how to conduct the learning processes.

The use of animated agents in such environments as a tutoring paradigm can be benefic and increase the learners’ motivation. Lester [Les97] investigates the impact of animated agents along the dimensions of motivation and helpfulness in an interactive learning environment. He coins the notion of ‘*persona effect*’ as “[…] the presence of a life-like character in an interactive learning environment – even one that is not expressive – can have a strong positive effect on student’s perception of their learning experiences”

1.2. Challenges and Research Opportunities

At present, educational agents exist in academic and commercial laboratories but are not widely available in real-world applications. To make the move from the laboratories to real-world applications happen, a number of technological issues for research and development needs to be solved:

First, if agent technologies are to be effective, software engineering issues need to be carefully considered:

- How can multi-agent architectures designed for maximum effect?
- Can such architectures be used effectively to support and enhance existing work practices?
- What kinds of agents and MASs are effective?
- How can such systems be designed to successfully *complement* people's existing practices and preferences?
- On which conceptual design approach should the agents be based?
- How should we design the functionality and human-agent interaction in distributed-learning environments?
- How should we design an experimental study to assess the impact of pedagogical agents on these environments?

Second, we need to increase the quality of agent software to industrial standards and provide effective agent standards to allow open system development.

Third, in addition to standard language and interaction protocols, agent societies for distributed learning will require the ability to collectively evolve language and protocols specific to distributed-learning applications and to the agents involved.

Fourth, we need to have a greater understanding of how agents for distributed learning and educational resource information systems interact.

Fifth, we need Web standards that enable structural and semantic description of information access at a higher level.

Sixth, we need to create common ontologies, thesauri, and knowledge bases, formally describe information, and potentially have a reference architecture to support the higher-level services.

Seventh, we need to develop agents' ability to understand learners' and educators' requirements and to adapt to changes in distributed learning environments.

Finally, we need to ensure confidence and trust in agents. A user must have confidence that an agent or agent system, which represents them within an open system, will act effectively on his or her behalf—it must be at least as effective as the user would be in similar circumstances. Moreover, agents must be secure and tamper-proof and must not reveal private information inappropriately. Besides, if a user is to trust the outcome of an open agent system, the user must have confidence that agents representing other parties or organizations will behave within certain constraints. These and other questions related to the maintenance, and cost of intelligent learning environments are dominating most of the researchers in this field.

As a valuable first step towards meeting these challenges, we propose the development of an explanatory framework within which to explore and describe the human actions and mental states we hope to emulate. Using this framework we can then start to develop an understanding of the architectural requirements that underlie such mentalistic terms as *motives*, *goals*, *intentions*, *attitudes*, *standards*, *emotions*, *personality, behaviour,* and how they relate to reactive and resource-bounded practical reasoning. Finally, by building complete agents, and testing them in realistic scenarios, we will then be in a position to start to learn how these mentalistic control states interact.

The research described within this thesis takes a number of decisive steps towards developing such a framework, and an understanding of the architectural requirements and design trade-offs that underlie some of our more common mentalistic terms and concepts.

The aim of this research is to discuss the role of agents not only in the enhancement of existing processes but also as a framework with which to design new processes. This thesis focuses on using agents to implement a learning environment that enables its human users to develop social competences rather than just technical ones. Next section will briefly highlight the other two research areas with their research questions.

1.3. Other Research Areas

1.3.1 CSCW

The beginning of the interdisciplinary research field CSCW leads back to the early 60s. Ellis gave one of the most frequently used definitions of CSCW:

> "CSCW looks at how groups work and seeks to discover how technology (especially computers) can help them work." [Ell91].

As CSCW describes the research fields, the term "groupware" identifies the systems (mainly software) that support the distributed teams in their cooperative actions. In 1988 Johansen gave first definition of the term "groupware":

> "Groupware is a generic term for specialized computer aids that are designed for the use of collaborative work groups. Typically, these groups are small project-oriented teams that have important tasks and tight deadlines. Groupware can involve software, hardware, services and/or group process support." [Joh98]

In CSCW research, *awareness,* which can increase communication opportunities in a distributed workspace, is one of the most exciting topics. Dourish and Bellotti [Dou92] defined awareness as: understanding of the activities of others, which provides a context for your own activity. This context is used to ensure that individual contributions are relevant to the distributed group's activity as a whole, and to evaluate individual actions with respect to group goals and progress. They further explained that the context is used to ensure that individual contributions are relevant to the group's activity as a whole and to evaluate individual actions with respect to group goals and progress. The information, then, allows groups to manage the process of collaborative working. Awareness information is always required to coordinate group activities.

Coordination, along with communication, is one main component of collaboration. Malone and Crowston [Mal94] described coordination theory as a research area focused on the interdisciplinary study of how coordination can occur in diverse kinds of systems. They also proposed an agenda for coordination research, where "designing new technologies for supporting human coordination" is considered to be one of the methodologies useful in developing coordination theory. In CSCW, understanding how computer systems can contribute to reducing the complexity of coordinating cooperative activities has been a major research issue and has been investigated by a range of eminent CSCW researchers (Carstensen & Sørensen, [Car96]; Divitini [Div96]; Malone [Mal97]).

Another absorbing topic in the CSCW Coordination research field is *floor control*, which is, according to Dommel [Dom99], a component of group coordination support that prevents or resolves resource contention. Myers [Mye93] defined floor control as the protocol that determines which user has control and how to take turns when multiple people share a limited resource such as a single cursor in a synchronous task.

In CSCW, facilitation was studied in group supporting systems (GSSs) (Hirokawa & Gouran, [Hir89]; Pollard & Vogel, [Pol91]; Antunes & Ho, [Ant99]). The activities of the facilitator in supporting group work have been identified. They are, among others, ensuring member identity and maintaining a discussion focus and a procedure for that focus; ensuring everyone has an opportunity to contribute to the discussion and decision regarding focus, procedures and decision issues; providing structure to focus group limits and boundaries; intervening when appropriate; and maintaining awareness of own feelings as an indicator (Chilberg, [Chi89]; Shelli & Hayne, [She92]).

Agent technology has been used in CSCW environments for some time, and a number of agents and MASs have been designed specifically for facilitation purposes. Related work of this research field, is COOPDRAW [Ram93], which can be considered one of the first multi-agent systems within a synchronous groupware. Ramstein [Ram93] briefly presents the problems involved in the software architecture of a synchronous groupware and the COOP environment – a shared editor for structured graphics and goes on to describe the development of COOPDRAW.

Bergenti [Ber00] describes an agent-based CSCW system designed to promote the productivity of distributed meetings by means of agents. The system architecture assigns a personal agent to every meeting participant and includes a project-manager agent and a resource manager agent to take care of the activities related to arranging and managing the meetings.

1.3.2 Intelligent Tutoring Systems

If designed properly and efficiently, intelligent tutoring systems can be of great benefit to distributed learning systems. Effective intelligent tutoring systems will make sure learners are given proper one-to-one instruction and attention to reach mastery as described by Bloom [Blo84] in his *two-sigma* article. The development of intelligent tutoring systems must include a team consisting of members from different disciplines to make sure that all of the expertise required for the one-to-one instruction is present.

Apart from research on how to develop intelligent tutoring systems, research must be completed in other areas to continually improve distributed-learning systems. One important area of research is to determine how an intelligent tutoring system can be used for individualized learning as well as collaborative learning. This area of research is important, because some learning styles prefer collaborative learning and some content areas require collaborative learning to achieve the learning outcomes for courses. Wasson's [Was00] research in this area is timely. She is looking at desired relationships between people, tools, and tasks that can stimulate collaborative behavior, such as genuine interdependence and the use of intelligent agents in collaborative learning. Another important area of research is how to design intelligent tutoring systems using constructivist's learning principles, where students are active, and they construct their own meanings and knowledge from the information presented during the distributed learning session.

The challenge for researchers and scientists is how to develop intelligent tutoring systems for distributed learning to duplicate the human tutor expertise in a one-to-one learning environment. Ally [All00] conducted a study to determine what tutors do when they tutor students in a one-to-one environment using distance education. The information from this study could be expanded and utilized to build intelligent tutoring systems so that they can become closer to and start to behave like human tutors. Also, more research and development are needed to investigate how intelligent tutoring systems can show affect, emotions, and have a sense of humor, similar to human tutors, during the tutoring process.

1.4. Research Contributions

The research presented in this thesis is based on a set of publications (see Appendix 1) and makes a number of contributions to the field of designing pedagogical agents. A briefly overview of these contributions is given below in chronological order –as they appear in this thesis.

1) Specifically, we: (a) present a design-based research methodology, and describe how it can be used to provide a powerful explanatory framework for elucidating complex systems such as intelligent pedagogical agents (b) describe how viewing the human mind as a complex control system allows the use of certain mentalistic terms and concepts to be justified by referring them to information-level descriptions of the

underlying architecture (c) introduce the concept of motivational control states and describe the *functional* attributes of some of the many control states that are likely to play an important role in intelligent autonomous agency architecture, and finally (d) describe a cognitively inspired architectural framework for elucidating the *emotional*, and *functional* attributes of these control states.

2) We argue for a concern-centric stance to autonomous agent design and provide a design-based analysis of *motivational* control states in both deliberative and behaviour-based agent methodologies – Chapter 3. We identify a number of problems with these designs and we address these problems with our design for an intelligent autonomous agent in Chapter 4-8.

3) We provide a framework for analysis of emotional agent designs based on levels of social intelligence Chapter 5. We extended this framework by adding a new level: The Cultural Background. We identified this level as a need for believability in pedagogical agents and not only.

4) We provided a metamodel of a collaborative environment for a better understanding of role requirements for tutoring agents performing activities in such environments. (Chapter 4). Chapter 5 highlights how to model these requirements using AORML for a better analysis.

5) The human emotion process can be viewed as a classic example of Dynamic Networks Beliefs. We investigate the reasons for emotions and also how emotions are triggered. Chapter 6 adds depth to the motivated agent framework by making explicit the emotions triggering and reasoning process.

6) Finally, we present our abstract design for a motivational pedagogical agent Chapter 7 – built on the lessons learnt from Chapters 3 through Chapter 6.

1.5. Thesis Outline

This thesis is presented in the engineering style of the “design-based” research methodology [Slo93] to guide the reader towards a greater understanding of the types of mechanisms that render the *behaviour* of emotional pedagogical agents.

Chapter 1 The first chapter provides a general introduction into the problem area by establishing: the research objectives.

Chapter 2 The second chapter presents the main strands of our research. We introduce the idea of a mind as an information-processing control system, and identify some of the control states that are likely to play an important role in intelligent autonomous agency. We also take the first steps towards this by analysing agents as intentional systems.

Chapter 3 The third chapter provides an overview of existing design-based approaches for autonomous agent. We argue that due to the existence of many of the identified weaknesses in existing designs, these approaches cannot be addressed for solving this current research issues.

Chapter 4 The fourth chapter gathers the complete requirements for building motivational agent based tutoring systems. This chapter forms the initial design specification for agent's architecture to meet the basic requirements of intelligent autonomous agency.

Chapter 5 The fifth chapter introduces role theory and agent object relationship modelling for a better understanding of pedagogical requirements of our prototype. This chapter also presents an information-level design-based framework analysis of the phenomena we commonly call emotion. We then extend our analysis by mapping a new level the *Cultural Awareness* for a better understanding of emotions.

Chapter 6 The sixth chapter presents how to model emotions using two different theories: OCC emotion theory and Dynamic Belief Networks.

Chapter 7 The seventh chapter presents our abstract design for an motivational pedagogical agent Chapter 7 – built on the lessons learnt from Chapters 3 through Chapter 6

Chapter 8 The eighth chapter presents an implementation of our agent design. We also present the first experimental steps and how to address some of the architectural requirements needed to support basic human emotions.

Chapter 9 Chapter nine summarises the contributions this research, and points to new directions in which the research can be taken in the future.

Chapter 10 Chapter ten provides a list of references.

Chapter 2: Introducing Pedagogical Agents

Even while it changes, it stands still
Heraclitus

Pedagogical agents are defined, according to Johnson et al.[Joh00] as "autonomous and/or interface agents that support human learning in the context of an interactive learning environment." They are built upon previous research on intelligent tutoring systems (ITSs) [Wen87]. Many researchers have designed and developed pedagogical agents for Intelligent Tutoring Systems –ITSs – [Joh97], [Cas00], where the agents play the role of a guide or tutor. The ITSs can conduct learner analysis based on initial interaction with the learner; adapt the instruction to meet the student learning style; monitor the learner's progress, providing declarative knowledge when required; decide on the best way to present the next problem or instructional sequence [Hef98]; diagnose problems and provide corrective feedback; and oversee the successful completion of the learning process. This chapter describes the details of the components of an intelligent tutoring system like also the research trends in the area of intelligent tutoring systems for distributed learning. Before the details of intelligent tutoring systems are covered, it is important to discuss basic concepts of intelligent agents, because an intelligent tutoring system is considered to be an intelligent agent system

2.1. Introducing Software Agents' Technology

In the last few years, software agents' technology has come to the forefront in the area of research interest. As agent technology has matured and become more accepted, agent-oriented software engineering has become an important topic for many researchers who wish to develop reliable intelligent agents [Jen00]. The concept of *agent* has become important in both Artificial Intelligence and mainstream computer science. It derives from the concept of *agency*, which is to employ someone (like a theatrical agent) to act on your behalf.

2.1.1. Software Agents: Definitions and Terminology

An obvious way to start this chapter would be by presenting a definition of the term *agent*. Sadly, there is no universally accepted definition of the term agent, and indeed there is much ongoing debate and controversy on this subject. Nevertheless, some sort of definition is important – otherwise, there is danger that the term will lose all meaning. Several definitions have been given to the notion of agent.

> "An agent is anything that can be viewed as perceiving its environment through sensors and acting on that environment through effectors." [Rus95].
>
> "Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed" [Mae95].
>
> "An intelligent agent is software that assists people and acts on their behalf. Intelligent agents work by allowing people to delegate work that they could have done to the agent software. Agents can, just as assistants can, automate repetitive tasks, remember things you forgot, intelligently summarize complex data, learn from you, and even make recommendations to you" [Gil97].
>
> "(An agent is) a piece of software that performs a given task using information gleaned from its environment to act in a suitable manner so as to complete the task successfully. The software should be able to adapt itself based on changes occurring in its environment, so that a change in circumstances will still yield the intended result" [Her97].
>
> Despite the variety of definitions, intelligent agents continuously perform three functions: perception of dynamic conditions in the environment, action to affect conditions in the environment, and reasoning to interpret perceptions, solve problems, draw inferences, and determine actions [Hay95].

According to Wooldridge and Jennings [Jen95]:

> "An *agent* is a computer system that is *situated* in some *environment*, and is capable of flexible, *autonomous action* in that environment in order to meet its design objectives."

Here, the flexibility characteristic means that the agent is reactive, pro-active and social.

Briefly, a *software* agent can be seen as an independently executing program able to handle autonomously the selections of actions when expected or limited unexpected events occur. Many researchers consider agent technology as the translation of social theories into computer programs [Ekd99].

Even though there are several approaches to defining agents in the literature, only two of them appear to be more relevant than the others:

1. The software engineering approach emphasizes the significance of application-independent high-level agent-to-agent communication as a basis for general software interoperability. E.g., in [Gen94], the following definition of agents is proposed: "An entity is a software agent if and only if it communicates correctly in an agent communication language."

2. The mentalistic approach, based on the knowledge representation paradigm of AI, points out that the state of an agent consists of mental components such as beliefs,

perceptions, memory, commitments, expectations, goals and intentions, and its behavior is the result of the concurrent operation of its perception (or event handling) system, its knowledge system (comprising an update and an inference operation), and its action system (responsible for epistemic, communicative and physical actions and reactions). E.g., in the approach of [Sho93], "an agent is an entity whose state is viewed as consisting of mental components such as beliefs, capabilities, choices, and commitments."

Essentially, while there is little agreement concerning the definition of an agent, there is a general consensus regarding several key characteristics of agents like *autonomy*, *proactivity*, *situatedness*, and *interactivity*. More characteristics could be added, such as mobility, locality, openness, believability, learning, adaptation capabilities, comprehensibility, etc. It must be emphasized that not all agent implementations incorporate all these features. Rather, following characteristics may illustrate potential frameworks for agent-based applications:

- *autonomy* is the ability of an agent to operate without the direct intervention of humans or others, and have some kind of control over its actions. Agents have internal state and knowledge about their own actions. This knowledge is either explicitly specified or it can be provided implicitly through information on how and where to obtain the relevant knowledge.

- *social ability* represents the possibility to interact with humans or with other agents via some agent-communication languages.

- *reactivity* represents the ability of the agent to perceive their environment and respond to the changes that occur in it.

- *pro-activeness* is the ability of an agent to exhibit goal-directed behavior by taking the initiative.

- *mobility* is the agent's ability to travel through a network in order to achieve its goal.

- *reflectivity* represents the ability of an agent to monitor its own behaviour and modify it in case of environmental changes.

- *beliefs, desires, intentions* agents possess human characteristics besides knowledge. BDI agents will be described later in this chapter.

A common classification scheme of agents is the weak and strong notion of agency [Woo95]. In the *weak notion of agency*, agents have their own will *(autonomy)*, they are able to interact with each other *(social ability)*, they respond to stimulus *(reactivity)*, and they take initiative *(pro-activity)*. In the *strong notion of agency* the weak notions of agency are preserved, in addition agents can move around *(mobility)*, they are truthful *(veracity)*, they do what they're told to do *(benevolence)*, and they will perform in an optimal manner to achieve goals *(rationality)*.

Summarizing, an agent need to have computational abilities (reasoning, searching, etc) and can use its knowledge and rationality models to map inputs to outputs that would maximize its utility (its performance measure according to the rationality). According to the interaction strategy that is used, an agent could be *cooperative*, *self-interested*, and *hostile*. Cooperative agents could work together with other agents and humans with the intention of solving a joint problem. Self-interested agents try to maximize their own goods without any concern for the global good, and will perform services for other agents only for compensation (e.g. in terms of money). Hostile agents have a utility that increases with their own gains, and increases also with the competitor's losses.

Starting from a simple comparison between a human agent-which is a person who acts autonomously and behaves intelligently- and a software agent several open problems can be noticed:

- Agents as intentional systems: is it legitimate or useful to human attributes like beliefs, desires to artificial agents? Being an intentional system seems to be a necessary condition for an agent, but is it a sufficient one?
- Knowledge a pre-condition for actions: what an agent needs to know in order to perform several actions?
- Realism of an agent: how an agent's decisions about the future and actions affect its goals for which it has been developed?
- When building intelligent agents it is important that a rational balance is achieved between the complexity and the goals of an agent?

2.1.2. Reasoning for Agents: Agents as Intentional Systems

When explaining human activity, it is often useful to make statements such as the following:

> Janine took her umbrella because she *believed* it was going to rain.
>
> Michael worked hard because he *wanted* to possess a car.

These statements makes use of a *folk psychology*, by which human behaviour is predicted and explained through the attribution of *attitudes*, such as believing and wanting (as in the above examples), hoping, fearing, and so on. This folk psychology is well established: most people reading the above statements would say they found their meaning entirely clear, and would not give them a second glance.

The attitudes employed in such folk psychological descriptions are called the *intentional* notions. The philosopher Daniel Dennett has coined the term *intentional system* to describe entities 'whose behaviour can be predicted by the method of attributing belief, desires and rational acumen' [Den87].

Dennett identifies different 'grades' of intentional system:

> 'A *first-order* intentional system has beliefs and desires (etc.) but no beliefs and desires *about* beliefs and desires. […] A *second-order* intentional system is more sophisticated; it has beliefs and desires (and no doubt other intentional states) about beliefs and desires (and other intentional states) — both those of others and its own'. [Den87, p. 243]

"Intentionality" is a philosophical term for *aboutness*. Something exhibits "intentionality" if its competence is in some way *about* something else. A thermostat is an "intentional" system – it contains representations of both the current temperature (the curvature of the bimetallic strip) and the desired temperature (the position of the dial). Autonomous agents are also "intentional" systems, but at levels of richness and complexity orders of magnitude greater than the humble thermostat. According to Dennett there is a three level intentionality's hierarchy as far as required [Den78], [Den87], [Den96]:

1) *The physical stance*. We apply the physical stance to objects when we refer our predictions to the classic laws of physics, i.e. objects fall to the ground because they are subject to the law of gravity. The physical stance affords us a great deal of confidence in our prediction.

2) *The design stance*. When we wish to understand and predict features of *design*, we need to adopt the design stance. The design stance allows us to ignore implementation details and make predictions based on *designed for* characteristics, i.e., that the alarm clock will make a loud noise at 7:15.

3) *The intentional stance*. We adopt the intentional stance whenever we treat observed systems *as if* they were rational agents who governed their "choice" of "action" by a "consideration" of their "beliefs" and "desires." The intentional stance is the most powerful, and yet the most risky of Dennett's predictive stances. Its riskiness stems from two connected problems: (i) we are non-privileged observers having to infer intention (in the philosophical sense of *aboutness*) from observed behaviour; and (ii) complex systems are inherently resource-bounded, and as such can only approximate rationality (without rationality there can be no basis for inferring intention from observed behaviour). But even with these caveats, the intentional stance is still a remarkably robust tool. It allows us to make workable predictions about the external behaviour of very complex systems such as animals and other human beings.

Dennett suggests that "*if done with care*, adopting the intentional stance is not just a good idea, but offers the key to unravelling the mysteries of the mind" [Den96, p. 27]. However, such an approach extorts a heavy price: (a) care must be taken not to confuse the philosophical term "intentionality" (*aboutness*) with the common language term referring to whether someone's action was intentional or not – as in the case of intentional control states [Bra87]; and (b) care must also be taken to recognise the limits of agent rationality. Much behaviour is simply *automatic* (neither rational nor irrational), and devoid of any form of "consideration". Such behaviour often appears rational because we are adept at spotting patterns and regularities in our environment. Some of these regularities are derived from the *designed for* characteristics of the system, be that a chess playing machine designed to win, an animal designed to carry genes from one generation to the next, or a stressed nursemaid designed to handle multiple goals. Other regularities emerge from the *physical* characteristics of the system, i.e. the resource constraints of the architecture, or the temperature of the room.

In reality, the limits of agent rationality, and the requirement of balancing multiple competing concerns in an unknowable environment, ensures that the "intentional stance" is at best a methodology of approximation rather than one of design and analysis. By assuming that systems behave *as if* they were rational agents the "intentional stance" allows us to approximate behaviour by approximating the "intentionality" (*aboutness*) of the system. However, these approximations invariably mask the real "intentionality" of the constituent components, leading to an overestimate of the complexity of the system in what Braitenberg calls the "law of uphill analysis and downhill invention" [Bra84, p. 27].

An obvious question is whether it is legitimate or useful to attribute beliefs, desires, and so on, to artificial agents. Isn't this just anthropomorphism? McCarthy, among others, has argued that there are occasions when the *intentional stance* is appropriate:

> 'To ascribe *beliefs*, *free will*, *intentions*, *consciousness*, *abilities*, or *wants* to a machine is legitimate when such an ascription expresses the same information about the machine that it expresses about a person. It is useful when the ascription helps us understand the structure of the machine, its past or future behaviour, or how to repair or improve it. It is perhaps never logically required even for humans, but expressing reasonably briefly what is actually known about the state of the machine in a particular situation may require mental qualities or qualities isomorphic to them. Theories of belief, knowledge and wanting can be constructed for machines in a simpler setting than for humans, and later applied to humans' [McC78].

Ascription of mental qualities is most straightforward for machines of known structure such as thermostats and computer operating systems, but is most useful when applied to entities whose structure is incompletely known'. [McC78]. What objects can be described by the intentional stance? As it turns out, more or less anything can.

In his doctoral thesis, Seel showed that even very simple, automata-like objects can be consistently ascribed intentional descriptions [See89]; similar work by Rosenschein and Kaelbling, (albeit with a different motivation), arrived at a similar conclusion [Ros86]. For example, consider a light switch:

> 'It is perfectly coherent to treat a light switch as a (very cooperative) agent with the capability of transmitting current at will, who invariably transmits current when it believes that we want it transmitted and not otherwise; flicking the switch is simply our way of communicating our desires' [Sho90].

And yet most adults would find such a description absurd — perhaps even infantile. Why is this? The answer seems to be that while the intentional stance description is perfectly consistent with the observed behaviour of a light switch, and is internally consistent, '…it does not *buy us anything*, since we essentially understand the mechanism sufficiently to have a simpler, mechanistic description of its behaviour' [Sho90].

Put crudely, the more we know about a system, the less we need to rely on animistic, intentional explanations of its behaviour. However, with very complex systems, even if a complete, accurate picture of the system's architecture and working *is* available, a mechanistic, *design stance* explanation of its behaviour may not be practicable. Consider a computer. Although we might have a complete technical description of a computer available, it is hardly practicable to appeal to such a description when explaining why a menu appears when we click a mouse on an icon. In such situations, it may be more appropriate to adopt an intentional stance description, if that description is consistent, and simpler than the alternatives. The intentional notions are thus *abstraction tools*, which provide us with a convenient and familiar way of describing, explaining, and predicting the behaviour of complex systems. Being an intentional system seems to be a *necessary* condition for agenthood, but is it a *sufficient* condition? In his Master's thesis, Shardlow trawled through the literature of cognitive science and its component disciplines in an attempt to find a unifying concept that underlies the notion of agenthood. He was forced to the following conclusion:

> 'Perhaps there is something more to an agent than its capacity for beliefs and desires, but whatever that thing is, it admits no unified account within cognitive science' [Sha90].

So, an agent is a system that is most conveniently described by the intentional stance; one whose simplest consistent description requires the intentional stance. Before proceeding, it is worth considering exactly which attitudes are appropriate for representing agents. For the purposes of this survey, the two most important categories are *information attitudes* (like belief, knowledge) and *pro*-attitudes (desire, intention, obligation, commitment, choice).

Thus information attitudes are related to the information that an agent has about the world it occupies, whereas pro-attitudes are those that in some way guide the agent's actions. Precisely which *combination* of attitudes is most appropriate to characterise an agent is, as we shall see later, an issue of some debate. However, it seems reasonable to suggest that an agent must be represented in terms of at least one information attitude, and at least one pro-attitude. Note that pro- and information attitudes are closely linked, as a rational agent will make choices and form intentions, etc., on the basis of the information it has about the world.

2.1.3. Reasoning for Agents: BDI Foundations

The BDI model was conceived by Bratman as a theory of human practical reasoning [Bra87]. Its success is based on its simplicity reducing the explanation framework for complex human behaviour to the *motivational stance* [Den87]. This means that the causes for actions are always related to human desires ignoring other facets of human cognition such as emotions. Another strength of the BDI model is the consistent usage of folk psychological notions that closely correspond to the way people talk about human behaviour.

Beliefs are informational attitudes of an agent, i.e. beliefs represent the information, an agent has about the world it inhabits, and about its own internal state. But beliefs do not just represent entities in a kind of one-to-one mapping; they provide a domain-dependent abstraction of entities by highlighting important properties while omitting irrelevant details. This introduces a personal world view inside the agent: the way in which the agent perceives and thinks about the world.

The motivational attitudes of agents are captured in *desires*. They represent the agent's wishes and drive the course of its actions. Desires need not necessarily be consistent and therefore may not be achieved simultaneously. A "goal deliberation" process has the task to select a subset of consistent desires (often referred to as *goals)*. Actual systems and formal theory mostly ignore this step (with the exception of 3APL [Das03], [Das04]) and assume that an agent only possesses non-conflicting desires. In a goal-oriented design, different goal types such as achieve or maintain goals can be used to explicitly represent the states to be achieved or maintained, and therefore the reasons, why actions are executed [Bra04]. When actions fail it can be checked if the goal is achieved, or if not, if it would be useful to retry the failed action, or try out another set of actions to achieve the goal. Moreover, the goal concept allows to model agents which are not purely reactive i.e., only act after the occurrence of some event. Agents that pursue their own goals exhibit pro-active behaviour.

Plans are the means by which agents achieve their goals and react to occurring events. Thereby a plan is not just a sequence of basic actions, but may also include more abstract elements such as sub-goals. Other plans are executed to achieve the sub-goals of a plan, thereby forming a hierarchy of plans. When an agent decides on pursuing a goal with a certain plan, it commits itself (momentarily) to this kind of goal accomplishment and hence has established a so called *intention* towards the sequence of plan actions. Flexibility in BDI plans is achieved by the combination of two facets. The first aspect concerns the dynamic selection of suitable plans for a certain goal which is performed by a process called "meta-level reasoning". This process decides with respect to the actual situation which plan will get a chance to satisfy the goal. If a plan is not successful, the meta-level reasoning can be done again allowing a recovery from plan failures. The second criteria relates to the definition of plans, which can be specified in a continuum from very abstract plans using only sub-goals to very concrete plans composed of only basic actions.

```
BDI-interpreter
Initialize-state();
repeat
    options := option-generator(event-queue);
    selected-options := deliberate(options);
    update-intentions(selected-options);
    execute();
    get-new-external-events();
    drop-successful-attitudes();
    drop-impossible-attitudes();
end repeat
```

Figure 2-1. BDI Interpreter from [Rao95]

The foundation for most implemented BDI systems is the abstract interpreter proposed by Rao and Georgeff (see algorithm Fig.2-1) [Rao95]. At the beginning of every interpreter cycle a set of applicable plans is determined for the actual goal or event from the event queue. Thereafter, a subset of these candidate plans will be selected for execution (meta-level-reasoning) and will be added to the intention structure. After execution of an atomic action belonging to some intention any new external events are added to the event queue. In the final step successful and impossible goals and intentions are dropped. Even though this abstract interpreter loop served as direct implementation template for early Procedural Reasoning Systems (PRS) [Ing96], nowadays it should be regarded more as an explanation of the basic building blocks of a BDI system. Several important topics such as goal deliberation and the distinction between goals and events are not considered in this approach.

2.2. Background of Intelligent Tutoring Systems

Since the 1960s, researchers have created numerous Computer Assisted Instructional systems [Sle82, Urb04]. The purpose of applying computers in assisting instructions is to help students to learn more efficiently. Traditional education systems instructing via computers are called Computer-Assisted Instruction (CAI) systems. CAI systems present instructional material in a rigid tree structure to guide the students from one content page to another depending on their answer [Leh95], as illustrated in Fig. 2-2. While traditional CAI systems may be somewhat effective in helping learners, they are restrictive in that they do not consider the diversity of students' knowledge states and their particular needs (see [Bru02] and [Yao03]). Such systems do not generate flexible instructional plans. Instead, they follow a pre-specified and fixed plan. Moreover, CAI systems are not adaptive and unable to dynamically provide the same kind of individualized attention that students would receive from human teachers [Ben99].

This drawback has prompted a promising direction in the application of Artificial Intelligence techniques in education known as *Intelligent Tutoring Systems* (ITSs) [Bur88]. Intelligent Tutoring Systems are computer-based programs that present educational materials in a flexible and personalized way that is similar to one-to-one tutoring [Bru99]. In particular, ITSs have the ability to provide learners with tailored instructions and feedback. The basic underlying idea of ITSs is to realize that each student is unique. These systems can be used in the traditional educational setting or in distant learning courses, either operating on stand-alone computers or as applications that deliver knowledge through the internet.

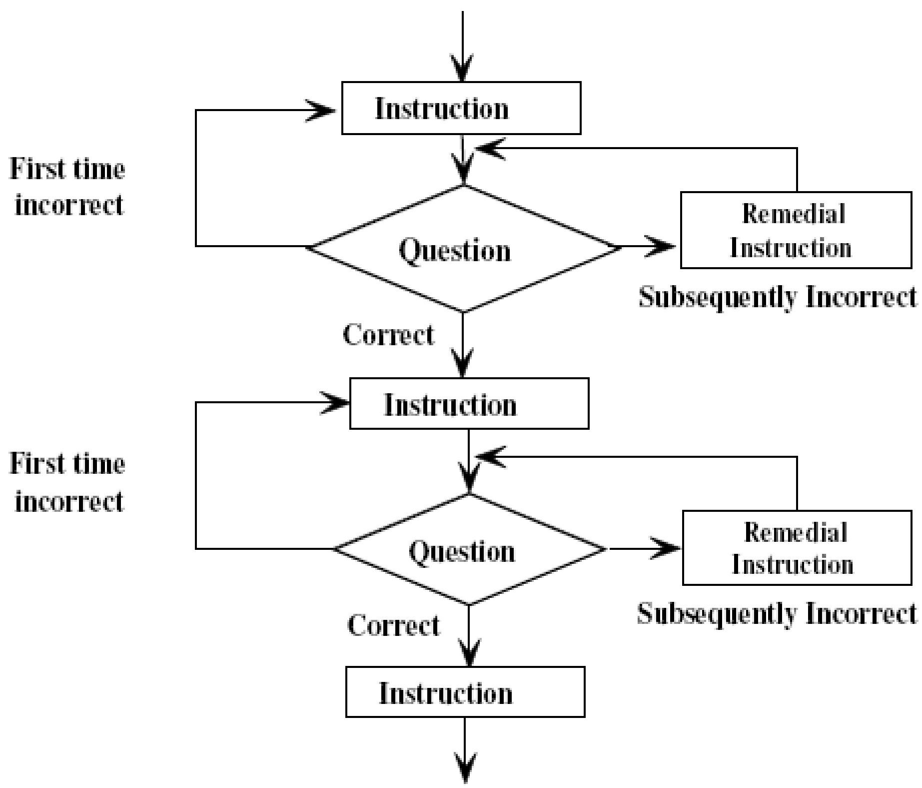

Figure 2-2 Tree Structure of a CAI system

An intelligent tutoring system is one type of expert system [Sow84]. Boose [Boo86] defined an expert system as a knowledge-based reasoning system that captures and replicates the expertise of human experts. Kearsley [Kea87] defined an intelligent tutoring system as application of artificial intelligence techniques to teach students. Sleeman and Brown [Sle82] defined an intelligent tutoring system as a program that uses artificial intelligence techniques for representing knowledge and carrying on an interaction with a student. According to them, an intelligent tutoring system must have its own problem-solving expertise, its own diagnostic or student modeling capabilities, and its own explanatory capabilities. It must know when to interpret a student's problem-solving activity, what to say, and how best to say it. Hence, an intelligent tutoring system closely resembles the process when a student and teacher interact in a one-to-one situation [Ten87].

In his work [Mur99], Murray identifies two major trends of ITSs' research according to the system's objective:

Problem Solving Support: For many years, problem solving support was considered as the primary duty of an ITS [Bru99]. The purpose for problem solving support is usually to provide the student with intelligent help for each step when resolving a task, such as a project or a problem. When the student is stuck on one step, the system provides a hint showing the next correct solution step for the student, or offering appropriate error feedback. In this setting, the critical problem for the system is to interpret the student's actions and infer the solution plan that the student is currently following based on a partial sequence of observable actions. That is, the system needs to understand the student's plan, and apply this understanding to provide help. Examples of this type of ITS are [And89, Ger98, Joh01, Syk03].

Curriculum Sequencing: Curriculum sequencing is now the most popular and important technology in Web-based ITSs [Bru99]. The objective of curriculum sequencing technology (also referred to as instructional planning technology) [Bru99] is to provide the student with a personalized optimal path through the learning material. The examples are [Bar76, Bru97]. Recommending appropriate learning sequencing is necessary in Web-based education. Web-based learning students usually work alone without a teacher's instructional assistance and they study the subject at their own pace. As a result, appropriate learning sequencing recommendations are essential in order to enable each student to learn the subject in the most beneficial and individualized way [Bru99].

ITSs have been shown to be highly effective in increasing students' performance and motivation levels compared with traditional instructional methods [Koe97]. One of the key elements that distinguishes ITSs from more traditional CAI systems is ITSs' capability to dynamically maintain a model of a student's reasoning and learning that keeps track of a student's knowledge during the study [Shu96]. As noted by Shute and Psotka [Shu96], ITSs must be able to achieve three main tasks:

(i) accurately diagnose a student's knowledge level using principles rather than preprogrammed responses;

(ii) decide what to do next and adapt instruction accordingly;

(iii) provide feedback.

This kind of diagnosis and adaptation, which is usually accomplished using Artificial Intelligence techniques, is what distinguishes ITSs from CAIs. Bloom [Blo84] demonstrates that individual one-on-one tutoring is the most effective mode of teaching and learning. Carefully designed and individualized tutoring produces the best learning for the majority of people. ITSs uniquely offer a technology that implements computer-assisted one-on-one tutoring.

2.2.1. The Key Components of ITS

Early CAI systems were not modular [Woo91]. This unfavourable structure caused problems when a system required modification, and it was sometimes necessary to restructure the whole system. There was, then, a need to divide the system into separate components: the knowledge to be taught, the instructional method, the user interface and the student modelling.

Researchers typically separate an ITS into several different parts, and each part plays an individual function. Usually, most ITSs have four common major components [Sle82], as illustrated in Figure 2-3:

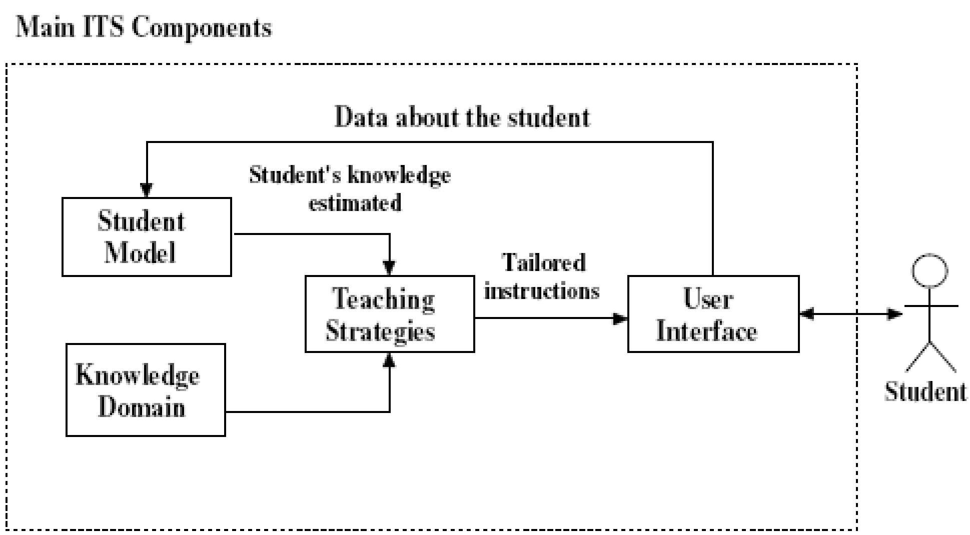

Figure 2-3. Major ITS Components

1. *Knowledge Domain***:** The knowledge domain stores learning materials that the students are required to study for the topic or curriculum being taught.

2. *Student Model:* The student model stores information that is specific to each individual learner and enables the system to identify different users. Usually, this information reflects the system's understanding of one learner's current knowledge state. Thus, the student model can track a student's understanding and particular need. Without an explicit student model, the teaching strategies component is unable to make decisions to adapt instructional content and guidance (see Figure 2-2) and is forced to treat all students similarly. Student modelling is sometimes thought of as a sub-problem of the user modelling problem [Jam95], whereby the target application is an ITS. Student modelling presents well-known difficulties stemming from the fact that modelling the student within an intelligent tutoring system involves a good deal of inherent

uncertainty [Con02a]. It is hard to establish unequivocally what a student knows and what she is learning [Con02a], [Jam95]. Thus, one of the biggest challenges in designing ITSs is the effective assessment and representation of the student's knowledge state and specific needs in the problem domain based on uncertainty information.

3. *Teaching Strategies:* The teaching-strategies component refers to instructional techniques for teaching. For example, the component decides when to present a new topic, how to provide recommendations and guidance, and which topic to present. As mentioned earlier, the assessment result of the student model is input to this component, so the system's pedagogical decisions reflect differing needs of students. Thus, this component needs to take appropriate actions to manage one-on-one tutoring, such as switching teaching strategies and using a variety of teaching approaches at the appropriate times according to the student's particular needs and problems.

4. *User Interface Component:* The user interface component decides how the system interacts with a user. The dialogue and the screen layouts are controlled by this component. A well designed interface can enhance the capabilities of an ITS by allowing the system to present instructions and feedback to the student in a clear and direct way.

2.2.2. Review of ITSs Technology Research

According to a recent panel [Cor99], the current generation of intelligent tutoring systems is only half as effective as human tutors, and we need to develop tutoring systems that are as effective as human tutors. We need to study human tutors when they provide one-to-one instruction in distributed-learning systems and use the expertise to build intelligent tutoring systems. Existing distributed-learning systems are designed to instruct students based on information already stored in memory. The systems do not adapt to the needs of the learner by diagnosing, in minute detail, the sources of errors and by providing specific instruction to overcome the errors. Distributed-learning systems need to form a model of the learner and provide instruction similar to a tutor in a one-to-one interaction mode. Bloom [Blo84] described the *two-sigma problem,* which suggests that learners who are given one-to-one instruction performed two standard deviations higher when compared to learners who received face-to-face group instruction; however, providing one-to-one attention using a human tutor could be expensive and time consuming in distributed-learning systems. As a result, distributed-learning systems need to develop and use intelligent tutoring systems so that the human tutor expertise is built into the computer system to provide the one-to-one tutoring to students. This is critical, because in distributed-learning systems, students could be in any location and may not have access to human tutors for one-to-one instruction.

An effective intelligent tutoring system should simulate what good human tutors do when tutoring in a one-to-one situation. Bloom [Blo84] mentioned that educators should try to replicate the same strategies used by students and teachers in a one-to-one environment to other teaching situations, because the one-to-one tutoring environment is ideal for learning. Woolf [Woo96] described a Cardiac and an Engineering intelligent tutor that include strategies to help achieve the *two-sigma* effect described by Bloom. Anderson et al. [And85] conducted a study where two groups of students were given the same lectures, but one group used an intelligent tutoring system tutor for the exercises. They found that the tutored students spent 30% less time on the problems than those working on their own. The tutored group also scored 43% better on the post- test. Anderson et al. [And85] noted that the presence of the tutor is more significant for the performance of low achievers. The goal of an intelligent tutoring system is to replicate the one-to-one interaction between a tutor and a learner. This should include all of the expertise (interface with the learner, content, a model of the student, and pedagogical) that is involved in the tutoring process.

Dede [Ded86] mentioned that an intelligent tutor is a stand-alone device, which can initiate interactions with its user and incorporates all the knowledge needed to teach a subject. However, to build a good intelligent tutoring system for distributed-learning systems, the expertise has to be elicited from experts in the domain. Acquiring sufficient and correct teaching expertise is a major problem for builders of intelligent tutoring systems [Woo87]. Most expert systems projects claimed that the knowledge elicitation process is the most complex and time consuming in the development of expert systems [Ber87], [Ols87]. Some reasons given are as follows:

1. Experts in the field are not able to articulate and make explicit their expertise.

2. Expertise from experts tend to be of a high level, and this cannot be used to tutor the learner.

3. Experts obtain their expertise through an implicit learning process that cannot be made explicit [Ber87].

The more experienced one becomes in a knowledge domain, the more difficult it is to make the knowledge explicit [Ber87]. Experts possess compiled knowledge, which exists in large chunks accumulated over the years, and this knowledge is difficult to elicit. Also, the knowledge elicitation process may influence the quality and quantity of expertise extracted.

Intelligent tutoring systems require an interdisciplinary team to design and develop. In addition to requiring domain and coding knowledge, it requires educators and psychologists to specify the instructional strategies and pedagogical model to incorporate into the system. Because conventional educational research has focused on group instruction, little is known about the same individual learning characteristics vital in developing the student model and pedagogical modules for intelligent tutoring systems [Ded86]. One such learner characteristic is learning style. Ally and Fahy [All02] found that students with different learning styles prefer different pedagogical support when learning in a one-to-one distance education environment. Pedagogical expertise of a tutor in a one-to-one situation is the least understood and does not exist in an explicit form to be included in an intelligent tutoring system [Ohl87].

Some of the intelligent tutoring systems have been developed to explore the capabilities of artificial intelligence techniques in the instructional process rather than to build an effective instructional system [Par87]. The next generation of intelligent tutoring systems should be concerned with instructional and pedagogical issues rather than computer science or artificial intelligence issues, such as specific programming techniques, software architecture, etc. [Par87]. The goal of an intelligent tutoring system is to replicate the one-to-one interaction between a tutor and a learner. This should include all of the expertise (interface, content, a model of the student, and pedagogical) that is involved in the tutoring process.

An intelligent tutoring system does not act on the basis of pre-entered questions, anticipated answers, pre-specified branches, and the knowledge accumulated when a student learns [Ten87]. An intelligent tutoring system should have domain expertise, it should build a model of the learner, and tutor the learner based on the learner's model. It should behave as a tutor does in a one-to-one situation. According to [Woo87a], an intelligent tutoring system should reason about a student's knowledge, monitor a student's solutions, and adapt the teaching strategy to the student's individual learning pattern. The intelligent tutoring system should be able to conduct its own learner analysis and continually improve as it interacts with learners to become a more effective tutoring system. After many learning cycles, the intelligent tutoring systems should be comprehensive enough to meet the needs of learners with different learning styles and preferences. For example, an intelligent tutoring system could monitor strategies that different learning styles use successfully and build a best practice database of effective learning strategies for different learning styles. The intelligent tutoring systems could also track common errors made by students and prescribe strategies to students to prevent them from making these errors.

2.3. Why Software Agents as Tutors?

In distributed learning, students can be in any location to take courses, as long as they have access to communication technology with which to access the course. Distributed learning could be either synchronous or asynchronous. In synchronous learning, the learning is in real time, where the learner interacts and receives information as needed. In asynchronous learning, there is a delay in the interaction between the system and the student. The information in this thesis is related to distributed synchronous learning, where students interact with the intelligent tutoring system in real time and receive feedback as they interact with the system.

The main objective of distributed learning systems is to enable individually subscribed to learning services to be delivered to their associated users whenever they request them, and wherever the users are, in a customized form that matches their profile. Thus, intelligent mobile agents have been introduced to provide this kind of dynamic service provisioning and management. The agents' technology has several advantages for implementing new services on distributed systems. In fact, this technology enables these systems to distribute the functionalities in small, reproducible, and distributed software entities. It also allows for a clear and easy separation between their internal, private knowledge, and their interface toward the external world and other agents.

The distributed learning services provisioning and management fits well for exploiting the agents' properties.

- *Agents' autonomy:*
    - Allows for making decisions on service access, the interfaces' configuration, and service provisioning without human assistance.
    - Allows for automating the control and management tasks and, hence, reducing the operator's workload.
    - Allows for automating the service deployment and provision, thus reducing the effort and time required for the installation and the maintenance of services.
- *Agents' intelligence:*
    - Allows for the dynamic customization and configuration of services. The agents can learn and adapt to the preferences of their users and detect and update old versions of services.
    - Allows for the service intelligence to be downloaded dynamically from the providers and for collaboration between different providers.
- *Agents' mobility:*
    - Supports the dynamic topological of service provisioning.
    - Enables e-learning services to be provided instantly and to be customized directly at the locations where services are needed.
    - Enables dynamic provision of customized services by downloading service agents from the service-provider system to the network nodes or user terminals.
- *Agents' sociability:*
    - Offers the potential to distribute service-related processing, and also offers a mechanism for the nodes in different networks to cooperate in order to provide a service to the user.
    - Allows for negotiation for service features.
    - Provides multi-services interaction and coordination.
    - Allows for the asynchronous and cooperative processing of tasks.
    - The agents' technology fits well for e-learning, because it supports the following requirements:
- *Dynamic scalability:*

  - Multi Agent Systems (MASs) support huge distributed systems such as the Internet. In fact, each service is modeled as a collection of agents, each agent occupying different places at different times, because it can move from one place to another.
  - MASs support on-demand provisioning of services. In fact, when servers are implemented with MASs, the agents' mobility allows them to deploy new replicas when the demand arises or to migrate to the location where the demand is high.
  - MASs enable the provision of flexible solutions, in which services are partitioned into mobile service agents achieving multiple functions that can be spread across the network.
- *Distribution of services:*
  - MASs fit well for modeling the ideal situation for a mobile user, using mobile agents that can provide the ubiquitous availability of applications, data files, and user profiles by using the concepts of mobility and cloning.
  - MASs enable control tasks to be performed in a distributed manner by mobile intelligent agents using the concept of remote programming [Fug98] instead of the client/server programming concept used currently in most Web-based learning systems.
  - The possibility of bringing control or management agents as close as possible to the resources allows for a more decentralized realization of service control and management software than could be achieved otherwise.
- *Reduction of traffic:*
  - MASs decrease pressure on centralized network-management systems and network bandwidth by using spatial distribution and temporal distribution.
  - MASs' autonomous and asynchronous operation reduces the requirements regarding traffic load and the availability of the underlying networks
- *Independence regarding failures:*
  - MASs reduce the influence of signaling network faults during service processing, because once a service agent has migrated, the processing will be performed locally.
  - The agents' migration to the required data reduces dependence regarding network availability, so more robustness is achieved in the distributed system.

There have been a number of efforts to introduce agents into learning environments in order to create better and more human-like support for exploratory learning, and social events that support tutor-tutee interactions and collaborative learning [Cla01], [Chi01]. The fundamental reason for introducing agents as tutoring knowledge elements is their capabilities of communication and interaction. These characteristics are fundamental for agent's usage in an educational environment. Tutoring agents are entities whose ultimate purpose is to communicate with the student in order to efficiently fulfill their respective tutoring function, as part of the pedagogical mission of the system [Vic98]. Several of these tutoring functions are:

- to present a topic
- to explain the topic
- to give an example
- to answer a student's question
- to ask a student a question and evaluate the student's answer
- to examine and diagnose a student's behaviour during the learning process

Therefore, the agent should also have the roles of a human tutor within a group of students, which are:

- *Interrogator* – poses questions and the students of a collaborative group then provide answers. The questions should provide help for the students to reach a common learning goal.
- *Reviewer* – analyzes the students' answers, including whether it is correct or not.
- *Monitor* – records the answers from all the students and the communications among students during the collaborative learning process.
- *Instructor* – gives individualized instructions and helps those students who cannot keep up with the progress of their group-mates.

All together: the tutor-agent should be able to present and explain a learning subject, to pose questions about it, to evaluate the learner answers and also to provide specific feedback. The tutor-agent should generate relevant replies from a knowledge base in response to the questions posed by the learner. If the tutor-agent cannot generate an adequate response to one question then it should be able to communicate with other tutors (human or agents) in order to accomplish its task.

Finally, agent technology offers a number of very interesting advantages, but it should not be seen as the only solution for all tutor based learning environments. Rather, it should be seen as a technology that can resolve some problems. Furthermore, we have to consider some of MASs' disadvantages:

- Require a specific run-time environment (agent execution environment) to be present in all nodes to be visited.
- Create a security problem. The platforms have to be protected from malicious agents and vice versa.
- May increase the network load in some situations. One of the mobile agents'

goals is to reduce the network traffic, but it does not seem useful for every agent to migrate in every situation; doing so would probably increase the network traffic. Therefore, new strategies have to be developed to establish under which circumstances an agent will migrate.

- Do not provide location transparency. Each agent must be aware of the location to be visited.

2.4. Agents based Intelligent Tutoring Systems: A Review

Advances in computer technology have lead to the development of sophisticated computer based Intelligent Tutoring Systems ITS for the distance educational environments [Wen87]. The ITS paradigm is generally Expert Systems based and it selects problems for users to solve and provides them feedback on their solutions.

Another paradigm for ITS is agent based tutoring systems. The intelligent agent in an intelligent tutoring system performs on behalf of the tutor to help learners achieve learning outcomes and to prescribe teaching strategies based on learners' profiles in the student model and content in the domain module. As the agent interacts with the learner, it gains more experience by learning about the learner. The expertise in the intelligent tutoring system intelligent agent should allow the agent to help learners achieve the learning outcome without human intervention. The intelligent agent should anticipate learners' responses and respond immediately to take corrective action or to present the next learning intervention based on learners' characteristics and styles to maximize learning benefits. In other words, the intelligent agent should form dynamic profiles of the learner and work ahead of the learner by guiding the learner in what to do next in the learning process. The intelligent agent system should behave like an expert tutor by interacting with the different components in the intelligent tutoring system to assemble the expertise required to help learners achieve the learning outcome.

The application of agents in the educational sector comes about mainly in the form of personal assistants, user guides, alternative help systems, dynamic distributed system architectures, human-system mediators and others. As a result of all of the changes that have taken place in the educational system, one now sees the increasing emergence of complex and dynamic educational infrastructure that needs to be efficiently managed. Corroborating this, new (types of) educational mechanisms and services need to be developed and supplied.

In particular these services need to satisfy a series of requirements such as personalization, adaptation, support for user mobility, support for users while they are dealing with new technologies, among others. Agents emerge to provide solutions for these requirements in a way that is more efficient when compared to other existing technologies [Aro99].

According to Aroyo and Kommers [Aro99], agents can influence different aspects in educational systems. They supply new educational paradigms, support theories and can be very helpful both for learners and for teachers in the task of computer-aided learning.

Lees and Ye [Lee01] believe that the application of the agent paradigm to CSCW potentially can exchange information more fluid among the participants of groupware systems (as decision-making systems), help in control of the process flows and also supply groupware interfaces. These ideas also are applicable to other domains, such as is the case of interactive learning.

According to Kay [Kay01], in the first computer-assisted teaching environments the idea was to build "teachers" who could transmit knowledge to the learners. Currently, these types of environments are more geared up for exploration on the part of the learners, designing, building and using adaptive systems as tools. These environments also are being built to give greater responsibility to the learners regarding aspects of the learning process, and especially regarding control of its model, which is the central aspect in the adaptability of the tools.

For McCalla & all [McC00], learner models may have a variety of purposes depending upon the type of knowledge that needs to be stored and processed. For them, the computation of all of the learner (sub-)models of an environment can be computationally expensive and not always necessary. In the work cited four purposes are presented for a model: reflection, validation, matchmakers and negotiation.

Guizzardi et al. [Gui02] investigate the nonhierarchical relationship between teachers and students in an environment where everyone can teach and learn. They gather two perspectives: one from an implementation point of view and the other one from a software engineering perspective and propose an agent-based system to support extra-class discussions between students and teachers.

For Kay [Kay01], there are several problems from the learners’ point of view. One is the increase in the power of choice and control over the model. This could increase the learners’ workloads or even turn into a distraction. In this case, the learners should take advantage of the moments such as the end of a course or a topic to evaluate and reflect upon their participation and the learning process. Another potential problem is incorrect data being supplied by the learners. The solution adopted in this work for that problem was to store the type of information learners are providing and the type the environment extracts.

Mustapha [Mus04] considers the roles of an agent in an educational environment to be the following: to monitor, control and catalyze the social knowledge building among the community of learning. Social knowledge is considered to be derivable from socializing oneself with the peers, communities through formal or informal discussion, chitchat or social gatherings.

For Jennings & all [Jen98], autonomous agents and Multi-Agent Systems represent a new modality of analyzing, designing and implementing complex software. The agent concept has a wide area of applications, ranging from the creation of personal assistants to air traffic control systems, electronic commerce and the group work support.

Prendinger and Ishizuka [Pre01] present the use of animated conversational agents in a pedagogical environment where Japanese native speakers practice English. Their approach identifies social role awareness as an important concept for the agents. User-agent interactions are materialized as role-playing interactions.

In the literature, there are a few authors who have written on the use of agents for distance learning. For example, Santos and Rodriguez [San02] discussed an agent architecture that provides knowledge-based facilities for distance education. Their approach is to take advantage of recent standardization activities to integrate information from different sources (in standardized formats) in order to improve the learning process, both detecting learner problems and recommending new contents that can be more suitable for the learner's skills and abilities. They accomplish this by using a suite of different agents, such as a "learning content agent," a "catalog agent," a "competency agent," a "certification agent," a "profile agent," and a "learner agent."

Rosié et al. [Ros02] looked at the application of the Semantic Web together with personal agents in distance education. They saw the following possibilities of such a combination: (a) enable sharing of knowledge bases regardless of how the information is presented, (b) allow access to services of other information systems that are offered through the Semantic Web, and (c) allow reuse of already stored data without the need to learn the relations and terminology of the knowledge base creator.

Koyama et al. [Koy01] proposed the use of a multifunctional agent for distance learning that would collect the learner's learning material requirements, perform management, do information analysis, determine the learner's understanding of a particular domain, handle the teaching material, and communicate with the learners. The distance-learning system would be built on the WWW, and this agent would reside in a Web server. Koyama et al. also proposed a fairly elaborate "judgment algorithm" that monitors the learner's progress and learning time and does learner testing in conjunction with learner requirements, learner personal history, and the existence of "re-learning items" in order to decide appropriate learning materials for the learner.

Finally, Cristea and Okamoto [Cri00] described an agent-managed adaptive distance-learning system for teaching English that adapts over time to a learner's needs and preferences in order to improve future learning performance. They use two agents, a Global agent (GA) and a personal agent (PA), to manage two student models, a global student model (GS) and an individual student model (IS), respectively. The GS contains global student information, such as the common mistakes, favorite pages, favorite lessons, search patterns, and so on. The IS contains personal student information, such as the last page accessed, grades for all tests taken, mistakes and their frequency, the order of access of texts inside each lesson, and so on. The PA manages the user information and extracts from it useful material for user guidance. The PA also requests information from the GA and collaborates with other PAs to obtain more specific information (e.g., what material other learners have used in a similar situation) than is available from the

GA. In short, the PA acts as a personal assistant to the learner to provide guidance as to what material the learner should be studying. The GA averages information from several users to fill in the general student model. Its role is to give the PAs condensed information that might show trends and patterns. The GA cannot contact the learner directly unless the PA requests it.

Chapter 3: Modeling Software Agent: A New Approach

There are three rules for writing a novel. Unfortunately, no one knows what they are.
Somerset Maugham (1874-1965)

Development methodologies for multiagent systems have received a lot of attention recently [Wei01]. These methodologies differ from each other in many respects. They differ on the software development phases they capture (analysis, design, implementation). Some of them focus on inter-agent aspects, while others also provide support for the internal workings of an agent. Finally, some methodologies explicitly deal with the environment, while others do not. We are interested in developing multiagent systems for applications that are best understood in terms of social and cognitive concepts like norms, roles, beliefs and goals. Such applications usually include resources and services that are part of the multiagent environment. Therefore a methodology should account for the environment too. Moreover, the methodologies should provide, besides guidelines for the analysis and design phase, also guidelines for implementation phase, and explain how the design concepts can be mapped onto instructions of an available programming language. The aim of this chapter is to highlight a method for designing agents to support collaborative learning applications.

3.1. Agent-oriented Methodologies: State of the Art

Much work in agent theory is concerned with sorting out exactly what the relationship between the different attitudes is. This work will follow this trend. However, we cannot proceed further without investigating methods for representing and reasoning about these intentional notions.

Methodologies for multiagent system development should assist the developer in making decisions about those aspects of the analysis, design and implementation, that are crucial for multiagent systems, namely, social and cognitive concepts (e.g. norms and goals). In this section, we review several existing agent-oriented methodologies.

3.1.1. GAIA

Gaia [Woo00] comprises an analysis and design phase and explicitly refrains from including an implementation phase. Analysis is driven by a set of requirements and aims at understanding the system and its structure. It provides two models: a role model and an interaction model. The role model specifies the key roles in the system and characterizes them in terms of permissions (the right to exploit a resource) and responsibilities (functionalities). The interaction model captures the dependencies and relations between roles by means of protocol definitions. Gaia is only concerned with the society level; it does not capture the internal aspects of agent design. The design phase provides three models: the agent model, the service model, and the acquaintance model. The agent model identifies so called agent types, which are sets of roles. The service model identifies the services (or functions) associated with a role. Finally, the acquaintance model identifies the communication links between agent types. This model can be used to detect potential communication bottlenecks. The method has been extended with a model of organizational rules and organizational structure [Zam00]. This allows the developer to specify global rules that the organization should respect or enforce.

Like norms, such rules are typically formulated at a high conceptual level. Little is said about ways of implementing them. The interaction of agents with the environment is not treated separately.

Discussion Gaia does not support the implementation phase. However, the type of design choices, concepts and their relations are at least partly driven by the type of implementation language one has in mind. In Gaia it is clear that implementing the system in a procedural language that could be easily described using MetateM would fit best. However, Gaia provides very little support for building BDI agents that reason about their different responsibilities, plans and beliefs. Although aspects like *permission* and *responsibility* have a formal description, they do not have a formal semantics. Therefore it is difficult to check whether agents really implement a certain role. Especially when different roles containing several responsibilities are joined into an agent type. Although permissions seem to be norms, it is unclear how they are

3.1.2. AAII

The AAII methodology proposed by David Kinny [Kin96] makes no distinction between the analysis and design phase. The methodology generates a set of models, based on existing object-oriented models. From an external viewpoint (inter-agent), the system is decomposed into agents, which are modeled as complex objects characterized by their purpose, their responsibilities, the services they perform, the information they require and maintain, and their interaction.

This leads to two models: an agent model and an interaction model. From an internal viewpoint (intra-agent) the elements required by a computational BDI architecture are modeled for each agent. This leads to a belief model, a goal model and a plan model. The development of an intra-agent model starts from the services provided by an agent and the associated events and interactions. This leads to an identification of plans. The belief model is constructed by analyzing the conditions that control the execution of actions and decompose them into basic components: beliefs. The environment is not modeled separately.

Discussion David Kinny's AAII is one of the few approaches that takes the intra agent perspective seriously. Roles can be considered as responsibilities, which can in turn be considered as sets of services. Services are activities that are not natural to decompose any further. However, because of this nice hierarchical decomposition, the potential power of the BDI concepts is not realized. It leads to goal models that are simple AND/OR graphs. Hardly any reasoning is required by the agents. The methodology is very practice-oriented which leads to graphical models, but without much semantics of the concepts. It is left to the programmer to fill in the gaps. Like Gaia, AAII does not support open agent systems. The organization of the system is almost completely hierarchical in a truly object-oriented manner. No norms or rules are specified as such.

3.1.3. SODA

The SODA methodology [Omi00] has a clear distinction between analysis and design. The methodology is only concerned with the inter-agent viewpoint. The analysis phase provides three models: the role model, the resource model, and the interaction model. The role model defines global application goals in terms of the tasks to be achieved. Tasks can be individual or social. Individual tasks are assigned to roles while social tasks are assigned to groups. A group is an abstract concept that can be analysed as a set of roles. The resource model captures the application environment and identifies the services that are available. The resource model defines abstract access modes (permission), modeling the different ways in which the services associated with a resource can be exploited by agents. The interaction model defines the interaction between roles, groups and resources in terms of protocols.

The design phase refines the abstract models from the analysis phase and provides three models: the agent model, the society model and the environment model. The agent model specifies the mapping from roles onto agent classes. An agent class is characterized by the tasks, permissions and interaction rules associated to a role. It also specifies the cardinality (the number of agents in that class), their location (fixed for static agents and variable for mobile agents) and their origin (inside or outside the system). The society model specifies a mapping from groups onto societies of agents. An agent society is characterized by the social tasks, the set of permissions, the participating social roles, and the interaction protocols. Finally, the environment model specifies a mapping from resources onto infrastructure classes. Infrastructure classes are characterized by the services, the access modes for roles and groups, and the protocols for interacting with the environment.

Discussion SODA is a very usable development methodology. The inter-agent aspect is well developed. The interaction among agents, but also the interaction between agents and the environment is taken seriously. However, SODA does not specify the design of the agents themselves. Therefore it too leaves a gap between the design and implementation of the multi-agent system.

Due to the fact that SODA recognizes explicit organizational structures and rules, it becomes applicable for open agent systems. However, in SODA, agents are conceived of as pieces of software designed for one purpose only: to fulfill their roles. The assignment of agents to roles is done at design time, and remains stable. We have a much more dynamic picture of role assignment. Agents are conceived of as entities that are given. Agents may enter a society, and start acting through some API. This means that the roles or tasks of an agent may start to conflict, and that the agent must have the means to resolve such conflicts.

Although many concepts are used for the inter-agent specification, they are not formalised. Therefore it becomes difficult to check whether agents fulfilling a role comply to all the organizational rules. Another worry is that the use of procedural specifications of behavior, like standardized tasks, will bias the design. It suggests traditional imperative programming constructs. Such a simple choice is nice, when it is enough. However, such a view may limit the potential benefits of multi-agent systems, such as flexibility and robustness, because it does not take advantage of the autonomy and possible intelligence of the agents. A declarative specification in terms of goals, objectives or landmarks, would leave the manner in which it is to be achieved up to the individual agent.

3.1.4. Tropos

The Tropos methodology [Cas02] distinguishes between an early and a late requirements phase, and between architectural design and detailed design. It considers both inter-agent and intra-agent issues. The early requirements phase, which is based on the *i*¤ organizational modeling framework [Yu97], is concerned with understanding an application by studying its organizational setting. This phase generates two models: a strategic dependency model and a strategic rationale model. These models specify the relevant actors, their respective goals and their inter-dependencies. In particular, the strategic dependency model describes an ‘agreement’ between two actors: the depender and the dependee. The strategic rationale model determines through a means-ends analysis how an actor’s goals (including soft goals) can actually be fulfilled through the contributions of other actors. The late requirements phase results in a list of functional and non-functional requirements for the system.

The architectural design defines the structure of a system in terms of subsystems that are interconnected through data, control and other dependencies. The detailed design defines the behavior of each component. Agent communication languages like FIPA-ACL or KQML, message transportation mechanisms, and other concepts and tools are used to specify these components. The implementation phase maps the models from the detailed design phase into software by means of Jack Intelligent Agents [How01]. Jack extends Java with five language constructs: agents, capabilities, database relations, events, and plans. It is claimed that these constructs implement cognitive notions such as beliefs, desires, and intentions.

Discussion Of all the methodologies considered, Tropos comes closest to a complete development methodology for multiagent systems. It treats most development phases, and it treats both inter-agent and intra-agent perspectives. The use of means-end analysis to determine the goals of an agent and to determine for what goals an agent depends on other agents, leaves enough room for the agents to fill in the plans for the goals, but is specific enough to give some handles on its implementation.

One can easily note several drawbacks and omissions in Tropos too. First of all the models of Tropos do not have a formal semantics and therefore it is hard to specify an implementation for the design models. It also neglects the environment, and fails to notice that roles affect the access modes or permissions for executing certain actions, or for accessing resources. Also Tropos is meant to design closed systems, in which the designer has control over the agents that enter. However, if a system would allow external agents to enter and interact, an interface between such external agents and the environment and the other agents is required. Such an interface can be thought of as a specific governor agent, as in Islander [Est02], a social contract, as in OperA [Dig03], or an agent coordination context (ACC) [Omi02] as developed in the research on coordination languages.

A final drawback is that processes internal to an agent are specified by plan graphs. It shows a bias towards a very procedural implementation of the BDI agents, which are realized by the JACK system. Although JACK agents do have representations for beliefs (database), intentions (plans) and desires (triggers) these implementations are very simple versions of the concepts used in the specification. JACK agents lack the ability to reason with their beliefs and/or about their desires and intentions. Instead, JACK agents have essentially an event-based architecture.

3.1.5. OperA

The OperA approach does not distinguish between analysis and design models, which can be considered a major drawback. However, OperA contains three models.

The social model describes roles and their dependencies. Roles have objectives: the goals the organization expects an agent to fulfill when enacting that role. OperA allows for agents to have their own goals which should be combined with those of a role when enacting that role [Dig03]. Therefore OperA caters for open agent systems. A role can be dependent on another role to fulfill (part of) its objective. What is crucial for OperA, is that the form of these dependencies is determined by the organization type. In a hierarchical organization dependencies are often translated to delegation relations, whereas in markets they translate to interactions such as the Contract Net.

The interaction model describes the process flow of the system, in terms of scenes and transitions between scenes. This is similar to the Islander approach [Est02]. The scope of a role is limited to a scene. Each scene contains an abstract and declarative specification of the landmarks to be achieved during interaction. Scenes do not (have to) specify complete protocols; they specify landmarks that can be reached in many different ways. Transitions between scenes are subject to constraints, and to a temporal ordering.

The normative model contains all the different types of norms that regulate behavior in the system. The following types of norms can be distinguished (1) Norms for roles; (2) Norms for scenes; (3) Norms on scene transitions. Norms cannot be translated into a design model directly. They will be distributed over the various models of the design phase. Although normative concepts are found in most of the methodologies discussed so far, they are usually immediately associated with roles. They are not formulated in a general way, or associated with activities or scenes. Therefore, norms for roles already bias the design of a system. By contrast, OperA allows one to first formulate norms, and then discuss the various ways of translating them in a society.

The final model of the analysis phase, which is not included in OperA, is the *environment model*. In this model there are specified resources that are available for the agents, e.g. databases, like also the available services.

3.1.6. MaSE

Wood and DeLoach [DeL99], [DeL00] suggest the Multiagent Systems Engineering Methodology (MaSE). MaSE is similar to Gaia with respect to generality and the application domain supported, but in addition MaSE goes further regarding support for automatic code creation through the MaSE tool. The motivation behind MaSE is the current lack of proven methodology and industrial-strength toolkits for creating agent-based systems. The goal of MaSE is to lead the designer from the initial system specification to the implemented agent system. Domain restrictions of MaSE are similar to those of Gaia's, but in addition it requires that agent-interactions are one-to-one and not multicast.

The MaSE methodology is divided into seven sections (phases) in a logical pipeline. *Capturing goals*, the first phase, transforms the initial system specification into a structured hierarchy of system goals. This is done by first identifying goals based on the initial system specification's requirements, and then ordering the goals according to importance in a structured and topically ordered hierarchy. *Applying Use Cases*, the second phase, creates use cases and sequence diagrams based on the initial system specification. *Use Cases* presents the logical interaction paths between various roles in and the system itself. Sequence diagrams are used to determine the minimum number of messages that have to be passed between roles in the system. The third phase is *refining roles*, it creates roles that are responsible for the goals defined in phase one. In general each goal is represented by one role, but a set of related goals may map to one role. Together with the roles a set of tasks are created, the tasks defines how to solve goals related to the role. Tasks are defined as state diagrams. The fourth phase, *creating agent classes*, maps roles to agent classes in an agent class diagram. This diagram resembles object class diagrams, but the semantic of relationships is high-level conversation as opposed to the object class diagrams' inheritance of structure. The fifth phase, *constructing conversations*, defines a coordination protocol in the form of state diagrams that define the conversation state for interacting agents. In the sixth phase, *assembling agent classes*, the internal functionality of agent classes are created. Selected functionality is based on five different types of agent architectures: Belief-Desire-Intention (BDI), reactive, planning, knowledge based and user-defined architecture. The final phase, *system design*, create actual agent instances based on the agent classes, the final result is presented in a deployment diagram. Vision of the future for MaSE is to provide completely automatic code generation based on the deployment diagram.

3.2. Issues in Modeling Agent Systems

We are interested in developing multiagent systems for distance learning applications that are best understood in terms of social and cognitive concepts like emotions, personality, learning styles, organizational structures – social position of participants, roles – beliefs and goals. Such applications usually include resources and services that are part of the multiagent environment. Therefore a methodology should account for the environment too. Moreover, the methodologies should provide, besides guidelines for the analysis and design phase, also guidelines for implementation phase, and explain how the design concepts can be mapped onto instructions of an available programming language.

This choice has important consequences for the internal design of an agent. Therefore we believe that a methodology should not only consider inter-agent, but also intra-agent aspects. These methodologies, except AAII, were selected because they are not based on existing object – oriented development methods, or methods for knowledge-based systems. AAII was selected because it does provide a model for intra-agent aspects.

Given our interest in the development of multiagent systems using social and cognitive concepts, and our concern for implementation, we identified a number of issues that are problematic for these methodologies:

1. There is no agreement on how to identify and characterize *roles* in the analysis phase and *agent types* in the design phase.

2. The concepts used in the methodologies, like responsibility, permission, goals and tasks do not have a *formal semantics* or *explicit formal properties.* This becomes an important issue when these concepts are implemented; implementation constructs do have exact semantics.

3. There is a *gap* between the *design models* of the methodologies and the existing *implementation languages.* It is unreasonable to expect a programmer to implement the proposed complex design models. To bridge the gap, a methodology should either introduce refined design models that can be directly implemented in an available programming language, or use a dedicated agent-oriented programming language which provides constructs to implement the high-level design concepts.

4. The methodologies that include an implementation phase, such as Tropos, propose an implementation language in which it is not explained how to implement reasoning about *beliefs,* reasoning about *goals* and *plans,* reasoning about *planning goals,* or reasoning about *communication.*

5. It is widely recognized that an agent may enact several roles. None of the methodologies addresses the implementation of agents that need to represent and reason about *playing different roles.*

6. The methodologies, with the exception of the organisational rules, ignore *organizational norms* and do not explain how to specify and design them.

7. *Open systems* are not really supported. The methodologies implicitly suppose that agents are purposely designed to enact roles in a system. But as soon as agents from the outside may enter, the analysis, design and implementation needs to treat agents as given entities.

8. In the analysis, methodologies do not consider the *environmental embedding* of a system. The structure of the organization in which a system will be embedded, has a large influence on the type of organizational structure of the system, at least when it interacts with more than one person.

To overcome some of these problems we propose an alternative methodology for building life-like pedagogical agents. Within the next section, the proposed methodology is briefly highlighted.

3.3. A New Approach for Modeling Pedagogical Agents

All of the described methodologies so far have predominantly centred on developing specific applications and have not been integrated with existing methodologies. The general approach is to identify agents from an analysis of a specific application and then build specific agents for that application. Nevertheless, all above mentioned methodologies, typically, do not cover all phases of the software development life cycle (Figure 3-1).

Requirements
⇓
Analysis
⇓
Design
⇓
Implementation
⇓
Maintenance

Figure 3-1 Software development life cycle

For our aim, we find the design-based approach [Slo93] because it takes the stance of an engineer attempting to build a system to exhibit the phenomena/behaviour of interest. Complex systems can also be understood through a *succession* of designs, in the downhill mode of invention. Here, each design gradually increases our explanatory power and allows us to account for more and more of the phenomena of interest. Formally, this can be represented as a recursive methodology with five parallel threads of execution.

Threads 1-3 represent common engineering practices, and threads 4-5 give the methodology the rigour needed for scientific validity:

1) *Requirements gathering:* concerned with the understanding of the problem by studying an existing organizational setting; the output of this phase is an organizational model which includes relevant actors and their respective dependencies. This first phase does not significantly change for agent oriented software projects. Techniques for requirements gathering already exist, for example formalised specifications, use cases
2) *Requirements analysis of the system of interest*, i.e. a specification of the capabilities of the autonomous agent using information-level descriptions. These should include: the key features of the environment; the resource constraints within the agent; the behaviours the agent must exhibit and their causal links; and a description of the agent's concerns and coping strategies. This analysis phase is based also on OperA which captures the notions of norms and organisation structure.
3) *A design specification for a working system to meet those requirements*. This is an architectural analysis of the design, to include its major components and the causal links between these components. A design can be recursive, replicating threads 1-5 at individual component levels, i.e. a low-level implementation specification of one component and a theoretical analysis of another.

4) *A detailed implementation or implementation specification of the working system*. In this thesis we will develop a cognitive inspired agent architecture for elucidating “emotional” states of pedagogical agents.

5) *A theoretical analysis of how this design meets the initial requirements*. It is more than likely that an implementation will not meet all the requirements set out in the requirements analysis. A design verification analysis is therefore required to determine the extent to which: (a) the design meets the requirements; and (b) the implementation/simulation embodies the design. Ideally this should take the form of a rigorous mathematical proof, but in practice we must rely on intuitive analysis combined with systematic testing of the implementation.

Chapter 4: Modeling Pedagogical Agents: Requirements Gathering

First step of the proposed method is the Requirements gathering Phase. Within this phase, our aim is to identify the capabilities of the proposed tutor agent using information-level descriptions. These should include: the key features of the environment; the resource constraints within the agent; the behaviours the agent must exhibit and their causal links. A MAS is always situated in some environment, therefore we believe this should be as the primary abstraction during the requirements gathering, analysis and design phase. Generally speaking identifying and modelling the environment involves determining all the entities and resources that the MAS can exploit, control or consume when it is working to achieve its (organizational) goal(s).

4.1. Requirements for a Complete Model of Collaborative Virtual Learning Environments

Several studies [Gla98] have established that knowledge needs to be connected and organized in important concepts and this structure should allow transfer to other contexts. It was also shown that the learning process is getting improved when the students are in charge with their own learning, develop meta-cognitive strategies to assess what they know and acquire more knowledge if necessary. In other words the learning process must help students build knowledge from existing knowledge (constructivist learning), guide students to discover learning opportunities while problem solving (explorative learning) and help them to define learning goals and monitor their progress in achieving them (meta-cognitive strategies).

Applying these theories to distance education systems can lead us to a constructivist learning environment which encourages students to be more proactive in determining learning paths and synthesizing information from multiple sources. Hence, learners should not be constrained to a predefined learning path [Kin97]. This requires not only adequate tools but also the environment to allow meaningful interaction between the student and the learning system. According to Sims [Sim94] meaningful interaction is not merely pacing back and forth in a linear manner along prescribed paths but involves engaging the student with the learning content in a proactive manner.

Keeping in mind this, an interesting question arises from a system design and implementation viewpoint: “How do we design a truly interactive environment based on the learning paradigms presented above?” By truly interactive environment one can understand an environment which keeps the learner(s) motivated and interacts with them.

4.1.1. A Model for Collaborative Virtual Environments

Integrating CSCW, AI, multimedia and network technology, Collaborative Virtual Environment (CVE) becomes one of the most important modalities of the next-generation computing [God99]. CVEs allow people geographically dispersed to interact with each other in real time, share information, manipulate objects and work together to perform a common group task in remote virtual environments over the network. To solve many problems such as concurrent manipulation in CVEs, most researchers have developed some programming package and libraries to help programming instead of formal modeling [Car92].

A CVE is usually composed from three essential collaborative entities: *Hybrid Avatar, Participant* or its *Virtual Actor* and *Shared Object*. *Shared Objects* existing in CVE constitute common resources that *Hybrid Avatar* and *Participants* can operate on. Viewed as an intelligent entity, Hybrid Avatar is an essential component supporting collaborative activities in CVEs. Therefore, there adopted agent technology and concept of role form roles theory to implement the *Hybrid Avatar*. First we propose a formal definition for the overall CVE system as follows.

Definition 1: A role-based and agent-oriented collaborative virtual environment Π is defined as a five-tuple: $<\Sigma_s, U, SO, SR, R>$, in which, Σ_s includes a set of collaborative activities which are active in the current CVE, U involves all users in CVE, SO declares all shared objects besides media, SR indicates a set of resources such as floor, R is a set of roles correlated to users.

Collaborative activities are the most essential concept in the above definition. Therefore, we present formal definition for this concept:

Definition 2: A *Collaborative Activity,* a dynamic sequence of *Collaborative Event,* is defined as a three-tuple: $<\Gamma_{id}, G, EventQueue>$, in which, Γ_{id} is the identifier of *Collaborative Group* which takes this activitiy, *G* declares the collaborative goal that this activity is engaged in, *EventQueue* is an ordered queue of *Collaborative event,* which satisfies $\{(E_0, t_s), \ldots, (E_n, t_e)\}$ $n \geq 1$, $t_s < \ldots < t_e$.

Definition 3: A *Collaborative Event* is defined as a six-tuple: $<$*Event ID, Type,* Γ_{id}*, ProCondition, Operation, PostCondition*$>$, in which, *Event ID* is the identifier of *Collaborative Event, Type* declares kind that collaborative event belongs to, Γ_{id} is the identifier of *Collaborative Group* that this event acts on, *ProCondition* explains the precondition that enable this event to occur, *Operation* shows all operations that each *Intelligent Entity in Running State* takes, *PostCondition* points out postcondition after this event to trigger other event.

These two definitions involve *Collaborative Group,* the minimum unit to process collaborative activities. Before we give its definition, we discuss another new concept first: *Intelligent Entity.*

Definition 4: An *Intelligent Entity (IE)* is defined as a kind of visual collaborative entity which can model and sense its external virtual environment in real-time, plan its short - term actions according with its current roles and situation goals, revise its own status and execute all kinds of required collaborative events through its graphic representation. To describe dynamic state transition of *Intelligent Entities,* an *Intelligent Entity in Running State (IERS)* model is defined in BNF as follows:

<Intelligent Entity in Running State> ::= "INTELLIGENT ENTITY IN RUNNING STATE"
<ID> <Variational Attributes> <Rules><Goal>
<Target_Role_Rule Associated List>
"END_INTELLIGENT ENTITY IN RUNNING STATE"
<Target_Role_Rule Associated List>::= *<Target>< RoleList(Goal, t)>< RuleList(Goal, t)>*

The attributes of an *IERS* have the following meaning: *ID* is the identifier of *IE* to associate *IERS* with its correlative unique static definition. *Variational Attributes* involve all changeable attributes that are time based of *IE. Rules* is self-determined reaction rule base and its record is a set of clauses as follows: *IF Conditional Statement THEN Executing Triggered Collaborative Event. Goal* indicates a certain *Collaborative Goal* that the *IE* is engaged in at time *t. Target* is current *Target Status* determined by *Goal. RoleList(Goal, t)* is a set of *Roles* and for any *r~ RoleList(Goal, t) , r* depends on *Goal* and time *t. RuleList(Goal, t)* is a set of collaborative restriction rules educed from *Goal* and collaborative resource library including *Collaborative Rules Base.*

Definition 5: *Collaborative Group* model, a finite dynamic combination of *IERSs,* is defined in BNF as follows:

<Collaborative Group> ::= "COLLABORATIVE GROUP"
<Group ID> <Goal> <Scene> <Attributes> <Leader>
<Member_Role Associated List>
<Collaborative Rule List>
"END_COLLABORATIVE GROUP"
<Member_Role Associated List> ::=
<Member, RoleList(t)> {","<Member, RoleList(t)> }

The above attributes have the following meaning: *Group ID* is a unique identifier distinguishing the *Collaborative Group* from others. *Goal* is a *Collaborative Goal* that this *Collaborative Group* is engaged in. *Scene* is a unique scene where collaborative activity of this group is taking place. *Attributes* are a set of basic attributes of the group at time *t* including created time, current member number, and so on. *Leader* indicates that which one of the *IERSs* is the controller of *Collaborative Activity* at time *t* and assigns role of *Collaborative Leader* to it. *Member_Role Associated List* is defined as an aggregation of nested 2-tuple *<Member, RoleList(t)>* at time *t.* For each nested 2-tuple, *Member* is an *IERS* as a member of the group while *RoleList(t)* is a set of *Roles.*

Collaborative Rule List is a set of *Collaborative Rules* used by the group in *Collaborative activity* at time *t.*

Nevertheless, these definitions are better understood with an essential metamodel that describes human collaborative processes. These concepts are identified from real world phenomena and combined into a metamodel. Afterwards, the concepts are used in a design process to develop a conceptual model of collaborative learning applications. The metamodel concepts are supported directly on a workspace system.

4.1.2. A Metamodel of Human Collaboration

In this present study, "human collaboration in a group" means a set of intentional actions that one makes in order to help another member of a group accomplish a task or an activity that is relevant to the group. We argue that the existence of interactive tools such as e-mail, discussion lists, forums or chat sites is not enough to configure cooperative environments, even for working or learning. This assumption is based on the present approaches of Learning Social Theory (Engeström, [Eng99]; Wenger, [Wen98]). Such a theoretical model is based on the assumption that human beings continuously need to construct their identities to motivate them to participate in social activities. In such a context, every action is meaningful in terms of how people recognize themselves and are recognized by others.

For our purpose, we consider Hawryszkiewycz's metamodel as the result of social relations, in the specific activities of a given community, where people assume that specific roles and values exist, and these role and values are recognized and adhered to by everyone in the community.

The metamodel is shown in Figure 4-1 and has evolved through a variety of applications that include business networking [Haw96], strategic planning [Haw97] and is an extension of an earlier description [Haw02]. A further foundation for the conceptual model described here is organization computational theory [Car99] as a basis for managing knowledge about collaborative relationships. Thus rather than identifying data objects and processes, we identify organizational entities and their agencies. The model described in Figure 4-1 combines organizational structures such as activities and work-items, work processes including events and workflows, and social structures that enable groups to be formed and participants to be included in such groups. It provides ways to combine work-items into activities with members of groups assigned responsibilities through roles for those work-items. It supports social interactions, through group formations, discussions or notifications, as well as more structured workflows by associating events with artefacts.

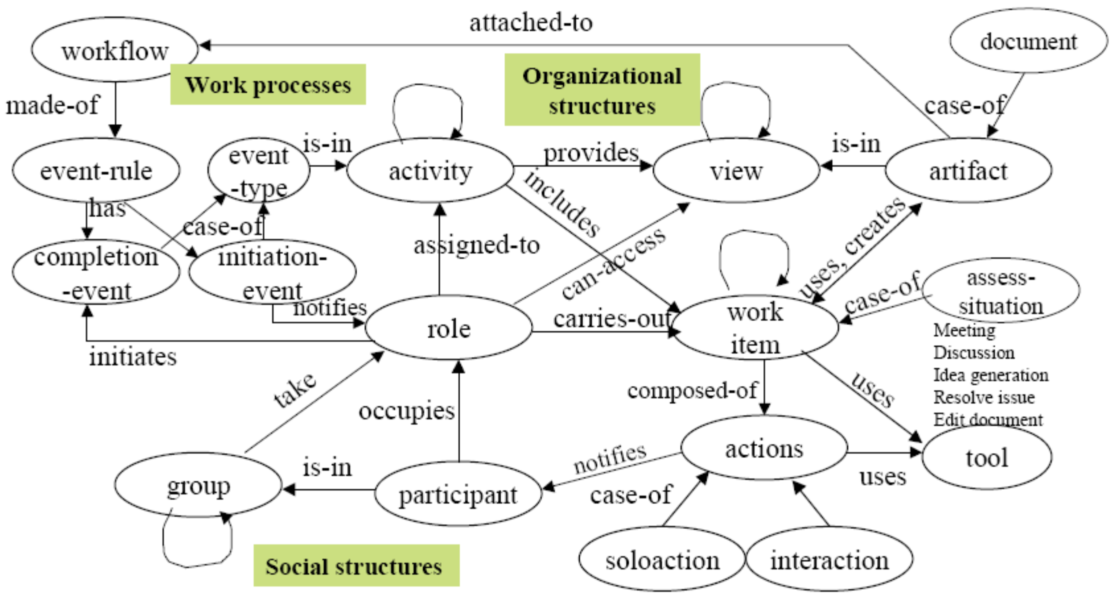

Figure 4-1 A Metamodel of Collaborative Processes [Haw02]

i. **Artefact** - data objects such as documents, calendars. It can also be a record of discussions or other personal interactions.

ii. **View** – a collection of artefacts. These can be documents, calendars, or multi-media records. They can also be other views.

iii. **Activity** - produces a well defined artefact as its output (e.g. Produce a planning document), and can include many work-items to do so. Provides views, which are needed to carry-out the work-items.

iv. **Role** - defines responsibilities in system in terms of work-items that it can carry out and the views that it can access

v. **Participant** – a specific person that is-in a group and can be assigned to a role

vi. **Group** – a collection of participants that can be assigned to a role.

vii. **Work-item** - a set of actions and interactions needed to produce intermediate outcomes that eventually produce an activity output (e.g. Review part of a planning document - which may include a number of actions, assess a situation). A work-item is composed of a number of actions and provides tools to carry out the actions. They can also represent the knowledge management activities:

viii. **Action** - a specific unit of work carried out by a role (e.g. change an artefact, send an artefact). Can notify selected roles when completed. Can be a:

a. **Solo-action** – carried out by one participant, or

b. **Interaction** - the basic exchanges between people when they collaborate in the activities. **Event type–** is in an activity and can either be an 'initiation-event' that notifies a role in the activity to carry out work-item, or is a 'completion-event' initiated by a role following completion of a work item,

ix. **Event-rule –** defines the next initiation-event or events to be activated following the completion-event

x. **Workflow** - describes a sequence of event rules. It can be attached to an artifact.

The model also includes a variety of commands that can be used up to setup and change systems specified in terms of the model. These include ways to create new groups, activities or work-items and set up the necessary views, workflows and notifications. A typical set of steps followed to create a model are:

1. Define the high level activities in the system and the work-items in each activity.

2. Define groups or teams and add participants to the groups.

3. Add artefacts needed by the work-items and create views for the work-items to identify the artefacts needed by the work-item.

4. Define roles and assign responsibilities for work-items to roles and then assign participants to the roles.

5. Expand work-items in terms of their actions and identify views for these to 'use or create' artefacts.

6. Identify any predefined events and consequent rules and specify the 'initiation-events; to be activated following a 'completion-event'. Workflows are 'made of' that specify sequences of event rules.

According to this approach, human activities in which cooperation can be identified are those that have something more than a common objective, a shared vocabulary, and the possibility of interaction. Wenger, [Wen98], points to engagement as one significant element. For us, it is still a fuzzy concept, in the sense that "engagement" is a broad concept that can include many others. For our purpose, we consider "engagement" as the result of social relations, in the specific activities of a given community, where people assume that specific roles and values exist, and these role and values are recognized and adhered to by everyone in the community. Roles and values help people project and reflect images of identity. If cooperation is part of such roles, then it can become a value and, through its practice, can help develop cooperative attitudes among the members of a community. Thus, cooperative-learning environments need something more than technological frameworks to allow for interaction among people. We believe that technological, economic, and social models can best allow for cooperation. Through these models, people can construct their identities by assuming roles that are recognized and prized by the others who are participating in the same effective, cooperative-learning environments.

The next step is to use the metamodel concepts to describe the synchronous collaborative learning environment and afterwards, its extension to agents.

4.1.3. Requirements of Synchronous Collaborative Learning Environments

Collaborative learning is an advanced group learning in which all students share the same process of problem solving at once, exchange their opinions or questions/ replies mutually in accordance with the discussion progress or actions/reactions among students, and attain to the learning goal with each other [Koj98], [Koj99].

The use of collaborative learning in classroom has proven to be effective in improving academic achievement [Hor98], motivation for learning [Joh90], and social interaction [Sla96]. Nowadays the interest in collaborative environments has increased considerably, mainly due to the current technological advances especially on the Internet computing [Pin01].

In the context of distributed collaborative learning, it is usually difficult for students to be aware of others' activities and for instructors to overview the process and regulate the collaboration. Based on the activities which can take place in such environments, the resources required are as follows:

- Technologically mediated communication channel
- Shared workspace for the group of students
- Individual workspace
- Learning materials/ learning tools

For a synchronous collaborative learning environment, we propose an ensemble of services allowing students to work in small groups (a maximum of 4 students per group) in order to propose solutions to a problem presented by an instructor. We assume that learning is a discovering process and that the evaluation of a student must consider such points as the quality of the solution proposed by the student's group, how the student uses and connects concepts belonging to the subjects under evaluation, how the student collaborates with the other group members, and whether the student acts for the benefit of the group.

For us, the general characteristics of such an environment are as follows:

▪ A pedagogical approach encouraging the student to be creative in the knowledge-discovering processes.

▪ An assessment process that considers content-related capabilities and social capabilities, individual learning processes and their results, and collective learning processes and their results.

▪ Students working in small groups so that they can easily learn about each other.

- A team based learning approach (e.g.: project-based learning approach).
- The use of portfolios to visualize learning processes and authorship.
- The use of ideas to evaluate students—One “idea” is made of a hypothesis (proposed by one student), some arguments (documents that the student chose in a digital library), and at least one intellectual product (a document produced by the student after reasoning about his or her hypothesis and the arguments he or she found). Arguments are related to the hypothesis by semantic links.
- The use of portfolios to organize ideas in semantic nets.
- Assessments done not by evaluating students according to a final work, but by considering its quality and the contributions of every group member.
- The use of document annotation—Students can make semantic marks on the documents they choose. These marks explain how the documents can contribute to an idea. Annotations are important, because they make explicit to students what they know about something and can also be used by teachers to monitor and intervene in the students’ learning processes.
- An assessment method that considers both individual and group portfolios—The assessment criteria adopted may consider dimensions like richness of ideas (originality of hypotheses, robustness of argumentation, and quality of intellectual products) and collaboration (suggesting arguments or intellectual products in response to others’ ideas, negotiation ability, responsibility, and responsiveness).
- A learning process composed of cycles of individual and group phases— A limited number of interactions is necessary for the students to negotiate the construction of the group portfolio, and also for teachers needing time when analyzing the data present in the learning environment and deciding how to intervene in the learning process.
- An individual phase in which every student creates and organizes his or her ideas inside his or her portfolio, studies the assessment criteria that the teachers will use, and prepares his or her arguments to convince the other group members that these arguments will benefit the group’s project.
- A group phase in which students submit their ideas to their peers, who vote on the best ones to keep in the group’s portfolio—Before voting, students defend the quality of their ideas against the criticisms of the other group members.

In order to implement the above characteristics, many specialized and transparent services are necessary. Identifying such services is a difficult task requiring the use of appropriate tools, which help to identify the necessary services and the agents that will implement them.

Table 4-1 Primary services necessary in a synchronous collaborative learning environment

Service Description	**Teacher**	**Student**
Editing a portfolio (declaring a hypothesis, linking it to arguments and to an intellectual production)		X
Searching for arguments (documents) in the digital library	X	
Searching for arguments (documents) in the Internet	X	X
Introducing and indexing new arguments (documents) in the digital library	X	X
Submitting an idea to other group members		X
Voting on an idea to be added into a group portfolio		X
Enriching someone else's idea	X	X
Visualizing and evaluating one's own performance	X	
Visualizing a student's performance	X	X
Visualizing a group's performance	X	
Finding people in other groups with ideas "close" to one's own	X	X
Annotating pedagogical strategies and actions adopted with students	X	
Sending messages to other group members		X
Sending messages to a student or a group	X	
Evaluating one's performance in the use of the digital library	X	
Evaluating one's performance in the negotiation of ideas	X	
Evaluating one's performance in collaborating with ideas of others	X	
Evaluating one's performance in the creation of intellectual production	X	
Evaluating one's performance in performing social duties	X	

4.2. Basic Requirements for a Pedagogical Agent in a CVE

The result of all of the changes that have taken place in educational system can be seen as the increasing emergence of complex and dynamic educational infrastructure that needs to be efficiently managed. Corroborating this, new (types of) educational mechanisms and services need to be developed and supplied. In particular these services need to satisfy a series of requirements such as personalization, adaptation, support for user mobility, support for users while they are dealing with new technologies, among others. Agents emerge to provide solutions for these requirements in a way that is more efficient when compared to other existing technologies [Aro99]. Nevertheless, there is also a set of requirements which a designer has to follow when he implements agents for such environments. Furthermore, we structure these requirements into three trends for a better understanding.

4.2.1. Requirements for autonomous agents

An autonomous agent is required to produce coherent, effective and robust behaviour [Bee90] in a complex and unpredictable domain such that its goals are achieved. These requirements for successful operation in unpredictable, dynamic real time environments pose certain design issues.

The computational resources of the agent will be finite. For example, humans find it difficult to listen to more than one conversation at once. Also, the agent will have several physical constraints like: it will be able to move at a certain pace, manipulate a finite number of objects and so forth. Good design solutions will manage an agent's finite resources as efficiently as possible, even though efficiency is a difficult notion to define [Slo95].

An autonomous agent is capable of producing its own goals but has limited resources with which to satisfy them, the fundamental goals of the agent may have been designed by a human engineer, or evolved by natural selection, and they may be enduring throughout the lifetime of the agent, or subject to modification. The agent will need to pursue multiple goals, with perhaps conflicting objectives. It will have many divers tasks to perform. In addition goals might have associated temporal constraints. Therefore , the agent needs to schedule its goal processing and actions. This requires the ability to select between multiple motives [Slo85], prioritize goals, decide on a level of commitment towards current intentions, and notice opportunities for actions that satisfy more than one motive.

The requirement for time scheduling of actions and the constraint of limited resources, both computational and physical, require that current processing be interruptible. For example, to react to new, motivationally relevant events in the environment the agent will need to interrupt its ongoing processing and switch its 'attention' to new contingencies [Slo90].

The unpredictability of the environment renders complete planning prior to action impossible. Instead, opportunities and threats to plan will need to be constantly monitored for. To detect such events the agent must be able to generate motivations asynchronously to current processing. A level of coarse-grained parallelism is therefore necessary [Mae90]. Architectures that model autonomous agency will need to integrate a wide range of behavioral capabilities.

4.2.2. Requirements for agents as ITS

The choice of intelligent pedagogical agents is based on online students' need for good support in distributed collaborative learning environments. The role these agents have to play is new, and they have to deal with three simultaneous dimensions:

1. The technical (how to work with- and use the environment's available tools)
2. The pedagogical (how to construct their own representation about a given domain and use a graphic environment to represent it)
3. The strategic (how to use and develop their own social competences to achieve their goals in the collaborative learning scenario)

Educators responsible for following up details from all three dimensions would not be able to pay sufficient attention to the aim of the learning process. Intelligent agents can monitor students' steps and, according to the knowledge models they have, inform students about procedures the students are not yet used to.

In the context of distributed collaborative learning, it is usually difficult for students to be aware of others' activities and for instructors to overview the process and regulate the collaboration. In order to facilitate collaborative learning, intelligent agents were developed to support the awareness and regulation of the collaboration.

Malone, Grant, and Lai [Mal97] reviewed their experience in designing agents to support humans working together (sharing information and coordination). From the experience, they found two design principles:

- Semiformal systems: Do not build computational agents that try to solve complex problems all by themselves. Instead, build systems where the boundary between what the agents do and what the humans do is flexible.
- Radical tailorability: Do not build agents that try to figure out for themselves things that humans could easily tell them. Instead, try to build systems that make it as easy as possible for humans to see and modify the same information and reasoning processes their agents are using.

The design of our educational agents follows these two principles. On one hand, the agents are designed not to replace instructors but to work together with them to support the collaboration. On the other hand, the agents can be started, stopped, and turned off at the will of the users (students or instructors). We will also allow the users to customize the services provided by agents.

Maes, [Mae97], claims that there are two more concerns when agents are built: competence and trust. Competence refers to how an agent acquires the knowledge it needs to decide when, what, and how to perform the task. In our case, will the agent depend only on the rules written by the instructor? Should it be able to improve its performance by learning? For agent systems to be truly “smart,” we believe that they would have to learn as they react and interact with their external environments. The ability to learn is a key attribute for intelligent agents. Trust refers to how we can guarantee that the user, in our case the instructor, feels comfortable in following the advice of the agent or delegating tasks to the agent, for example, letting the agents send e-mails to students directly without the instructor’s confirmation. It is probably not a good idea to give a user an interface agent that is sophisticated, qualified, and autonomous from the start [Maes97]. That would leave the user with a feeling of loss of control and understanding. We have tried different methods. One of our solutions is that at the beginning, the agents work together with the instructor, providing advice and explaining its reasoning process. Gradually the agents learn from the instructor’s feedback on its advice, improve their performance over time, and build a trust relationship, until a point is reached where the agents are allowed to perform actions without confirmation from the instructor.

Pedagogical agents are defined, according to Johnson et al. [Joh00], as “autonomous and/or interface agents that support human learning in the context of an interactive learning environment.” They are built upon previous research on intelligent tutoring systems (ITSs) [Wen87]. Many researchers have designed and developed pedagogical agents for ITSs [Joh97], [Les99], [Cas00], where the agents play the role of a guide or tutor.

The role of the educational agent is to provide task-related feedback and assistance to the learner and to guide the learner through the learning process and help the learner reach his or her learning goals. In an education environment, multiple agents are usually involved, and each agent plays different roles. There are two aspects to be considered in designing and building educational agents:

- *Reusability:* Reuse agents in different kinds of systems and environments.
- *Interaction:* In an environment containing multiple educational agents, tutor agents interact with each other and customize their behaviors based on the behaviors of other agents in the environment.

Another major requirement of pedagogical agents is to *motivate learners*. The goal of motivation and emotion research in ITSs is not to make the tutoring system able to simulate emotions, but to provide a personalised learning environment that considers cognitive, emotional and motivational aspects, assisting students as whole and unique beings, associating reasoning and emotion. According to Viccari and Giraffa [Vic98], in order to provide individualised assistance to students, it is necessary more than percentiles or comparisons with pre-defined models – the tutoring system must consider dynamic information about the student, changing with time.

Carbonel [Car70] proposed a new approach to deal with computer assisted tutoring processes, taking into account the dynamic relationship between teachers and students in the classroom. In his approach three basic architectural modules are considered to integrate the learning and teaching tasks: the knowledge domain, the observable student's behaviour, and the set of strategic rules to be used by the tutor.

According to Burns and Capps [Bur88], to be considered an ITS a software must keep three characteristics. The first is to be related to a knowledge domain from which the tutoring system should "know" enough to behave like an expert, in terms of being able to infer solutions or to solve problems. The second aspect is that the software must be able to deduce the student's knowledge levels. The third characteristic is that the teaching strategy should be designed to reduce the knowledge gap between the expert and the student. In this approach the *expert* possesses domain-knowledge of the focused area. These three characteristics suggest a three-dimensional ITS framework, and should be accompanied by pedagogic principles so the tutoring system may succeed in promoting a proper learning environment.

Moreover, the perception of students as whole beings and that educational software must observe their wholeness and individuality, reinforces the interdisciplinary aspects inherent to ITS development, increasing its dimension and complexity.

In the architectures proposed by Carbonel and Burns & Capps ([Car70], [Bur88]), the pedagogic aspects are considered while modelling students, declaring the teaching strategies and in building the teaching/learning activities. However, they do not verify the motivational or emotional aspects contained in the theoretic framework provided by Psychology.

Despite its great importance, only recently the emotional and motivational aspects became ITS research objects. According to Goleman [Gol95], the emotional interferences in the mental life of a student are not new for teachers. Anxious, angry or depressed students do not learn; people in these conditions do not absorb information efficiently, consequentially it is an illusion to think that learning environments that do not consider motivational and emotional factors are adequate.

4.2.3. Requirements for believability

Nevertheless, the common research trend in designing animated agents is to make the life-like or believable [Bat94]. According to Bates [Bat94], a believable agent is one that is able to express emotion and to exhibit a given personality. De Rosis [Ros05] claims that an agent is made even more believable if it can behave in ways typical of given cultures, and finally, if it has a personal communicative style [Can97]. "This is, in fact what makes a human a human" states the author. An autonomous agent that fulfils all of these constraints is one where the communicative output, that is, the particular combination of multimodal communicative signals displayed (words, prosody, gesture, face, gaze, body posture and movements) are determined by different aspects like: contents to communicate, emotions, personality, culture, style, context and user sensitivity.

Life-likeness is supposed to provide the user with the illusion of life and believability should allow users to suspend their disbelief. Due to the fact that characters can be life-like in a “human-like” or an “animal-like” way an ongoing debate concerning whether the life-likeness of characters is more effectively by a realistic or by a cartoon style agents.

The answer to this debate can eventually be given empirically with respect to specific application scenario. For instance, while Blumberg in his thesis [Blu96] conducts a series of investigations on animal like characters, especially dogs, Thalmann [Tha97] aims to create virtual humans typically following the realistic approach, even strives for photorealism.

In her thesis [Kod96] Koda created a Web-based poker game in which a human user could compete with other personified computer characters including a realistic image, cartoon male, female characters, smiley face, no face and a dog. She gathered people’s impressions of the characters and she discovered that people’s impressions of a character were different in a task context than in isolation and were strongly influenced by perceived agent competence.

The work of Nass & Reeves [Ree96], [Nas00], focuses on the study of computers as social actors. They have conducted a series of experiments that examined how people react to computer systems and the applications that incorporate certain personified characteristics. They demonstrated that users like computers more when the computer flatters them. Their most important discovery is that people interact with and characterize computers in a social manner, similar as they do with other people. More exactly, they found that existing, accepted sociological principles (e.g. participants with similar personalities tend to get alone better that do those with different personalities) apply even in the case when one of the participants is a machine.

A related empirical question concerns the benefits of displaying life-like characters as facial agents (talking heads), full-body, or upper-body plus hands agents. Unfortunately, there is no predefined standard for designing synthetic agents due to the fact that the application context decides, by and large, what characteristics an animated agent should have. Still there are several major aspects one should keep in mind when designing animated agents:

- *the look*: is the agent going to represent a person or some other living creatures (e.g. animals)? Is the agent meant to be realistic (human-like character) or it is artistic (may be an exaggerated cartoon-like)? In case of a human-like character should the agent have a gender: male or female, if yes why?

- *Physical details*: what parts of the body are covered by the agent: head, torso or full body? Are the body used to express emotions? Can the character change its location? Are the hands used in coordination with speech (e.g. gestures for enumeration) or to point something? What varieties of hand gestures can be used? What about the face of the character? Does the face express emotions, approval/disapproval? Are the facial expressions meant to be realistic or characteristic of a real given group or should be generic?

- *Communication modality*: how should the agent communicate with the human user(s) from its environment? Should the agent have the ability to use speech or only text?

- *Mental model*: should the agent have its own mental model? Should the agent posses the ability to express emotions? Should the agent have its own personality or not?

- *Interacting with the agent*: how does the user interacts with the agent? Who is able to control the agent: the user directly (e.g. avatars), some applications (e.g. presentation agents) or both (e.g. tutoring agents)? This aspect highlights also the concern how to define what and how is perceived from the user.

Another (and maybe less emphasised) requirement concerns *moderation* in the external expression of thoughts and moods and 'harmonic variation' of this expression. The Agent should avoid extreme manifestations of its thought, unless this responds to specific needs, and should display specific aspects of its mental state at the 'right' moment, by varying them gradually in agreement with the content of her communication. The way these requirements may be implemented depends on the type of character and on the application in which this character acts. Cartoons are 'excessive' by their very nature, they systematically (and on purpose) violate the moderation requirement; their reactions to events are immediate and disappear very quickly, to leave place to reactions to new events. The same behaviour does not apply to the case of a *realistic* Agent: the more an Agent aims at being realistic rather than willing to amaze or astonish the user, the more complex its implementation becomes. A 'consistent' behaviour cannot be produced, in a realistic Agent, through a 'scripting' implementation process, but should reflect a powerful cognitive model of the Agent and an adequately complex reasoning process

Since we are interested in dimensions that have an effect on trust – which can be considered as one of the key factors of an agent's believability – we follow the approach of Svennevig [Sve99] based on three dimensions of the interpersonal relations in conversations: *familiarity*-highlights the way in which relationships develop through a session, *power* – ability of one participant to control the behaviour of others, and *solidarity* – having similar behaviour dispositions.

4.2.4. Requirements for cultured embedded agents

Agents may be built so as to be tailored to a particular culture or may be designed so as to adapt to different cultural contexts. In the first case, every time a new application context comes out, the agent has to be redesigned: Its way of thinking, appearance, and behavior have to be modified, and changes have to be introduced so as to ensure that consistency is not lost. In the second case, the long-lasting experience of user-adapted interfaces may be applied to design and build an agent whose mentioned characteristics change more or less naturally according to the context. In the most sophisticated case, changes may occur automatically: The agent observes the environment to understand the situation and progressively models it and decides how to behave. To have such a complex adaptation capability, the agent should hold an explicit knowledge of how to interpret situations and how to modify itself accordingly.

Interpretation and plan recognition are so far the most difficult components of user-adapted systems; one might then imagine systems that do not adapt automatically to the situation, but in which adaptation is introduced at the agent's design level. Although it is reasonable to assume that a person's body is a reflex of his or her mind and that this assumption should guide the generation of embodied animated agents as well, we claim that separation of the agent's mind from its body helps achieve a higher flexibility in adaptation of the agent's behaviour, acting, and appearance to the cultural context. It offers the opportunity of varying its mental state and reasoning style according to the context and establishing the communication forms to be applied after considering the technical resources available;

If the agent's mind is represented according to the BDI (belief, desire, intention) theory and architecture [Rao95], the way that its mental state is related to the cultural context may be represented explicitly. This enables the agent to vary the decision taken (including the discourse that achieves a given communicative goal) and the emotions triggered and displayed according to its mental state's structure;

Relations between the various components of a person's mind, and the way they control the affective state, cannot be established in a rigid and fixed way: Various forms of indeterminism govern this process from uncertainty attached to beliefs and their relations to weights associated with achieving terminal and instrumental goals. In addition, body expressions are not always specific: A gaze may have several meanings, and the same meaning may be conveyed through a combination of several signals. Again the way meanings and signals are related is not rigid and fixed, but governed by uncertainty. Therefore, some formalism is needed to appropriately represent these parameters in the agent's mental state and the way they affect its behaviour.

Cultural differences in norms, standards, and goals underlie cultural diversity, and these are reflected in differences in the way people reason, feel and display emotions, appear, and gesture, it is natural to wonder whether these differences affect human– computer interaction. In particular, should globalization be interpreted as going toward uniformity of interaction styles or should interaction with a machine be adapted to the cultural context to which it is applied, as it tends to be for other factors such as the user experience, preferences, interests, and so on?

Adaptation is usually justified by some empirical demonstration (or some well-grounded assumption) that it improves, in some sense, the usability of the application. So before investing efforts toward culture-adapted interaction (and, in particular, characters), some evidence should be given that this will improve the use of the systems to which it is applied.

In the scope of their computers as social actors long-term research plan, Nass [Nas00] and colleagues examined this question: Does the ethnicity of a computer agent have an effect on users' attitudes and behaviours? In a study comparing a group of Americans with a group of subjects from an ethnic minority (Koreans), they found that ethnic similarity had significant and consistent effects on the users' attitudes and behaviours. When the ethnicity of the subject was the same as that of the computer agent with whom the subject was interacting during the experiment, the agent was perceived to be more similar, more socially attractive, and more trustworthy. The agent's arguments were also perceived to be better and more convincing [Nas96]; [Nas00].

In lack of a strong background of empirical studies involving groups of different nationalities or ethnic origins, one may only speculate on the domains in which adaptation to culture might be justified. Lee [Lee01] and Nass [Nas96] proposed the domains of recommendation in general (e.g., training or medical advice), online shopping, and advertising. They claimed that users would be more willing to trust in the agents, take their suggestions, or give them their credit card number if these agents display a matching of values and norms with their own scale of values. Access to services in general and teaching are other domains in which expectations and communication forms are presumably informed by culture. In service systems and shopping, one of the differences between short- and long-term orientation cultures (or between masculinity and femininity) is in their being goal-directed versus being viewed as an opportunity for "living an experience" or "initiating casual conversations." Therefore, the kind of small-talk dialogues being developed in REA [Cas00] seems to be appropriate in the second type of culture, whereas in the first one helping the shopper get out of the shop more quickly might be preferable. Tutoring systems, on their side, reflect pedagogical approaches and viewpoints about teacher–student relationships: Tutoring agents have therefore to reflect these approaches and viewpoints in the culture in which they will be applied.

However, adaptation to culture did not yet influence HCI as it presumably should: Only a few cues on a tendency to act may be observed. For instance, although emoticons are ubiquitous, some cultural difference occurs even here: The Japanese emoticon for smile depicts that women are not supposed to show their teeth when smiling, and the second most popular icon in Japan is the cold sweat, with no clear Western equivalent [Don01].

Summary: the answers to the above questions are strongly dependent on the application context which is the major decision factor in the design process of such agent. Even though there are several requirements which every synthetic agent must satisfy:

- *a graphical representation:* it is a fundamental requirement to allow the presence of an agent within a CVE.

- *awareness:* it refers to the ability of the agent to perceive its environment and the changes that occur in it.

- *locomotion:* it is the ability which allows the agent to travel in its environment or in a network from one host to another

- *object manipulation:* capacity to perform different actions upon the virtual objects in the environment

- *communication:* ability to communicate with users or other agents through different ways using different communications protocols

- *roles:* a role defines those behaviors, characteristic of one or more persons in a context. In our case each agent should play different roles that are associated with different rights in different situation to enforce.

- *Goals:* a goal defines a special state that agent is pursuing to achieve in a current situation. There can be two types of goals: collaborative goals in case of multi-agent system, or individual goals for itself.

- *Reactivity:* ability to respond quickly to environment's changes. This ability is generally defined as a series of event-condition-action rules like: *ON event IF conditions THEN action* which means: when event occurs and if the evaluated conditions are true then action in rule are executed.

- *Planning:* ability to elaborate an action sequence to achieve their goal in a situation.

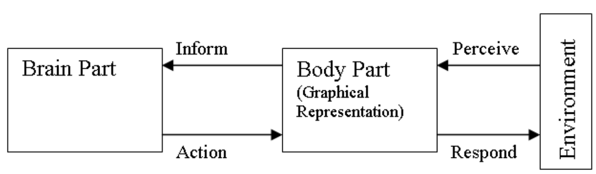

Figure 4-2 Generic architecture for a synthetic agent model

Based on careful analysis of the above requirements a generic architecture can be proposed for synthetic agents (see Figure 4-2). The model is composed from two components: the body part or the graphical representation of the agent in its environment and the brain part – the part which is responsible for all agent's actions. We chose a human character instead of an animal in order to impose learners a degree of realism due to the fact that the environment where our prototype acts is virtual environment for distance education system.

Chapter 5: Requirements Analysis Phase

Civilization advances by extending the number of important operations that we can perform without thinking about them.
A.N. Whitehead (1861–1947)

The aim of the analysis phase is to facilitate design and implementation, at a level of abstraction that is more adequate for complex information systems, by introducing the single, main concept of agent. We provide an outline of the basic ideas that we think are useful to this analysis and a preliminary sketch of the proposed agent-oriented design methodology. The sketch provided here serves to illustrate the main ideas and also to introduce the main research issues.

5.1. A Role Model

Roles have been introduced in MAS community as a way to coordinate the behaviour of individual agents by means of a normative system or an organization. Roles are associated with expertise, capabilities such as planning rules, and with responsibilities to maintain or achieve some state of affairs. Roles are often also associated with obligations and permissions that restrict the means by which they can fulfil the responsibilities. In our case, the basis for a methodology for agent-oriented design can be derived from the most important concepts in the agent-oriented approach, namely that of agent and group. A third design issue - that of the interaction between agents and groups - is inferred immediately from these notions. One of the basic concepts in our proposal for a methodology is that of a role. Roles are associated with agents and are composed of skills, knowledge and goals. In our design methodology a number of steps concern the further refinement of these skills, knowledge and goals of roles. There is already a model for an agent within a group:

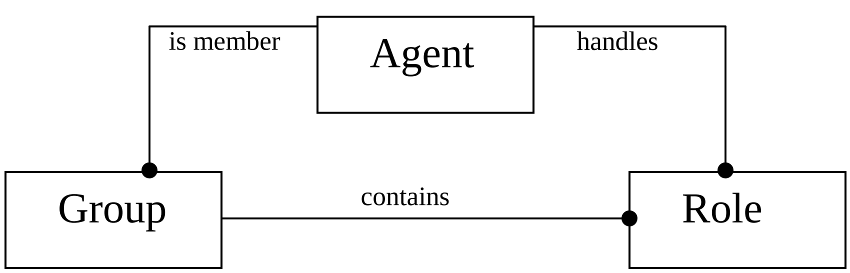

Figure 5-1 Model of an agent in a group

It is assumed that an agent is an active, communicating entity which plays roles within a group. The role means an abstract representation of an agent function, service or identification within a group. Each agent can handle several roles, each role handled by an agent must remain local to a group.

5.1.1. Roles and Role Models: Background Theory

Many modelling approaches use roles as basic building blocks. For example, roles are used in organisational theory [Sco92] to represent positions and responsibilities in human organisations. Roles are also used in software engineering [And97]. Roles are particularly suitable for modelling the behaviour of software agents, e.g. [Dep01], [Ken99]. Agent roles are defined in a manner similar to organisational roles referring to a position and a set of responsibilities in an organisation [Fer91b]. To better represent agent concepts, the agent role definition includes additional characteristics, for example planning, co-ordination and negotiation capabilities [Ken99].

Existing role-based approaches to multi-agent system design stress the need to identify and characterise relations between roles [And97], [Ken99]. Role identification was based on organisational principles and in particular on *role theory* [Bid79]. The essence of role theory is that persons are appointed to roles within an organisation, which are representations of concrete behaviour. This behaviour is characterised by authorities describing things that can be done and responsibilities describing things that must be done. For example, directors, help-desk staff, developers and test engineers are all associated with job descriptions specifying their responsibilities in the organisation. Organisational goals, policies and procedures further determine their rights and duties within the departments, projects or groups of which they are members.

Role theory emphasises that various relations may exist between roles. For example, an examiner cannot be a candidate at the same time and therefore appointing these roles to a person at the same time results to inconsistency. Role relations can be complex. For example, a university staff member who is also a private consultant may have conflicting interests. In this case, appointing these roles to the same person is possible but it would require appropriate mechanisms to resolve the conflicting behaviour.

Following [Ken99] a role is defined as a *position* and a set of *characteristics.* Each characteristic includes a set of *attributes.* Countable attributes may further take a range of values. More specifically, a role is considered capable of carrying out certain tasks and can have various *responsibilities or goals* that aims to achieve. Roles normally need to interact with other roles, which are their *collaborators.* Interaction takes place by exchanging messages according to interaction protocols.

According to Cabri [Cab04] a role *"is a set of capabilities/knowledge and expected behavior which can be assumed, used and released, accordingly to a set of starting-requirements. The above requirements are needed to assume the role and must be matched by current capabilities of the agent. The capabilities/knowledge added by the role improve the agent ones (intended before the assumption), in the case allowing it to* *assume other roles. Finally the expected behavior represents a set of duties that the agent playing the role has to take into account and that other agents (playing other roles) can rely on during interactions".*

Roles can be extended to create specialised roles by a process called role *specialisation* or *refinement* [And97], [Ken99]. Specialised roles represent additional behaviour on top of the original role behaviour in a manner similar to inheritance in object-oriented systems.

In order for roles to pragmatically represent behaviour in an application domain, they need to model issues relevant to non-functional requirements in that domain. Therefore, the above role definition is extended to include *performance variables.* Performance variables are parameters whose value defines the run-time behaviour represented by a role. For example, if the behaviour a role represents requires using some resource like memory, the resource capacity can be modelled by a performance variable. Performance variables can also be defined at an agent level. In that case, their value is a function of the values of the respective performance variables of all roles the agent is capable of playing. This allows us to apply design heuristics by imposing constraints on the values of the agent performance variables that must be observed when allocating roles to agents.

A collection of roles and their interactions constitutes a role model [Cab04]. A role model represents the behaviour required to carry out some activity in the system. An agent application normally consists of more than one activity and hence it will involve more than one role model. Role models that occur frequently in some application domain are called *role interaction patterns.* Role models can be used to represent reoccurring complex behaviour based on multiple points of interaction.

Therefore, they are considered to be first class design constructs, the entities that can be instantiated and given identity. Role models can be used to describe both application behaviour and organisational settings. An agent system designer should be able to reuse role interaction patterns and specify new role models as required. Therefore, the problem of designing an agent organisation refers to selecting and instantiating suitable application and organisational role models.

5.1.2. A Role Metamodel for Pedagogical Agents

We view agents in a virtual society of learners as entities that occupy a social position and perform several roles. Giddens [Gid97] defines social position within a group as the social identity an individual has in a given group or society. Biddle [Bid79b] defines roles as those behaviours, characteristic of one or more persons in a context. In our case a role specifies a characteristics pattern of behaviour for the interactions of the agent so that the agent which plays that role behaves in a specific way under certain situations involving other learner(s) or agents.

```
<Role >::= “ROLE”
    <Role ID>
    <Skills>
    <Roleset>
  <Prerequisites>
<Responsibilities>
   “END ROLE”
```

Figure 5-2 BNF specification of Role class

Figure 5-2 shows the Role class written in BNF specification. Role ID is used to distinguish a role from other roles. A skill can be defined as the ability to carry out a task at a pre-defined level of competence. In our concept Skills of a role describe the properties (or the abilities) that the agent will need to possess in order to perform successfully the role. Skills should be linked together with roles: if an agent knows what role it has to play then it also knows the skill(s) required to successfully accomplish that role. As far as the procedures (the agent is developed in Borland Delphi, and a class in Delphi has procedures similar with the methods from a Java class) of the Role class there are a group of procedures which associate Role class with skill class:

- *TAddSkill()* - this procedure is used to bind a particular role with a particular skill
- *TRemoveSkill()* – destroys the link created by TAddSkill
- *TGetSkill()* – returns all skills relevant to the role under consideration.

Roleset refers to a set of roles that agent interacts with given this role. Prerequisites of the role refer to the credentials an agent needs in order to occupy the social position under that role. Responsibilities of a role refer to the duties of an agent undertaken within the context of the actual role.

Tutoring agents are entities whose ultimate purpose is to communicate with the student in order to efficiently fulfil their respective tutoring function, as part of the pedagogical mission of the system. Tutoring functions may include the following [Mor98]:

- Select a subject element.
- Format and present a subject element.
- Format and present an explanation of a subject element.
- Compare different concepts.
- Select, format, and present an example.
- Answer a student’s question.
- Evaluate the student’s answer to a system-asked question.
- Send feedback to a student about his answer to a system-asked question.
- Diagnose a student’s behaviors.
- Update student model.

The main important roles which our prototype should have are:

- Tutoring role: it can present new topics during a learning-session, ask questions and understand the students' responses, and the students can ask clarification questions and receive appropriate explanations but it should be able also to provide individual help to a participant if it is necessary.

- Follow-up role: it can monitor the whole collaborative process and also individual activities. After each session the agent provides to a human-tutor a full report concerning the activities of each participant within that session.

To exemplify the concept in an educational environment the tutor-agent should have the following skills (capabilities) within a group of students:

- *Interrogator* – poses questions and the students of a collaborative group then provide answers. The questions should provide help for the students to reach a common learning goal.

- *Reviewer* – analyzes the students' answers, including whether it is correct or not.

- *Monitor* – records the answers from all the students and the communications among students during the collaborative learning process.

- *Instructor* – gives individualized instructions and helps those students who cannot keep up with the progress of their group-mates.

- *Group Manager* – has the ability to control the coherence of the group.

Let's take for example the *Reviewer* role: the skills required for this role are: agent should be able to understand the student's answer (natural language processing) then it should analyze whether it is correct or not. An important skill of this role can be considered the ability to display emotions and gestures (animations) to students' answers. Our prototype responds to the students' answers (also to questions) by synthetic speech, facial display and gestures.

The *Roleset* for this role can be considered as the set composed from the roles ={*Interrogator*, *Instructor*}. Of course the *Prerequisites* for this role can be simply deducted: if the students answer to the question posed by agent (during the *Interrogator* role – here is also the link between these 2 roles: *Interrogator* and *Reviewer*). ´

In the case of the *Monitor* role the agent needs the skill to create a student profile [Mar05] from the students' interaction. The student profile includes student's goals, plans, capabilities, attitudes and knowledge. This profile is based on student's activity during a learning session.

Let's consider a trait called *Activity_Level* which specifies level of a student activity during a session, as a numerical integer value from the interval [-5, 5]. -5 means that the student is very lazy or he or she is not interested in participating in the common learning, 0 that our student is neither lazy nor energetic, and 5 to define that the student is very active, energetic. The default value is 0 and it is assigned by the agent to each student at the beginning of a new learning session. This value can be incremented or decremented based on student's behaviour during the learning process in the following manner: if the student has the initiative and performs successfully a learning assignment his activity level is increased by one, if he/she performs the learning task guided by agent without being able to make own decision the activity level is not modified. The activity level is decreased only when the student refuses to perform its task. Of course the most active student within a learning session can be considered as a leader for the group. In case that the final value of this trait is 0 this can characterize either a lazy or a cautious learner.

The tutor-agent has the responsibility to deliver after each session a report with the students' profile and its own beliefs and conclusions to a human tutor. This profile is based on several parameters like Activity_Level and the human teacher is capable to deduct student's behaviour during a session. Let's assume that after a successful session the levels of activity are 5, 0, -2 for Jan, Mary and Robert. The human teacher can deduct the following things:

Assessing collaboration: it is obvious that Jan dominated all the phases of the activity, Mary did everything under tutor supervision and Robert didn't want to participate.

Assessing contribution: Jan did almost everything while Robert did nothing.

Based on the activity level agent can regulate its future teaching strategies and also create a report with its own mental beliefs and actions during a learning session:

Selecting tasks(subtasks) that need focus, a more detailed analysis or explanation: in case the students reached a deadlock the agent should be able to provide more hints, detailed theoretical demonstrations or other similar examples which can help students to overcome the current situation.

Assessing its own interventions: agent should be able to decide whether to intervene or not (students must learn to work in a team and as long as they make progresses do not intervene) and choose the right time when to intervene in the learning process.

Planning the group structure: decide based on the students profile whether the group structure is the optimal one or not. In our example Jan is too active for his group maybe moving him into another group and bringing someone else into his place can motivate also Mary and Roberts to involve more in the learning process.

5.1.3. Model of Follow-up of a Collaborative Activity

The model of follow-up of learners engaged in a distance collective activity lists the set of the tasks which the tutor is able to realize in order to perceive the individual work, the collective work, and the group dynamics.

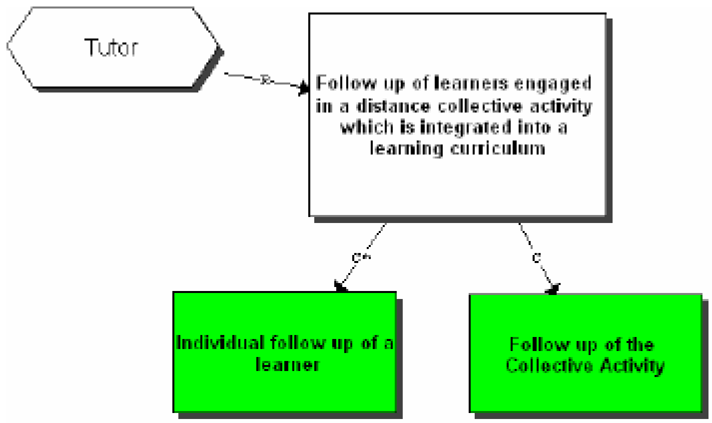

Figure 5-3 Decomposition of the follow-up of learners engaged in a collective activity

The model also lists the set of computer resources which is able to bring information to the tutor (discussion, individual and collective productions, social behaviours, sociometric analysis, and information about the individual curriculum of the learner).We propose to divide this follow-up into an individual follow-up of the learner and a follow-up of the Collective Activity (see Fig. 5-3).

The interest of such a model in our design method, is that it will guide us in the design of the tutor's tools, on the level of the type of information to be presented, but also on the level of the type of the various possible combinations between this information (for example, a tool could combine the visualization of a collective production, the associated collective discussion, and the individual productions of the preceding phase).

A. Individual follow-up of a learner

The individual follow-up of a learner is composed of four tasks namely: the consultation of the learner's curriculum, the consultation of the activity in an individual task, the consultation of the learner's activity in a collective task, and the intervention with the learner.

The task of consultation of the activity in an individual task allows to visualize the individual productions and to consult the messages sent by the learner (to the tutor, or to learners of the group).

The task of consultation of the learner's activity in a collective task is divided into four tasks of visualization. The first is a task of visualization of the collective discussions from an individual point of view: description of the learner's interventions during the discussion. The second is a task of visualization of the learner's contribution in the collective productions: this is all the more possible as the production is cooperative, because the learner's contribution results naturally from the collective production. The third is a task of visualization of the representations of the social behaviour, automatically calculated from the collective discussions. The last is a task of visualization of the sociometric analysis of the group [Lap04]. The sociometric analysis of a group is the combination of the points of view of each member about all the other members; this analysis is obtained from learners' answers to a specific questionnaire. During most of his tasks of consultation of the learner's activity in the collective task, the tutor can take notes about the learner.

The task of intervention with the learner concerns two types of intervention: intervention on the learner's role in the group and intervention on the learner's activity in an individual and/or a collective task. To act, the tutor can read the notes he took about the learner during his tasks of consultation of the learner's activity in an individual and in a collective task. For example, the tutor can intervene by sending messages to the learner.

B. Follow-up of the Collective Activity

The follow-up of the Collective Activity (see Figure 5-4) is carried out through the visualization of the collective discussions from a group point of view, through the visualization of the group dynamics and through the consultation of the collective productions

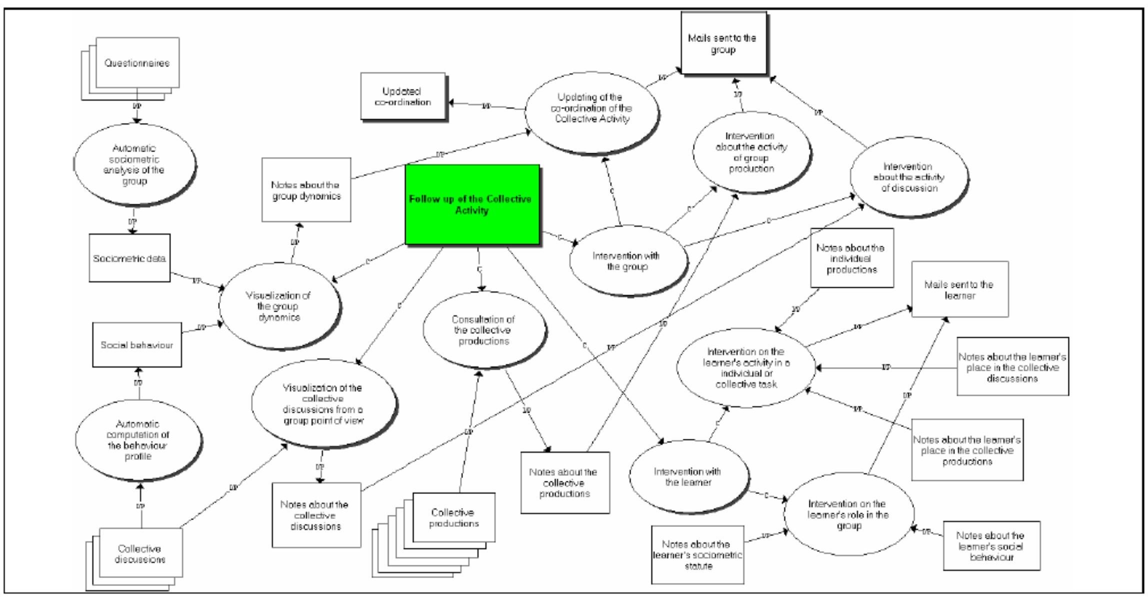

Figure 5-4 Follow up of Collective Activity [Lap04]

The tutor can react to the group. The task of intervention with the group is divided into a task of intervention relating to group production activity, a task of intervention relating to discussion activity, a task of updating of the coordination of the Collective Activity. So, the tutor can act with the group on the three spaces of the functional clover (see Figure 5-5) of collaborative learning environments [Ell94]: the discussion space, the production space and the coordination space. To intervene with the group, the tutor can send messages to the group.

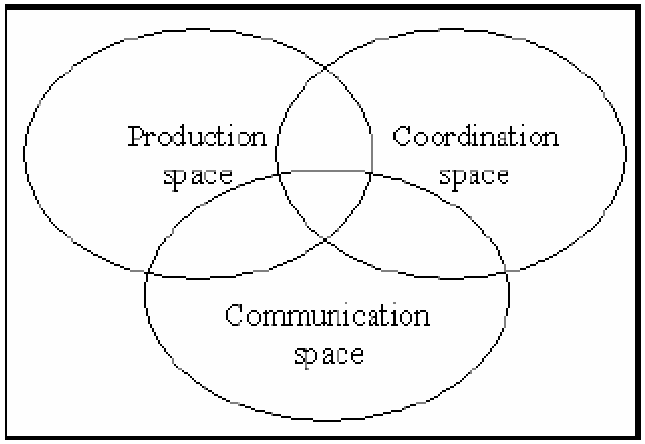

Figure 5-5 Functional clover of collaborative learning environments [Ell94]

Moreover, the tutor can act with the learner. The task of intervention with the learner is the same as the task of intervention of the individual follow-up of the learner. In order to supply this model with information, we present our study about various scenarios of collective activities found in the literature and we present our model of distance collective activity which results from this study and from our model of follow-up.

The above model makes it possible for the agent to initiate interactions with the learner so as to maximize positive effect on the learner and minimize negative effects. These include the following:

- Proactively offering assistance when the learner is focusing on a particular task but is failing to make progress on it, and
- Offering assistance when the learner has failed to complete a task and has moved on to another task.

However, a complete solution to the problem of deciding when to intervene with a learner depends upon a number of additional factors:

- The immediate history of past learner performance,
- The learner's individual characteristics (e.g., whether or not they prefer to work on other own),
- Motivational state (e.g., self-confidence),
- Affective state (e.g., is the learner confused or frustrated),
- The degree of disruptiveness of the offered help (e.g., does the agent's comment require an explicit response from the learner), as well as
- The relationship that the agent has established with the learner (e.g., does the learner trust the agent's advice).

Access to this information can permit the agent to be more selective in choosing when to provide feedback, e.g., provide more confirmatory feedback to learners who lack self-confidence. Some of these factors can be derived through further analysis of the learner's activities, as it will be later in this chapter described.

5.1.4. Modeling Tutoring Agents using AORML

In this section we introduce a new modelling paradigm: the Agent-Object-Relationship (AOR) metamodel [Wag03] for modelling agent-oriented information systems. As in Entity relationship (ER) modeling, the purpose is to provide a generic methodology for information systems analysis and design. In the same way as an ER model can be effectively transformed into a relational or object-relational database schema, an AOR model should be transformable into a corresponding database schema. Notice that this implies that the elements of the AOR metamodel must have a formal semantics.

ER modelling does not account for the dynamic aspects of information and knowledge processing systems. These aspects are related to notions like communication, interaction, events, activities and processes. For capturing semantic aspects related to the dynamics of information systems, it is necessary to distinguish between agents and passive objects. While both objects and agents are represented in the system, only agents interact with it, and the possible interactions may have to be represented in the system as well.

The UML does not support the concept of an agent as a first class citizen. In the UML, there is a certain ambiguity with respect to the agent concept. Human and artificial agents, if they are 'users' of a system, are called actors being involved in use cases but remaining external to the system model, while software agents within the boundaries of the system considered are called 'active objects'. In the UML, the customers and the employees of a company would have to be modelled as 'objects' in the same way as rental cars and bank accounts, while in the AOR approach they would be modelled as institutional or human agents to be represented in the system of that company (which itself could be modelled as an artificial agent).

Since interaction between agents takes place in a social context, deontic concepts such as commitments and claims with respect to external agents, and rights and duties with respect to internal agents, are essential for understanding and controlling coherent interaction between agents and other systems. Neither ER modelling nor UML provide any means to account for the deontic aspects of an information system.

In AOR modelling, an entity is either an event, an action, a claim, a commitment, an agent, or an object. Only agents can communicate, perceive, act, make commitments and satisfy claims. Objects do not communicate, cannot perceive anything, are unable to act, and do not have any commitments or claims. Being entities, agents and objects of the same type share a number of attributes representing their properties or characteristics. So, in AOR modelling, there are the same notions as in ER modelling (such as entity types, relationship types, attributes, etc.).

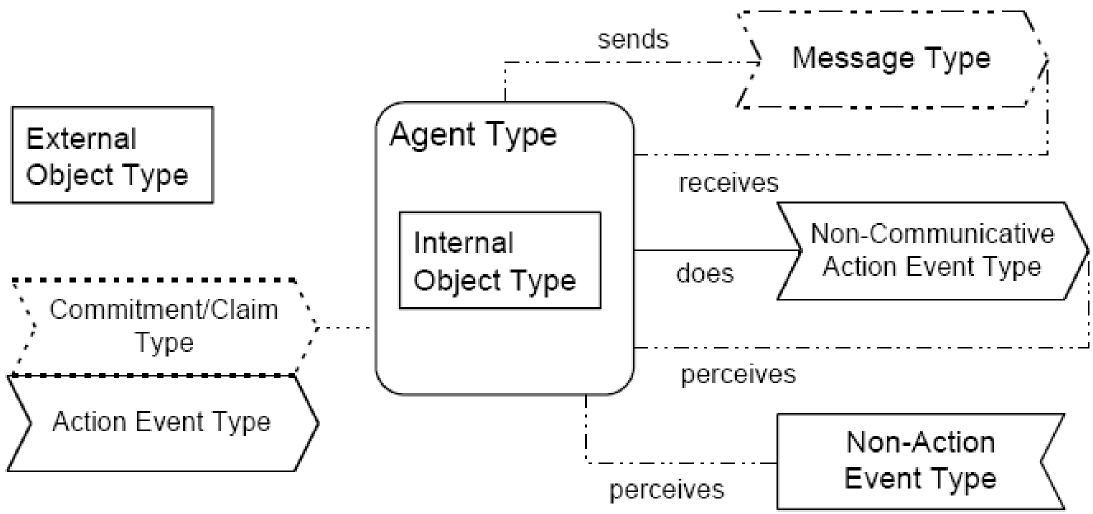

Figure 5-6 The core elements of AOR modelling

The AOR modelling language (AORML) is based on the AOR metamodel. While ER modelling and UML support the design of object-oriented information systems realized with the help of relational and object-relational database technology, AORML is to support the high-level design of agent-oriented information systems. For a more detailed overview of AORML the reader is kindly refereed to [Wag03].

For showing how to model pedagogical agents with AORML let us assume the following scenario: during a learning session a student needs help on a certain topic. There are two possibilities:

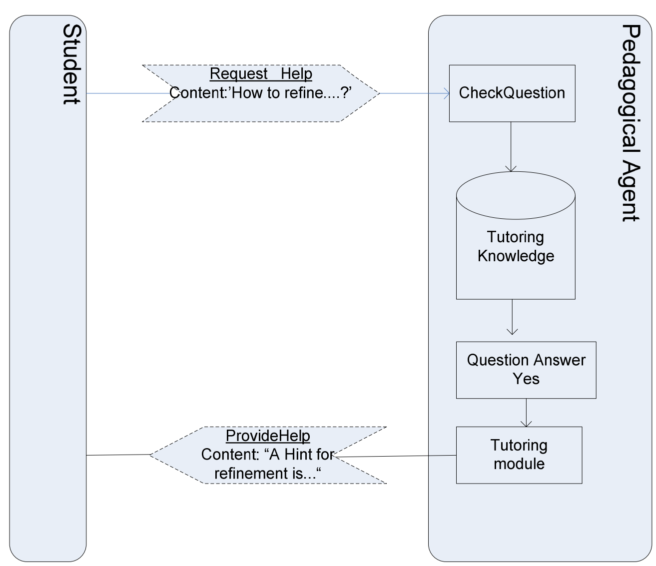

Figure 5-7 Scenario Solution A

A.) as shown in figure above the pedagogical agent can decide using the tutoring module about the way of providing help or hints to the student

B) the pedagogical agent doesn't posses enough knowledge to answer the question, therefore it "claims" an answer to the mobile agent. Mobile agent is responsible for pedagogical-agent's '*knowledge needs*[1]' when it needs to communicate with other tutors (humans or agents) in order to receive help for accomplishing its task. In our case we assume that the mobile agent can find another 'Peer' Tutoring Agent which can provide help.

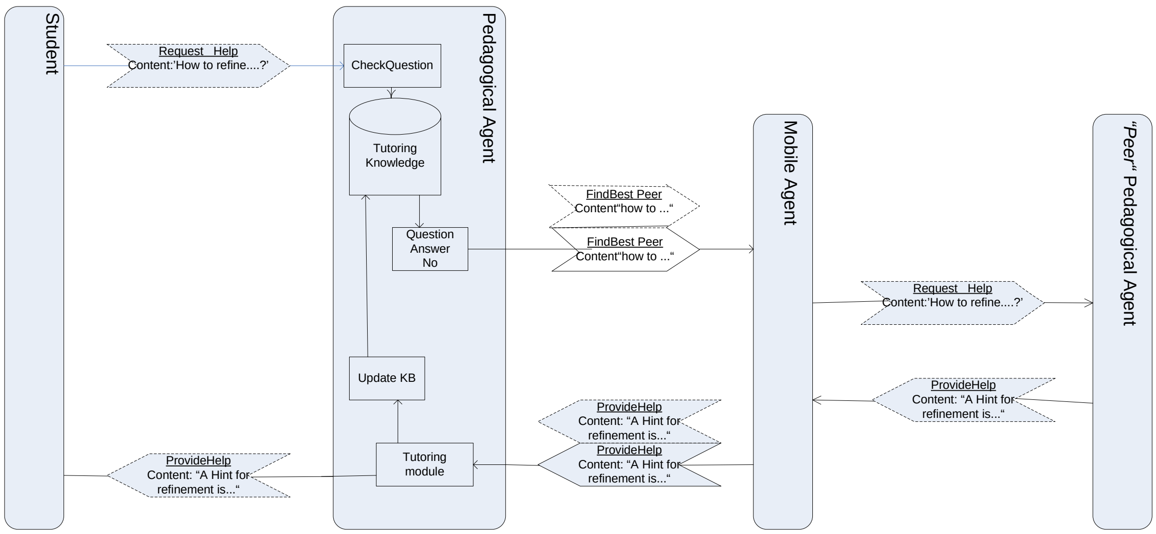

Figure 5-8 Scenario Solution B

One can easily notice that our Agent based learning systems needs to incorporate two more agent types:

i. the *mobile agent* type: responsible for providing help for pedagogical agents' 'knowledge needs'

ii. the *coordinator agent* type: it exists on the server-side, responsible for coordinating all pedagogical (there can be more than 2 pedagogical agents due to the simultaneous learning sessions) and mobile agents working in reasonable orders in case of chaos. Its work is to receive incoming mobile agents, supply to mobile agents the host list in the network, and send them to their destinations. Coordinator agents are always watching the status of the whole network, and they can offer the best path for mobile agents' migrating. Also it is responsible for starting a new tutor-agent for every new session started.

Nevertheless, this work does not focus on modelling these two types of agents.

---

[1] Here knowledge need denote the fact that pedagogical agent lacks the proper knowledge to fulfil successfully its tutoring goal

5.2. Analyzing the Social Model of Tutors

Tutoring situations can be characterized as a social event, the goal of which is for a student to learn some task or acquire knowledge with the tutor acting in all kinds of ways to assist the student with this goal. As with about all actions we carry out, our emotional state plays an important part in the selection of action and the evaluation of the result and in turn the actions we carry out and their results have an impact on how we feel. Bales [Bal50] systematically observed groups in laboratories and found that a substantial proportion of group interaction is devoted to the socio-emotional issues of expressing affect and dealing with tension. The actions of the tutor are also not just restricted to pure instructions but they should also create the right emotional conditions for a student to act. The fact that the tutoring situation is a social encounter means that influencing the emotional state proceeds through social acts with emotion changing potential. For instance, the tutor has the status to judge and criticize (or praise) the student for his actions. Other interpersonal actions that give rise to affect appraisals are defining a task (consider the difference between the psychological effect of formulating this as an order or as a suggestion). The tutor has to steer and motivate the student, know when the student welcomes a hint, etcetera. The emotional state related to this form of social interaction typically involves elements and variables such as: social rewards, dependence, status, power, face. In general one of the goals that people want to come out of social interaction is to enhance the self of each actor. The ideal outcome is that the student is proud of his achievements and feels highly estimated by the tutor.

Motivation is one of the key terms used in education. The emotional state of the student contributes a lot to whether a student is motivated or challenged, which are key conditions for certain actions. Curiosity and puzzlement may lead to investigate problems. But also frustration may lead to action, although it is a more negative affect. The tutor can choose to consider taking certain actions to bring about a change in the emotional state. Lepper [Lep93] identified four main goals in motivating learners: challenge, confidence, curiosity and control. Some ways to implement these tactics are the following:

- The student can be challenged by selecting appropriately difficult tasks, or by having the difficulty emphasized or by having some kind of competition set up.

- Confidence can be boosted by maximizing success directly (praising) or indirectly (‘it was a difficult task, you managed to do’).

- Curiosity is typically raised in Socratic methods.

- The tutor can leave the initiative to the student or offer options that suggest the student can make choices and thereby influence the student’s feeling of being in control.

By observing a teacher in the classroom one can see that many factors are involved in the teaching/learning process. In the cognitive exploitation of the knowledge domain, the perception of the facts and the decision taking process are surrounded by “non rational" aspects involving students and teachers. To transport the non rational behaviour of the teacher to the ITS, it is necessary to look at motivational and emotional aspects to apply in the context, mainly in regards to the accommodation to the needs of the student [Vic98b].

Motivation in ITSs requires communication channels in order to allow the tutoring system to get some awareness of the student's emotional and motivational state. The identification and exploitation of such channels are not the goals of this paper. Previous research shows the viability of affective diagnoses based on cognitive evaluation [Sol95] [Vic98b]. In that research De Vicent based the cognitive evaluation on indicators such as response delay, answer correctness, frequency of demand for help and objective questions about the student affective state. The first three indicators above are the starting points for defining the emotional and motivational structure of this work [Vic98b]. However, these indicators are considered here with slight modifications, as described forward. In the future other channels, such as Web cameras or microphones attached to the computer may be used to collect information to the ITS, complementing the emotional and motivational database. For now the information obtained from the indicators above mentioned will be stored in the database to support the ITS inferences. The definition of the relevant information, the inference methods, and their insertion into the overlay expert architecture are the focuses of this research.

The ITSs' cognitive scheme is expected to deal with the student modelling of the subject domain: the right answers, the mistakes, the performance, the mapping of the learned topics, the sequence of steps, the demands for help, and so on. This composes the student learning history, one of the basis for the cognitive evaluation.

The primitive variables represent the series of student acts in a period of time during the interaction[2] with the tutoring system. They will be stored and consolidated considering three different issues. The first issue is related to the student. As the student interacts with the tutor, information about response delay, error rates, help demand and chronological sequence of answers are modelled and consolidated in the emotional student model.

The second issue is about the relationship between the emotional student model and the knowledge domain. As an example, the student can show an increasing rate of demand for help in a particular curriculum item of the knowledge domain because of the lack of information that should precede that item. In this case, the behaviour does not indicate any alteration in the emotional model. However, if this is a recurrent behaviour, the tutor should consider the re-evaluation of the emotional model.

[2] In this context, interaction is the period of time between the beginning and the end of a learning activity. Many actions can occur and be caught by the communication channels during an interaction. An action can be, for instance, a help demand, a click on the screen, or an answer to a posed question.

The last issue is related to the teaching/learning activity[3] itself. In its definition, through a specific authoring tool [Sol95], the expert will provide the expected response time, the activity complexity and its category (preparation, fixation, repetition or recreation).

All these primitive variables are considered as the basis for inferring affective factors that determine particular behaviours. For instance, the lack of persistence in a task can be modelled after the frequency of demand for help. According to Viccari and Giraffa [Vic98], the effort modelled after the student's progress is a relatively reliable indicator of motivation. The behaviour of a student who asks for help without trying to solve a task may indicate lack of confidence. The factors that should be identified through the student interaction with the tutor are effort, confidence and independence. They directly influence motivation [Sol95], [Vic98b].

The effort is defined as the persistence in solving an activity to acquire a particular knowledge, or as the energy spent to accomplish the task. The confidence is the sense of control and the domain of the environment, as demonstrated by a gradual and continuous knowledge acquisition. The independence is the ability to execute the task and to acquire knowledge without asking for help in excess. It is clear that this set of behaviours does not encompass the wholeness of the human being. The intention here is to model and implement a structure to represent emotion and motivation for widening ITSs' horizons and supporting teaching and learning in a more individualised basis. In the future this work may be extended to a more generic structure where the expert could, for instance, tell which behaviours should be modelled and which actions should be related to primitive variables to indicate these behaviours.

The student behaviour is initially defined based on momentary primitive variables. With the persistence of the behaviour, the student's temperament is deduced. According to Goleman [Gol95], the temperament can be defined in terms of the moods that typify our emotional life. It is a set of innate individual characteristics. For instance, low effort and lack of confidence may imply in shy temperament. Goleman worked with a model of four types of temperament - shy, daring, optimistic and melancholic – and each one is due to a different pattern of brain activity. Goleman defined behaviours that are related to these temperaments. This knowledge will be transported to the ITS area to define the temperament of the student, providing the needed information to establish an individualised environment.

[3] The learning activities can be *lessons, examples, exercises* and *tests,* and are related to the knowledge domain. They represent an interaction space with goals, timing and characteristics well defined by the expert teacher

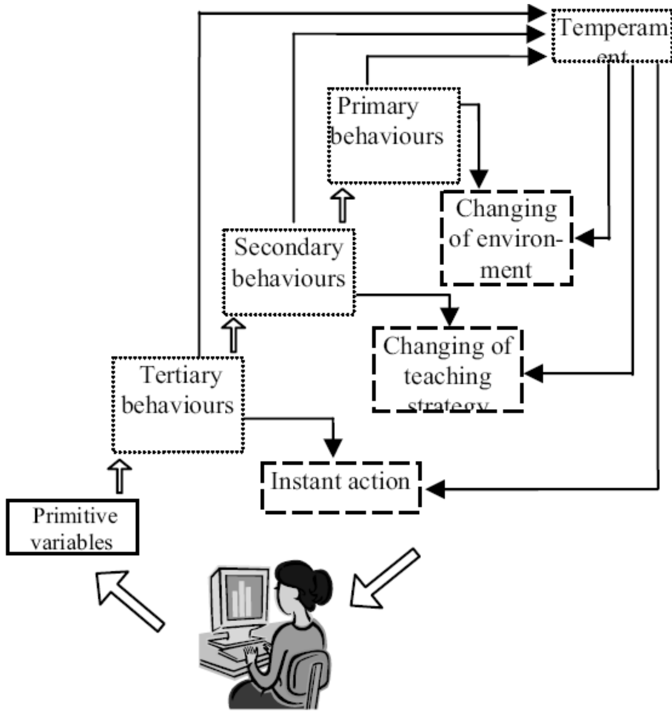

Figure 5-9 Emotional Structure [Gol95]

The *daring* temperament is characterised by the eagerness for exploiting the unknown, by the easiness to transpose new barriers with little emotional perturbation. In opposition to this, there is the *shy* temperament. The *optimistic* temperament in this context is connected to energy and good vibration, as opposed to the *melancholic* temperament.

Temperament is not destiny [Gol95]. Does our biology determine our destiny? Can a shy child become a confident adult? If a particular emotional set can be changed after the experience, when and how should the tutoring system change its expectations about the temperament of a student? To answer these questions we propose a scale of gradual inferences relating the current behaviour and the temperament, and the definition of the changes to be held during the tutoring process.

The Emotional Structure, as shown in Figure 5-9, is composed by a series of information that are obtained through the communication channels (the primitive variables) and other inferred information (in the dotted boxes). These are the basis for the tutoring system decisions.

The primitive variables are captured by the tutoring system and compared with the expected values, as specified by the expert teacher to each learning/teaching activity. The variables are stored in three temporal spaces: the current value, the average value and the gradient, which represents the primitive tendency, whether it is decreasing (value –1), increasing (value +1) or it is stable (value 0).

The primitive variables are used to infer the behaviour patterns – effort, confidence and independence. These patterns may indicate the behavioural tendencies as time progresses. The model illustrated in figure 1 is intended to organise the behavioural chronology in a scale to graduate the behaviours. There, the tertiary behaviours influence the secondary behaviours, which eventually influence the primary behaviours. They are composed by the same sort of patterns, but are related to the persistence of their occurrences. The behaviour patterns determine the temperament *(shy, daring, optimistic* and *melancholic).*

The tertiary behaviours are indicators for the tutoring system to act instantly. For instance, it may decide to start a motivational agent, which can be a process represented by a wizard or other character to "provoke" (telling a joke, for example) or prompt the student for some immediate action. The secondary behaviours define the necessity for changing the teaching strategy. They will be trigged after some persistence of a tertiary behaviour, and will cause a deeper customisation of the tutoring process, once the *immediate* acts seem to be ineffective. The requirement of some persistence of the behaviour to start a bigger change is intended to balance the tutor behaviour avoiding considering drastic changes of the student temperament without good reasons. Finally, a changing in the primary behaviour imply in a major change in the tutoring system environment. For instance, the tutor may decide to keep a help agent permanently present to act as a companion for a child who shows long term low confidence.

The tertiary, secondary and primary behaviours determine what the tutoring system has to change to keep the student motivated. The temperament, on its turn, defines how the tutor should carry on the changing. As an example, the tutor may not propose daring learning activities when the child shows shy temperament.

The aim of the presented research is to build a '*believable*' pedagogical agent capable of communicative and expressive behaviour. To be believable an agent should posses '*social intelligence'* which allows it to act consistently with its goals, its state of mind and its personality. Social intelligence allows the agent should be able to express its emotions but also to refrain from expressing them: a reflexive and not impulsive agent. A reflexive agent means an agent who "thinks" it over before displaying one's emotions, that is one who, when feeling an emotion, "decides" not to display it immediately. Nevertheless, one can easily notice several issues when building agents capable of feeling emotions: what could it do? How could it behave? Would it express its emotion to a potential interlocutor, or would it ruminate on its own emotion by itself, without showing its feeling? Nevertheless, the answers to these questions require conducting an analysis which highlights a broad range of intelligences that humans use in the construction of their everyday social life and how these can be effectively triggered by artificial agents.

5.2.1. Introducing Social Intelligence

Nowadays, more and more applications require systems that can interact with humans. *Socially Intelligent Agents* (SIAs) are agent systems that are able to connect and interface with humans, i.e. robotic or computational systems that show aspects of human-style social intelligence [Dau01]. In addition to their relevance in application areas such as e-commerce and entertainment, building artefacts in software and hardware has been recognized as a powerful tool for establishing a science of social minds [Whi95] which is a constructive approach toward understanding social intelligence in humans and other animals.

Social intelligence in humans and other animals has a number of fascinating facets and implications for the design of SIAs. Human beings are biological agents that are embodied members of a social environment and are autobiographic agents who have a unique personality. They are situated in time and space and interpret new experiences based on reconstructions of previous experiences. Due to their physical embodiment, they have a unique perspective on the world and a unique history: an autobiography [Eis86]. Also, humans are able to express and recognize emotions, that are important in regulating individual survival and problem-solving as well as social interactions.

Like artificial intelligence research trend, SIA research trend can be pursued with different goals in mind. A deep AI approach seeks to simulate real social intelligence and processes. A shallow AI approach, which will be highlighted also within this thesis, aims to create artefacts that are not socially intelligent *per se*, but rather appear socially intelligent to a given user. The shallow approach does not seek to create social intelligence unless it is meaningful social intelligence vis-à-vis some user situation [Sen98], [Ree96].

In order to develop believable SIAs we do not have to know how beliefs-desires and intentions actually relate to each other in the real minds of the people. If one wants to create the impression of an artificial social agent driven by beliefs and desires, it is enough to draw on investigations on how people with different cultural background, develop and use theories of mind to understand the behaviours of others. Therefore, SIA technology needs to model the folk-theory reasoning rather than the real thing. To a shallow AI approach, a model of mind based on folk-psychology is as valid as one based on cognitive theory.

The shallow AI approach adheres to a constructivist notion of humans, reality and meaning. Humans' experience of reality is never raw or direct but constructed in the interaction between reality and an active and richly equipped observer. Humans have perceptual dispositions, knowledge, common notions, schemas, cultural models and patterns, prejudices, reasoning capabilities, which are constantly employed in the interaction with physical and socio-cultural reality [Lak99], [Joh87].

One side of social intelligence deals with the agent's behaviour in relation to the situation and itself. For example, one would expect a social intelligent agent to display joy when it gains something it considered valuable, or when it achieves its goal. However, in order to appear socially intelligent on a deeper level, the agent must be able to take into consideration other social agents in the environment. A fundamental social ability in humans is empathy, which can be seen as a requirement for a social intelligent agent. Empathy can be defined as the ability of taking another person's perspective and trying to understand the mental lives of others, on a perceptual, cognitive and emotional level. On the perceptual level empathy refers to the ability to detect where other people are looking or being able to imagine the perspective of another person. On the cognitive level empathy involves the ability to infer and reconstruct the intentions, thoughts and the feelings of other people. On the affective level empathy includes the ability to not only understand that someone has a certain mental state but also to share that state, or a congruent one.

5.2.2. Understanding Social Intelligence

When humans communicate, they employ a variety of signals in combination with verbal utterances, such as body posture, gestures, facial expressions, and gaze. In a similar way, SIA may use their bodies to convey meaning and regulate communication. Recent years have witnessed a research trend to present a broad range of intelligences that humans use in the construction of their everyday social reality and how these can be triggered by SIA. The most extensive study of non-verbal behaviours for synthetic characters, especially gestures, can be found in Cassell's work on embodied conversational agents [Cas00]. Persson [Per01] argues that these "real" social intelligences are not bits and pieces, rather coherent structures of "knowledge," with which people reason and infer meaningfulness. These intelligences constitute a set of expectations, norms, or rules by which to judge social reality. If reality fails to meet those expectations, the meaningfulness of social reality will be in danger (categorizing the behaviour of others as "crazy," "strange," or in some way "other"). Furthermore, he outlines the levels of social intelligence in some order of abstraction beginning with the most basic:

1. *Expectations on Visual Appearance and Behaviour:* One way in which people understand social intelligence is through the sheer surface behaviour of other objects and creatures. These expectations belong to four broad categories:

    a. **Agency versus nonagency**: On the most basic level, people have structured expectations on how to differentiate between intentional and non intentional objects (see [Den87] and section 2.1.2) People use specific behavioural cues in order to make such fundamental categorizations, often in a non conscious manner. If the objects are classified as intentional, then all other social expectations come into play; if not, then the object is treated as a "dead thing." This distinction is deep-seated in human nature, and affects the behaviour toward the object in question. If we want to encourage users to categorize our SIAs as social agents, the systems have to exploit these processes in users.

    b. **Gestures**: In human interaction, facial and body gestures are important modulators of communication. People have a body, which is movable in certain configurations, around which mankind and cultures have developed codes, norms, and conventions that bring meaningfulness to those gestures. Face, body, and hands are important channels for such nonverbal communication, and quite a few SIAs are equipped with the possibility of such gestures. Emblematic gestures are culturally specified gestures, e.g., signalling "okay" by a "thumb-and-index-finger" ring gesture. An example of a propositional gesture is the use of both hands to measure the size of an object in symbolic space while saying "there is a big difference" (see Figure 5-10-c).There are four types of gestures that support the conveyance of communicative intent (so-called 'co-verbal' gestures [Cass00]): (i) *Iconic gestures* illustrate some feature of an object or action, e.g., mimicking to hold a phone while saying that someone has been called; (ii*) Metaphoric gestures* represent a concept without physical form, e.g., a rolling hand gesture while saying "let's go on now"; (iii) *Deictic gestures* locate physical space relative to the speaker, e.g., by pointing to an object; (iv) *Beat gestures* are small baton-like movements to emphasize speech. A special form of a beat gesture is the contrastive gesture that depicts a 'on the one hand ... on the other hand' relationship if two items are being contrasted (see Fig 5-10.-c). An important class of gestures (including facial gestures) serves the expression of an agent's emotional state such as 'put a hand to its head' to signal thinking (see Fig 5-10.-a). Although face may express emotions most succinctly [Ekm69], we rather rely on signals involving the whole body as the size of the characters used is relatively small. Gestures also realize communicative functions including conversation initiation, turn taking, back channelling ("nodding" see Fig 5-10-b) and breaking away from conversation [Cass00]. The communicative behaviour corresponding to the function of "giving turn" is typically realized by looking at the interlocutor with raised eyebrows, followed by silence, whereas "taking turn" is signalled by glancing away and starting to talk (see Fig 5-10.-d). In social interaction, we expect people to make use of such functions in their gestured and conversational behaviour.

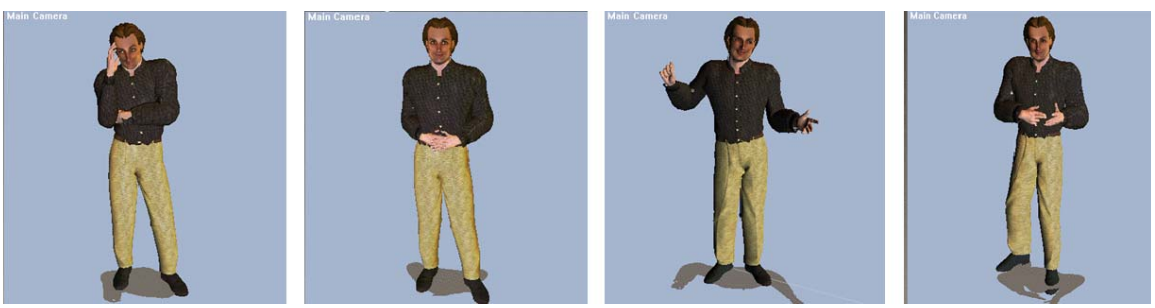

Figure 5-10 Gestures for: a) "thinking", b) "wait to take turn", c) "big difference", d) "explaining"

This does not mean that we expect all of the functions to take place all the time. Some people are more agitated than others, and we still have a basic understanding of the communication and discourse flow. However, the more types of gestures employed, the more expressive and perhaps even socially intelligent one appears, and if too few of the functions above are in place, there is a risk of conversational breakdown ("Is it my turn to speak?" "Does she understand what I am saying?", "Why on earth is he smiling now?"). As these different functions illustrate, the problem of generating gestures is not so much concerned with producing those gestures on a graphical level, but rather to know *when* to produce them in order for them to appear socially intelligent. Few gestures are understood to have a clear meaning irrespective of a situation or context. Such a context may consist of other gestures taking place synchronously, the conversational context, the assumed mental states and intentions of the individual, the traits and social status of the person, or assumed culture-specific codes and conventions.

c. **Gazing behaviour:** One crucial parameter of nonverbal behaviour is gazing. Gazing behaviour is an important aspect of social interaction between humans. Through face gazing, eye gazing, eye contact, gaze avoidance and staring, we regulate space and conversation, establish social relations, refer to things, or convey thoughts, intentions, mental states, personality traits, or social status to others. With gazing, we express attentiveness, attraction to others, intimacy, credibility, and dominance/submission. Through gazing behaviour, we execute control, threats, and deception (see [Kle86] for overview of functions). As with gestures, the meaningfulness of gazing cues is never established on the basis of the signals alone, but always in consideration to the context in which they occur. Such context may be the personal relationship between gazers, the conversational history, the personality of the gazers, the place in which the gazing occurs or the social codes of gazing in a given culture. If SIAs are to gaze in a socially intelligent way, all of these parameters, more or less, have to be taken into consideration. This is indeed a formidable task. There are, however, some low-level aspects of gazing that have attracted some attention among SIA engineers. First, we expect other people to know the difference between mutual gaze and deictic gaze, i.e., between "looking-at-me" and "looking-not-at-me." Eyes are socially more salient than other parts of the body (or the environment) since mutual gaze means acknowledgment and mutual awareness between two agents. If SIAs are to appear socially intelligent, the least thing it should be able to do is to judge the user's eyes as a salient feature of the user. It should understand that the user's eyes are more important than other parts of her body and that looking at her is something different than looking at her shoes or at a table. Besides directing the eyes toward other people's eyes, we also expect other people to infer when others are looking at *them.* This emphatic ability takes into consideration the optical perspective of other people. Secondly, humans (and to some extent animals—see [Ris91]) expect other people to use gazing as *deictic markers*, signalling objects in space by directing eyes (or fingers) to those objects. With the coloured iris on a white eye-globe, the eye is constructed as to support other people's estimations of direction of gaze, even at quite long distances from the person gazing. We expect socially intelligent creatures to be equipped with such directionally rich eyes, and to use them to point out objects in the environment (often in synchronization with discourse). On the emphatic side, we also expect socially intelligent agents not only to display deictic gaze but also understand deictic gazing in *others.* When looking at something in the environment, we expect attentive people to detect this and possibly also direct their gaze toward the object in question. Such social behaviour creates a *shared attention* or a *shared visual space* between two interlocutors [Bar95]. Once such a shared mental space has been established, the two persons can use that shared object in their communication, which requires less explicit information to be conveyed.

Visual appearance, gestures and gazing behaviour are all surface oriented behaviours. However, as was pointed out above, when people try to ascribe meaning to such behaviour, it is always placed in some broader context or conceptual framework. In the next sections, we describe a number of more socio-cognitive structures that relate to abstract patterns of behaviour. These are often described in terms of *social schemas [Aug95], cultural models [Hol87], conceptual frameworks,* or *folk-theories [Lak99]*. Such mental models are structured and coherent and thereby enabling (naive) reasoning about social reality. In order to appear socially intelligent, SIAs need to behave more or less according to those models, as well as possessing the ability to reason with them in relation to the behaviour of other people.

2. *Folk-Psychology:* presents a rich and complex structure enabling sophisticated reasoning about behaviour of others. Folk psychology, in our definition of the term, is a coherent and well-structured everyday model about mental states such as intentions, goals, beliefs, emotions, and desires. This folk-theory of the psyche involves terms and states (e.g., "thinking," "believes," "is angry," "is sad,") how these terms relate to each other (e.g., that beliefs and desires to a certain degree determine intentions and actions—so called *belief–desire reasoning*)—and how to attribute those mental terms to people on the basis of their behaviour but also on contextual features of the situation. When people are trying to understand the behaviours of others, they often use the framework of folk-psychology. Moreover, people expect others to act according to it. If a person's behaviour blatantly falls out of this framework, the person would probably be judged "other" in some, e.g., children, "crazies," "psychopaths," and "foreigners." In order for SIAs to appear socially intelligent, it is important that their behaviour is understandable in term of the folk-psychological framework. People will project these expectations on SIA technology and will try to attribute mental states and processes according to it. Folk-psychology consists of two parts: *mind* and *emotions* (emotions are later described within this chapter). Research on an "everyday theory of mind" has studied conceptions of relations between *perceptions, thinking, beliefs, feelings, desires*, *intentions*, and *sensations,* and how we reason about these [Whi91]. On the other hand, folk-psychology deals with emotions. The ways in which people attribute emotions to other people have been studied within *appraisal theory* [Har93], [Ort88]; for an overview see next section. In all of these cases, the autonomous agents have some model of the world, mind, emotions, and of their present internal state. This does not mean that users automatically infer the "correct" mental state of the agent or attribute the same emotion that the system wants to convey. However, with these background models regulating the agent's behaviour the system will support and encourage the user to employ her faculty of folk-psychology reasoning onto the agent. Hopefully, the models generate consistently enough behaviour to make folk-psychology a framework within which to understand and act upon the interactive characters.

3. *Personality* is a pattern of behavioural, temperamental, emotional, and mental traits that distinguish people from one another. Personality has been studied by psychologists interested in the behaviour and in mental processes [Plu80]. In everyday discourse *personality* refers to dimensions of a person that are assumed to be more stable and enduring than folk-psychological states. Whereas beliefs and emotions change quickly across situations, personality is thought to be quite stable. Whereas emotions and states of mind are often related to the intentions of the person, personality aspects are not. In this everyday understanding, personality refers to those aspects of a person that makes her different from other individuals.

   This folk-theory of personality involves a number of different social schemas:

- personality trait schemas;
- social roles schemas, including: occupancy roles, stereotypes, and archetypes.

a) Trait Schemas: When understanding and explaining behaviour of others, some form of *traits* is often referred to [And87]. We may, for instance, make sense of John's tendency to be late by referring to "his carelessness." Like folk-psychology, trait terms are handy ways of summarizing and abstracting complex chains of behaviours.

Quite a few artificial agents have been modelled on this level. SIA engineers often turn to trait psychology to find inspiration for their models, for instance theories about *The Big Five* [McC92]. As indicated in the beginning of this paper, however, the primary purpose of such modelling must not be to imitate how traits actually works in real people (many psychologists even doubt there are such things as traits). Instead SIA should imitate the ways in which users *believe* traits work in others.

Some psychological research seems to verify the role of traits in reasoning about others and that such mental models about personality are rich and structured. Cantor and Mischel [Can79] presented subjects with four equal length descriptions of persons:

i. a prototypical extrovert;
ii. a prototypical introvert;
iii. two control cases in which neither extrovert nor introvert wordings were used.

The subjects then had to do two things. First, they were to judge the person in each description along six trait scales, one of which included the extroversion/introversion dimension. As expected, the prototypic descriptions were highly rated on the extroversion/introversion scale, whereas the control cases were not. Then the subjects were asked to indicate from a list of 64 randomly presented words which of these they recognized from the character descriptions. Words from the prototypic descriptions were better recognized than words from the non prototypic control cases. In fact, often the subjects would indicate having recognized prototypic words (like "outgoing" or "spirited" for the extrovert case and "quite" and "shy" for the introvert case) although these were *not included in the description.* This tendency was much greater in the prototypic descriptions than in the non prototypic, which suggests that a person schema of extroversion/introversion was used as a basis for the recall, supplying missing features when memory failed. It is this person schema that users will employ in their interaction with real or artificial social agents. We also expect socially intelligent agents to attribute traits to others, and act on the basis of such categorizations. We are not aware of any SIAs that seek to make such attributions within or outside the digital environment.

b) Social Role Schemas: Interacting with other people in an everyday socio-cultural environment often requires expectations about social roles that people will play in a given situation or relation. Understanding a situation in terms of such *social role schemas,* is a fundamental dimension among humans and we should expect people to project such expectations on intelligent systems as well. Psychologists and sociologists have proposed a number of social schemas.

One type of social schema are *occupancy roles [Aug95],* providing us with normative expectations about the daily activities and standard whereabouts of e.g., *doctors, waiters, police officers, scholars, chefs, farmers,* and *bus drivers.* Such role schemas are often part of *event schemas* or *scripts [Sch77],* and may contain expectations on goals, beliefs, emotions, morals, and behaviour of those roles. Occupancy roles lie close to family role schemas, e.g., *mother, father, children, cousins, sisters, siblings, uncles,* or *lovers,* which contain expectations on how such roles interact with each other on a daily basis. In social life, these role schemas fill in information about what to expect of others once we have placed them in one of these categories/roles.

To extent the framework presented so far we add a new level to enhance the characteristics of life-like agents: cultural awareness level. The inclusion of culture in agents is needed in many different applications, including military training, games, like also learning based applications. To be effective in these applications, culture needs to be included in many aspects of embodied agents, from channels of nonverbal communication to verbal communication to decision making and values. In learning based applications, aspects of agent's body language could be strengthen with verbal language. Body language could even be used as feedback for students during the learning process. Facial expresses and gestures could be used to indicate that the agent does or does not agree with what is being communicated, thereby providing a feedback similar to what would be experienced in that culture.

5.2.2.1. Culture

> *"Culture may also be viewed in terms of different weights on goals. Both a Somali shepherd and an Italian housewife have the goal of feeding themselves and their family; but the sub-goal chosen to pursue this goal may be for the former to search the bush, for the latter to go shopping in a store."*
>
> Poggi [Pog00]

Humans pursue their goals by using their internal and external resources. External resources are the objective conditions holding in the environment (presence of food, characteristics of the territory, climate conditions and so forth); internal resources are the human's action capacities (physical strength, body agility, manual skill) and beliefs.

But, in different environments the physical conditions, hence the most easily available resources, and consequently also the actions to get them, are different. In the land it will be easier to get food by rearing sheep or cows, while on the coast fishing will be the most direct way to feed. So, in the land the most useful beliefs to store and process and the most necessary actions to learn will concern sheep and cows rather than fish or shrimps, and people will more easily become shepherds than fishermen. Any population, given the environment in which it lives, comes to accumulate a set of beliefs on the instrumental goals that most easily and economically serve the biological terminal goals in that environment. In other words, an instrumental goal becomes more or less important in a culture depending on the strength and necessity of its link to a terminal goal. At the extent to which that instrumental Goal is (or is very likely to be) the only possible means to reach a terminal Goal, given the external conditions available in that culture, that instrumental Goal will receive a higher weight than other possible ones. Thus, the instrumental goal chosen becomes a strategy of survival typical of that culture; and culture, overall, may be defined as a set of beliefs on the typical techniques to pursue goals.

Of course, the techniques chosen are determined by beliefs on the environment: for example, until a population does not know the mechanisms of plant reproduction, the technique of agriculture cannot be chosen as an instrumental goal to survival. Therefore, culture entails beliefs about the external world. And since language is both produced by beliefs and a vehicle of them, culture typically shows up in language. Language is made of the beliefs of a population, but is also a way to organize them, a set of rules on how to conceptualize and categorize information. Consequently, it implies, again, a set of settled typical communicative techniques, that is of settled instrumental goals stating how to convey information.

In addition to the beliefs about the best techniques to pursue terminal goals, culture entails also values and norms. Values are evaluative beliefs about what is good and then has to be pursued as a goal [Mic89]. But since particular ways to behave may be good or bad according to the environment, again due to what are the most useful techniques of survival, different populations in different environments may hold different values. For example, in the environments where individualistic behaviour has proved to be convenient, individualistic values will develop; on the contrary, in environments where collectivistic behaviour is more fit, values centred on the family or the group will hold.

Norms are obligations that rule the relationships among people in a group [Con95]. Again, in a culture more centred on interdependency, a highly weighted goal, and then a norm that holds, may prescribe to be very cooperative with each other, even when this implies intruding in the other person's affair. On the contrary, in a culture more centred on the individual's autonomy, the goal of keeping one's privacy will be more weighted, and a norm will hold of not intruding in others' affairs and of contrasting others' intrusions.

Now, both values and norms generate goals in people (the goal to pursue that value or to respect that norm); if they are thwarted, they provoke emotions. Not living up to one's values may induce shame, while violating norms may cause a sense of guilt. Therefore, if two populations have different values and norms, they will also feel these emotions as a consequence of different events.

To summarize our definition, we may say that culture is a set of beliefs shared by a population. These beliefs regard the environment in which the population lives and the best techniques (the most highly weighted instrumental goals) to reach the biological terminal goals in that environment, given the means-end relations that hold in the given physical conditions and the set of beliefs accumulated. Culture also includes beliefs about how to gather and organize beliefs themselves and about the norms and values that are functional to techniques of goal achievement that best fit the surrounding environment.

According to this definition, one may try to figure out how the way people communicate changes with cultural differences, by trying to distinguish what is universal (biological) and what is culturally determined in the different aspects of communication. These differences may then be simulated in a Believable Embodied Agent, in words or discourse planning, in gesture, gaze, facial expression, body posture and proxemic behaviour.

Culture-sensitive vs. universal gestures

The issue of universally shared vs. culture specific signals is particularly tricky in gestures, because different types of gestures exist. Among codified gestures, some are culturally codified: for example, the gesture for 'OK', or Churchill's gesture for 'Victory'). Others may be biologically codified: for example, those which are a ritualization of physiological movements, like the gesture of raising fists up to show elation. If we want to simulate gestures of the former kind in a culture-sensitive Agent, they will have to be varied from a culture to another. Creative gestures like the iconics, instead, might all be generated through the same set of inference rules (supposedly universal), whatever the culture the Agent comes from. Of course, at the extent to which the referent represented is typical of a culture or an action is performed in a way typical of it, then also a creative gesture may be culturally dependent.

Culture-sensitive vs. universal gaze and facial expression

Facial expression and gaze are more likely to be universally shared than gesture. They can communicate information on the world (we point at things with chin or gaze, squeeze eyes to say that something is little or difficult), information on the Speaker's beliefs, goals and emotions (we raise eyes while we remember or make inferences; we frown to communicate anger, concentration, or an order) and information on the Speaker's identity (our face and gaze provide information on sex, age, ethnicity, personality, sometimes even social class).

Let us focus our analysis on face and gaze expression of emotions, to discuss whether the feeling of emotions is universal. Affective lexicons do differ across cultures [Rus89]; however, the so-called *basic emotions* (happiness, sadness, anger, fear, surprise and disgust) are felt in all cultures, and everywhere they trigger an innate universal neural program for facial expression [Ekm82]. This does not necessarily imply that people in different cultures always show their emotions in the same way in the same situation. Two cultural factors may intervene in this case, to produce wide differences in emotion expression. First of all, an emotion is triggered by the cognitive categorization of a situation on the part of the subject. So, a situation that in a culture (because of its beliefs, norms and values) is categorized as a cause of sadness, in another culture (with different beliefs, norms and values) might be categorized as a cause of happiness. For example, the death of a beloved person in a catholic group or the death of a martyr in the Islamic culture may be greeted as a cause of joy. Secondly, the filtering of emotion display to decide whether and how the felt emotion should be expressed, includes factors like cognitive and personality traits of the Agent and of the Interlocutor, their relationship and the situation, but also the cultural norms about the expression or non expression of given emotions.

5.2.2.2. Emotion Theory

Providing a definition of emotion can be seen as a further problem due to the fact that different emotion researchers (e.g. psychologists, cognitive scientists, biologists etc.) often use different vocabulary for the same phenomena.

> "*Is there a right definition for emotion? I suspect we must wait for deeper theories about the underlying mechanisms before we can hope to define precisely what kinds of phenomena we are talking about, just as people had to wait for modern physics and chemistry before they could have good definitions for terms like 'water' and 'salt'* ".
>
> -Sloman, [Slo93]

The human emotion process can be viewed as a classic example of an information-processing system primarily geared towards "serving" concerns at all levels of an agent architecture. In this chapter we will provide a broad requirements specification for such an emotion process and, using recent theories from psychology and neurology [Fri86] explain the mechanisms inherent in the different classes of emotional states (*primary*, *secondary*, and *tertiary*) from an information-level design-based perspective.

> *"[M]any concerns consist of representations of states of affairs that evoke pleasant or unpleasant affect. … Affect elicited by objects or events defining such concerns cannot be said to "serve" these concerns; it merely expresses them. Emotions (affects plus some mode of action-readiness change) elicited by such objects do serve these concerns, by involving signals to the action system, and subsequent changes in action readiness."*

– Frijda, *The Emotions* (pages 118-119)[Fri86]

This work tries to define emotions according to the goal concept and belief model of action and social interaction [Con95].Actions in our life are often part of a plan aiming at some goal. Take for example an action of Oetzi, the 5000 B.C. pre-historic man of Similaun, who chooses a stone apt to sharpen well and makes a lance for chasing the wild-pig successfully. The actions of looking for a good stone and to sharpen it are just means for the complex action of chasing. But also chasing is aimed at feeding himself and his group, which in turn aims at the goal of survival. The goals of our everyday plans are not ends in themselves. They all aim at more general goals of biological import that are common to all humans, like the biological goals of survival and reproduction and some subgoals of them, physical well-being, safety, loving and being loved, self realization, image and self-image. These are *terminal* goals, that are ends in themselves and ones to which we assign the highest weights. So much that, if two of them are incompatible (as for instance freedom vs. life itself), giving up one of them is a heavy renunciation. With respect to terminal goals, the goals of our everyday life are *instrumental* goals, in that they directly or indirectly serve our terminal goals. For instance, chasing the wild-pig with a sharpened stone is instrumental to survival: if the lance is not sharp enough and does not hit the wild-pig to death, the wild-pig might aggress and kill Oetzi. Instrumental goals are more or less important to us, depending on the strength of their link with terminal goals. At the extent to which an Instrumental Goal is likely to be the only possible means to reach a Terminal Goal, that Instrumental Goal receives a high weight, just because it inherits its weight from the Terminal Goal it serves.

Emotions are a biological device aimed at monitoring the state of reaching or threatening our most important goals, be they Terminal or Instrumental (see, for instance, [Car80]. Anytime something happens (or the Agent believes it happens) that is likely to produce the achieving or threatening of a highly weighted goal, the biological device of emotion is triggered. From the agent's interpretation of the situation, a complex subjective state originates, generally of a short duration and with different degrees of intensity. This state includes physiological, expressive and motivational aspects. If Oetzi throws his lance but sees it has not run into the wild pig’s heart, fear is triggered since his goal of survival is challenged: physiological reactions are activated (blood flowing away from face to limbs) some of which may show in the perceivable state of his body (pale face, tremors); and the specific goal of escaping, that might serve the terminal goal of survival, is activated.

There is a strong relationship, then, between goals and emotions: goals both cause emotions and are caused by emotions. They cause emotions since, if an important goal is achieved or threatened, an emotion is triggered: emotions are therefore a feedback device that monitors the reaching or threatening of our high-weighted goals. At the same time, emotions activate goals and plans that are functional to re-establishing or preserving the well being of the individual, challenged by the events that produced the emotions. So, fear triggers flight, anger triggers aggression, guilt triggers the goal of helping the harmed person or of escaping sanction [Cas00b].

Emotion triggering vs. emotion display

Emotions may be implied in communication in at least two ways.

1. they may be the very reason that triggers communication: we activate the goal of communicating just because we want to express our emotion;

2. they may intervene during our communication, as a reaction to what our interlocutor is saying, or to some thought suddenly coming to our mind, either related to the ongoing dialogue or not.

In both cases, the triggering of emotion does not necessarily imply that the Agent displays it. There are many reasons why we may refrain from expressing our emotion, and the final (aware or non-aware) decision of displaying it may depend on a number of factors [Pre01], [DeC01]. Some of them concern the very nature of the emotion felt (emotional nature), others the interaction of several contextual (scenario) factors.

1. Emotional nature

    a. *Intensity* (a more intense emotion might be more likely displayed);

    b. *Valence* (it is not the same to display negative or positive emotions);

    c. *Social evaluation* (some emotions, like envy or shame, are subject

       to social sanction: then it is more difficult to express them);

    d. *Addressee* (it is different to express an emotion to the one who

       caused it or to a third person).

2. Scenario Factors

    a. Agent's *Display motive* (displaying or not depends on whether you do it to be helped, consoled, or if you want to demonstrate or teach something);

    b. Agent's *personality* (an impulsive person is generally more keen to displaying than a reflexive or a shy one);

    c. *Interlocutor*'s features (displaying depends on the other's personality, empathy, intelligence...).

    d. Agent - Interlocutor *Role relationship* (whether he has power over you or you over him);

    e. Agent - Interlocutor *Personal Relationship* (you might not display your being worried to someone you love, if you want to protect him);

f. Type of *social interaction* (being in public makes a difference for emotion display).

5.2.2.3. Millon's Personality Theory

We find the Millon's theory[Mil97] as the best approach for us due to the fact that his parameters (bipolarities) are strongly dependent on one's cultural background.

There have been several attempts to model personality within an agent, usually based on theoretical foundations about human personality, such as Big-Five [McC92], among others. In order to include some personality aspects in the proposed model of the agent, the theory of Theodore Millon has been investigated and has been used as basis for our model. The main reasons for choosing this theory to model personality within agents is that all aspects (topics and polarities) of personality can be easily modelled in an agent architecture. Also, as a result of a psychological test developed by Millon, it is possible to quantify some aspects of personalities (polarities of an individual) and this can be easily added to the decision making process of an agent. These features could not be found in other psychology theories.

The theoretical model of personality proposed by Millon takes into account the importance of biological factors such as genetics [Mil97]. These factors are described in one specific matrix for each individual, which represents a fundamental role in the formation of personality [Str99]. This matrix is also composed of environmental perceptions and actions taken during conflict situations. During the development of an individual, personality has influence from biological and psychological factors that interacts in an endless spiral, in which each circle of this spiral is constructed over previous interactions, creating, in this way, new bases for the next interactions.

Millon proposed a measurement to express personality, which is based on the theoretical comprehension of the actions taken to reach the goals that an individual has in his/her life. Also, it takes into account the way to process information received from the environment. In this sense, it has been elaborated a tool to verify a dynamic configuration of interactions, representing three large areas: motivational aims, cognitive styles and inter-personal relations.

The area *Motivational aims* includes three bipolarities, linking the ecological and evolutionist theories. It is based on the conceptual antecessors of theses two theories through three main formulations: the existence of organisms, its adaptation to the environment and the answers provided in this relation. Based on this theoretical model, Millon presented the following polarities: openness versus preservation, modification versus accommodation and individualism versus protection. In these groups of attributes, it can be observed the orientation of an individual in relation to the role of the environment as a motivation of the actions of an individual [Str99].

The cognitive styles can be found in the evolutionist perspective as well as in contribution of authors such as [Alc04]. They are related to the way how individuals are oriented when interacting to the environment. Their main aim is to evaluate the way an individual can process information, along with three main models proposed by Jung (extroversion versus introversion, feeling versus thinking and sensation versus intuition). Based on this and in [Alc04], the bipolarities can be defined as extroversion versus introversion, feeling versus thinking, reflection versus affectivity and systematization versus innovation.

Millon proposed to use the inter-personal component to evaluate the style of the relationship of an individual with the others. All bipolarities of this area are based on a bio-psychosocial theory of reinforcement learning as well as active and passive strategies. The bipolarities are: shyness versus communicability, doubt versus security, discrepancy versus conformity, control versus submission and satisfaction versus dissatisfaction.

In summarizing, according to Millon, personality can be expressed through twelve bipolar attributes. The values of all bipolarities are given as a result of a psychological test, containing 154 yes-or-no questions which are related to the aforementioned areas. In other words, when an individual takes this test, it is provided values for all polarities, calculated through an equation defined by Millon.

Summary: Since both emotions and personality have to do with the relative importance of goals, there is also some link between emotion and personality. Some personality traits may be viewed in terms of the general ‘propensity to feel emotions’ [Plu80]. Piccard [Pic97] calls ‘temperament’ this subset of personality traits, while other authors relate them directly to one of the factors in the ‘Big-Five’ model: for instance, neuroticism [McC92]. These traits imply, in a sense, a lower threshold in emotion feeling [Ort88]. For instance, a ‘shy’ person is keener to feel ‘shame’, especially in front of unknown people. A ‘proud’ person, who attributes a high weight to his goals of self-esteem and autonomy, will feel particular pride (will be proud of himself) every time one of these goals is achieved. And, conversely, every time they are threatened (if, say, he is obliged to ask for help), the person will feel the opposite emotion, shame. Thus, a personality trait (proud) is related to attaching a higher weight to a particular goal (self-esteem, autonomy); and, since that goal is particularly important to that kind of person, the person will feel the corresponding emotions (pride or shame) with a higher frequency or intensity.

Chapter 6: Modeling Pedagogical Agents: Design Phase

The designer usually finds himself floundering in a sea of possibilities, unclear about how one choice will limit his freedom to make other choices, or affect the size and performance of the entire system, or even any major part of it; much more important is to avoid choosing a terrible way, and to have clear division of responsibilities among the parts.
Lampson, [Lam83]

When communication is emotionally-oriented, intelligent software agents should be able to plan their (communicative) behaviour by means of an internal mechanism inspired by a consistent combination of cognition and emotion. The inspiration for the agent architecture comes from the recognition that thoughts and feelings are inseparable. The basic *sense-think-act* loop of a BDI agent [Rao95] may be modified to represent the idea that actions are a result of both *thinking* and *feeling,* as shown in figure 1.

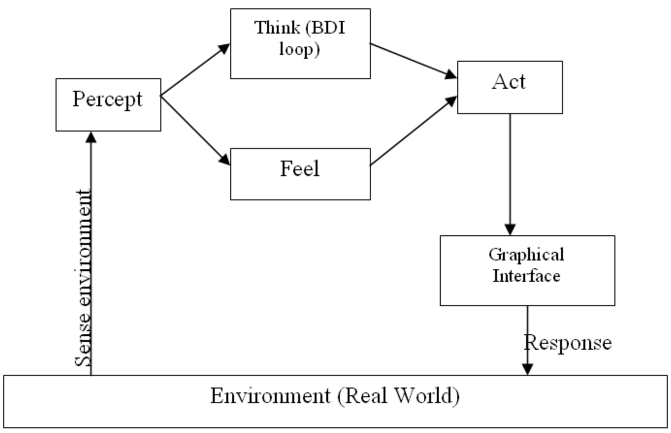

Figure 6-1 Emotionally Oriented Intelligent Agent Architecture

Representing concepts like *mood, emotional state* and *temperament* has been the goal of several research groups. Some of them extended language constructs employed for cognitive modelling to include representation of affective components [Car06]. However, these systems handle the two components separately. What's interesting, in our view, is to define a framework which enables (i) *to insure consistency between what an agent thinks (the cognitive state) and feels (the emotional state) over time* and (ii) *to exploit this* *consistent knowledge to plan a communicative act and to interpret the interlocutor 's emotional expressions.* In our proposal, the core of this framework is a truth maintenance system which works on *enforcing consistent emotional & rational behaviour.* As planning a communicative act requires predicting the interlocutor's behaviour consequent to this act, then *predicting this behaviour depends on how this enforcement is carried out.* The agent architecture in figure 1 allows a bidirectional kind of reasoning:

1. *what-if* type of reasoning (direction of the arrows) allows to reason on the *emotional and rational impact* of a communicative act on a given interlocutor starting from some knowledge of her mental state, and therefore to forecast –even if with uncertainty– how this state will be affected by communication;

2. *guessing* type of reasoning (opposite direction of the arrows) allows to: (i) *hypothesize the mental state which possibly produced a 'recognized' emotion* and (ii) *establish the event* (or the events) *which contributed to produce it,* by choosing among several alternative hypotheses.

Nevertheless, our intention is not to separate these two components but to adapt them to learning environments and to highlight the relationships between them.

6.1. Principles of an Emotional/Motivational Framework for Pedagogical Agents

A general consensus exists on the hypothesis that emotions are a biological device aimed at monitoring the state of reaching or threatening our most important goals, be they Terminal or Instrumental (see, for instance, [Car80]). Anytime something happens (or is assumed by an Agent to happen) that is likely to produce achieving or threatening a highly weighted goal, the biological device of emotion is triggered, with its whole syndrome of physiological activation, subjective feeling, expressive reaction and behavioral readiness.

This is, in our view, the relationship that holds between goals and emotions: goals, at the same time, *cause* emotions and *are caused by* emotions. They cause emotions since, if an important goal is achieved or threatened, an emotion is triggered: emotions are therefore a feedback device that monitors the reaching or threatening of our high-weight goals. At the same time, emotions activate goals and action plans that are functional to re-establishing or preserving the well being of the individual that was challenged by the events that produced them. So, fear triggers flight, anger triggers aggression; guilt triggers the goal of helping the damaged person or of escaping sanction, and so [Cas00b]. Personality is, as well, linked to goals and may be viewed in terms of weights that people put on different goals [Car80]; [Pog00]) for instance, sociable persons are ones who give importance to knowing and staying with other people. Of course, the weight attributed to goals may be biologically determined; but, if culture stresses the importance of some goals with respect to others, at least some aspects of personality may be culturally determined.

We now describe how we model the cognitive reasoning that is involved in triggering and regulating the display of emotions in the pedagogical agent, while its interaction with the student goes on. The following are, more in detail, the *factors we wish to consider in our approach:*

- *Temperament and personality influence:* the particular factors that, in a given situation, affect The pedagogical agent's propensity to feel and show emotions and the time they take to decay;

- *Socio-cultural context influence:* the way that the pedagogical agent's relationship with the student is influenced by agent's emotions triggering and display.

- *Dynamics of the Agent 's state:* the way the pedagogical agent's affective state changes over time;

- *Response decay:* the way that an emotion felt by the pedagogical agent evolves, in the absence of new specific stimuli, depending on the type of emotion and also on a particular personality trait that affects its 'persistence';

- *Multiple emotions:* the way that several emotions may be activated at the same time and may mix, according to either the 'tub of water' or the 'microwave oven' metaphors suggested by Picard [Pic97]. A user suddenly appearance in a session may trigger one or more emotional reactions, in the pedagogical agent; once activated, an emotion may last even for several minutes. Consequently, several emotions may coexist at a given time, either because they were activated by the same event or because some of them did not yet disappear while the others were triggered

Enhancing personality and dynamic behaviour to pedagogical agents can develop a new social-psychological model for animated tutoring agents similar with a human one. In particular humans can easily adjust their behaviour based on their role in a socio-organizational setting, where their actions tend to be driven emotions, attitudes and personality. The success of a life-like character in terms of user appreciation depends on factors like characters' role, competence and communicative skills relative to an application and its ability to present itself as a believable virtual personality. Our model includes the following concepts: personality, emotions and attitudes.

Personality traits correspond to patterns of behaviour and modes of thinking that determine a person's adjustment to the environment [Atk83]. Traits are basic tendencies that remain stable across the life span, but characteristic behaviour can change trough adaptive processes. Trait theories assume that an individual's personality profile can be described in terms of psychological traits that influence that person's behaviour. In other words, it is assumed that traits predispose people to behave consistently, indifferent to the situation. Thus the personality profile can be used to predict future behaviours.

Opposite to trait theories are social learning theories which assume that a personality is modified by each situation viewed as a learning experience. A person's behaviour may vary depending on the specific characteristics of the situation in interaction with the individual's appraisal of the situation and reinforcement history [Ban77]. Our agent's personality conceptual model uses both of these theories.

As an example of trait theories in our model let's consider a trait which specifies level of a student activity as a numerical integer value from the interval [-5, 5]. -5 means that the student is very lazy or he or she is not interested in participating in the common learning, 0 that our student is neither lazy nor energetic, and 5 to define that the student is very active, energetic. The default value is 0 and it is assigned by the agent to each student at the beginning of a new Passenger learning session. This value can be incremented or decremented based on student's behaviour during the learning process.

Mofat [Mof97] highlights the close relationship between personality and emotion, although they seem very different: emotions are short-lived and focused while personality is stable and global. He also considers mood rather short-live like emotion and not focused like personality.

As a product of evolution, emotions have a particular purpose: they have helped humans become the most successful species on earth. Emotions bypass the need for deliberative thought by providing biases toward the behaviours with better chances of survival—short-circuiting time-wasting rationalization. Other kinds of mammals also exhibit emotional capabilities with very similar reactions to humans.

Psychoevolutionary scientist Robert Plutchik shares such theories [Plu80]. According to Plutchik, there are eight primary emotions—associated in complementary pairs: anticipation and surprise, joy and sorrow, acceptance and disgust, fear and anger. These primary emotions can be observed in varied intensities (for instance, rage, anger, annoyance, terror, fear, apprehension). His theory states that it's not possible for humans to experience two complementary emotions at the same time; they balance out to provide diversity in the behaviours. Though, primary emotions can combine together into complex moods; acceptance and joy can be understood as love, fear and acceptance lead to submission, sadness and surprise form disappointment, and so forth. In psychoevolutionary terms, each emotion serves its purpose by triggering a reactive behaviour that's appropriate for survival.

Psychoevolutionary theory succeeds at explaining the reasons for emotions and provides a basic understanding of their roles as evolutionary tricks to improve survival rates. However, Plutchik's approach fails to take into account the cognitive process associated with emotions.

Elliott [Ell92] defines emotions as valence reactions to events, agents' actions, and objects, qualified by agents' goals (what an agent desires), standards (what the agent considers acceptable), and attitudes (what agent considers appealing). Ekman's ([Ekm82], [Ekm92]) basic emotion approach distinguishes those emotions that have different facial expressions associated with them: fear, anger, sadness, happiness, disgust and surprise.

Starting from a simple comparison with humans, our approach investigates the reasons for emotions' appearance: all emotions in embodied creatures are initiated by sensations. The notion of sensation can be defined as an immediate reaction to a creature's current status. By definition, sensations are experienced practically based on changes in the current situation. Two factors may cause sensations: the current perceptions (that is, stimuli from the environment), or cognitive activity (that is, thinking).

Perceptual sensations: are the sensations typically triggered by perceptions. The body detects stimuli from the environment, and the information causes an immediate reaction in the brain. For example, the tutor – agent may experience a sensation of surprise when a student appears suddenly in the middle of a learning session.

Cognitive sensations: represent the sensations triggered by reactions to the mental state (for instance, knowledge of the world or other emotions). Here, basic processing of information in the brain part causes the sensation. For instance, surprise can be caused by a student not being present, when the agent thought it should be there.

Sensations based on cognition and perception have common traits: both are triggered when a pattern is matched in the brain. With perceptions, this pattern is matched instantly based on sensory information. On the other hand, some cognition is necessary before a pattern develops in the brain (by thinking), which eventually engenders a sensation instantly when a pattern is matched.

Our approach embraces these theories and represents emotions as our agent-prototype's response to students' questions by synthetic speech, facial display and gestures. Verbal and non-verbal behaviour is synthesized in agent's mental model and interpreted in a learning-session (see Figure 6-5, or Figure 5-10). The facial display of our tutor –agent is limited to a predefined set of animations like happy, sad, etc. In order to extend the animations for our model we implemented also gestures to express emotions like confused: agent is lifting shoulders or don't recognize the question: put a hand to mouth. Figure 6-2 shows confuse and deny animations samples.

Figure 6-2 Animations for confuse and deny

Another important concept included in our model represents the agent's *attitudes* which characterize a relationship between an agent and a student. We defined attitudes as a complete set of emotions that constitutes someone's mental state at a particular time. The attitudes included in our model are based on degree of sympathy and trust. Those attitudes are: like and dislike, trust or don't trust. For example: if an agent likes a student it can offer him a second chance and hints in case that the student made a mistake or doesn't know the answer for a certain question. If the agent trusts a student it will give him the privilege to continue solving the problem in his/her manner even if the agent cannot foresee if the outcome will be the right solution or not.

We represented attitudes as numerical integers with values from the interval [-10,10], where above 0 means *trust* and bellow 0 means *don't trust*. 0 value corresponds to indifferent state: agent has no attitude toward the student. The default value for each parameter is assigned to 0 at the beginning of each semester. How these parameters are modified are shown in the following example: the following scenario can be assumed: during a session the students are asked whether they need help/hints or not after taking more time on a topic than the time allocated. When the students refuse the help and choose a different solution/action than the suggested one, the agent-tutor records this behaviour as indirect feedback. If that solution proves to be wrong this is materialized as decrease by 1 of the parameter's value. In case that the solution is good the parameter will be increased by 1. After each session those values are stored in a students' profile database.

The attitude like or dislike is correlated with the trait which defines the activity of a student. The agent *likes* a student with a high level of activity and *dislikes* a lazy student.

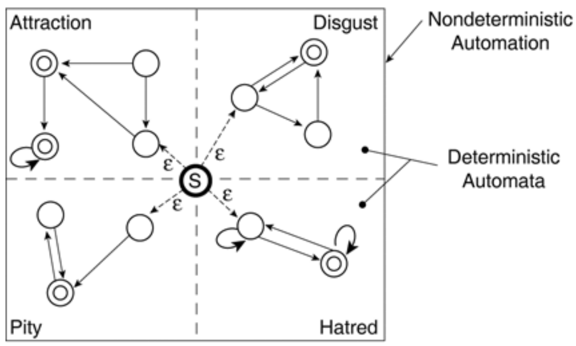

Figure 6-3 Feelings recognition module

In a current running learning-session if the activity level of a student is already 5, and that student is the current floor holder or he/she has always the tendency to take action/initiative the agent tries to temperate the student by taking the floor from him/her and passes it explicitly to the inactive users. After each session the tutor-agent realizes an individual report for each student based on the student behaviour during that session. The attitude of like is materialized as a bonus on the evaluation report: the agent recommends that the student should have a high mark at the final exam. In case of dislike the human tutor is informed of the negative behaviour of a student during the learning sessions. The human tutor will investigate the reasons which conducted to such of a negative behaviour and he will try to motivate the student to adjust his behaviour to a proper one.

Feelings are a more powerful concept; unlike emotions, they can be expressed about the past or future, and unlike sensations they do not rely directly on current state. Instead, feelings can be associated with arbitrary objects, in the past or future. Feelings are also very broad in that they can be applied to any concept (for instance, attributes or categories of objects). A feeling is a persistent association of an emotion with a class of object. Examples of human- feelings are disgust with varieties of food, hate for different types of car, or a phobia of moving obstacles.

The agent will be subject to four independent feelings: pity, hatred, attraction, and disgust. These are relatively easy to portray in the behaviours (especially in a learning session), and are sufficiently distinct from each other. Each feeling is recognized by a finite-state automaton. The automaton uses the data collected about the participants to decide what the agent's feelings are. For example, attraction is triggered for students which agent *likes* and *trusts*; disgust instead is felt for particularly *not-trusted* and *not-liked* students.

Instead of keeping each finite-state automaton (FSA) separate, these are grouped into one large nondeterministic FSA (NFSA) for convenience. Non-determinism allows the different automata to be merged together very simply, using (epsilon) ε transitions, as shown in Figure 5.

Additional feelings can be added very easily thanks to this mechanism. The FSA for the new feeling can be modelled separately, and grouped with the NFSA during the design. The feelings will mainly be used to generate new sensations. However, the feelings can be used by other behaviours as necessary—for example, selecting which student to ask to continue the exercise. Nevertheless, our research embraces all of these paradigms, but it focuses on emotions and how emotions are triggered.

6.2. An Analytical Approach for Emotion Modelling: OCC Theory

As previously said, we try to link emotions theory with agent technology in order to show how to motivate distance learners. For our aim, we find the so-called OCC theory by Ortony, Clore, and Collins [Ort88] the most appropriate one. First, the authors are very concerned with issues dear to the Artificial Intelligence community; for instance, they believe that cooperative problem-solving systems must be able to reason about emotions. Second, it is a very pragmatic theory, based on grouping emotions by their eliciting conditions l events and their consequences, agents1 and their actions, or objects l which best suits a computational implementation.

Further we highlight just a short overview of the structure of the OCC model based on types of emotions. It has three main branches, corresponding to the three ways people react to the world, which we have already mentioned. The first branch is very simple and relates to emotions which are arising from aspects of objects such as liking, disliking, etc. This constitutes the single class in this branch, namely that called attraction which includes the emotions love (if liking) or hate (if disliking). The other main branches are more complex, as they include further dimensions. We first present all items belonging to the second main branch, then those of the third one.

The second branch relates to emotions which are consequences of events. As a reaction to them, one can be pleased, displeased, etc. Further classes are described below.

fortunes-of-others

- person is pleased with event, is focusing on the consequences for the other and thinks the event is desirable for other: *happy-for*
- person is unpleased with event, is focusing on the consequences for the other and thinks the event is desirable for other: *resentment*
- person is pleased with event, is focusing on the consequences for the other and thinks the event is undesirable for other: *gloating*
- person is unpleased with event, is focusing on the consequences for the other and thinks the event is undesirable for other: *pity*

prospect-based

- person is pleased with event, is focusing on the consequences for self and thinks the consideration of the prospects is relevant: *hope* - which can be either confirmed (satisfaction) or not (disappointment)
- person is unpleased with event, is focusing on the consequences for self and thinks the consideration of the prospects is relevant: *fear* - which can be either confirmed (fears confirmed) or not (relief)

well-being

- *joy:* person is pleased with event, is focusing on the consequences for self and thinks the consideration of the prospects is irrelevant
- *distress:* person is unpleased with event, is focusing on the consequences for self and thinks the consideration of the prospects is irrelevant

The third branch, as the first one presented, also has only one class (although it is a richer one), namely the attribution class, comprising the following emotions:

- *pride:* person approves of self:
- *admiration:* person approves of other
- *shame:* person disapproves of self
- *reproach*: person disapproves of other

Also, it is worth mentioning that the prospect-based class was later augmented by Koda [Kod96] to include the element of *surprise* which materializes when *hope* or *fear* are neither confirmed nor discredited. This is important because surprise is an emotion which is normally included among the basic ones. Next section will highlight the state of the art in this research field.

6.2.1. Adapting OCC Theory

We have already motivated the use of the OCC theory in our framework. Additionally, this theory can be translated into a rule-based system which synthesises and generates cognitive-related emotions in an agent. Within this section, we will explain how rules look like in such a system.

Our approach uses the IF-THEN rules: the IF part tests either the desirability (of a consequence of an event), or the praiseworthiness (of an agent's action), or the appeal (of an object). The THEN part sets the potential for generating an emotional state (e.g., a joyful state).

Let $A(s; o; t)$ be the appeal that a student s assigns to the object o at time t, $P_h(s; o; t)$ the potential to generate the state of hate, $G = \langle gv1; : : : ; gvn \rangle$ a combination of global intensity variables, $I_h(s; o; t)$ the intensity of hate, $T_h(s; t)$ a threshold value, and $f_h()$ a function specific to hate. Then, a rule to generate a state of hate looks like:

IF $P_h(s; o; t) > T_h(s; t)$

THEN set $I_h(s; o; t) = P_h(s; o; t) - T_h(s; t)$

ELSE set $I_h(s; o; t) = 0$

This rule is triggered by another one:

IF $A(s; o; t) > 0$

THEN set $P_h(s; o; t) = f_h(A(s; o; t), G)$

Ortony et al. [Ort88] omit many implementation details; a difficult issue, for example, may be to find appropriate functions $f()$ specific to each emotion. However it was not very demanding to come up with such functions in a learning scenario.

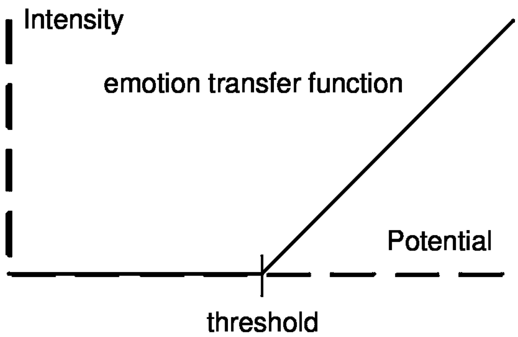

Figure 6-4 The transfer function (transfers emotions potential to intensities) [Pic97]

The intensity for each emotion under consideration is calculated by the following transfer function, (see Figure 6-4). It is based on the work of Ortony, Clore and Collins [Ort88].

This transfer is very simplistic, for each emotion it compares the emotion potential with a threshold, if the value is above the threshold then the intensity of the emotion is given by potential minus the threshold, else the intensity is zero. Moreover the intensity of an emotion is independent of the potentials of other emotions. A more advanced transfer function would be a "sigmoidal" one has proposed by Picard[Pic97]:

$$f(x) = \frac{g}{1 + \exp(-(p - t)/s)}$$

where g is amplitude of the intensity, p the potential of the emotion, t the threshold, s the steepness of the sigmoid. It is a bounded differential approximation of the transfer function proposed by Ortony et all. Still it lacks the property of taking into account the potential of other emotions. The decay of emotions is modelled by an exponential decay.

Before showing how to derive the IF-THEN-ELSE rule for different learning scenarios we need to highlight several notions concerning the learning material [Mar05]. The learning material for one semester is represented by a set of topics, $T = \{T_i | 1 \leq i < n\}$, where T_i represents a topic for a learning-session and n is the number of topics/lectures of the given course per semester.

Each topic T can be defined as $T= \{Q_k, P_j \mid 1 \leq k \leq n, 1 \leq k \leq m\}$, where Q_k represents a question and P_j represents possible answers by participants to this question, and also agent's plan/reply to each of these answers, *n* is the number of questions per topic, *m* is the number of possible plans per question. In other words we can define $P_j=\{M_j, A_j, | 1 \leq i \leq n\}$, where n is the number of possible answers/solutions for the question/task Q_i, M_j represents students reply to the question, while A_i is agents' response to these replies according to its goals.

There is a unary relationship *Time(t), t є T*, which represents the time allocated for each topic. The value of *Time(t)* is a number in time units and it differs from a topic to other. Also, there is a unary relationship *C(t), t є T*, which represents the credits allocated for each topic. The value of *C(t)* is a number in time units and it differs from a topic to other. We defined $C_{student}(t)$ as the relationship which defines the credits won by a *student X* during a *t*–topic learning session. It is obvious $C(t)=\Sigma\ C_{student}(t)$. It is worth mentioning that for each topic there is a predefined constant value *Y(t)* (called also minimum value) for the number of accumulated credits. In order to successfully pass a learning session, this relationship must be satisfied $Y(t) \leq C_{student}(t)$.

Besides these parameters we use below: *D(s; e; t)* for the desirability that a student *s* assigns to event *e* at time *t*, *W(s; a; t)* for the praiseworthiness that a student *s* assigns to ask for help to agent *a* at time *t*, $L = \langle lv1; : : : ; lvn \rangle$ a combination of local intensity variables, *x* is a pre-defined constant on the number of students (in our case x=3 which represents the number of students/session), *y* is a pre-defined constant on a student's number of accumulated credits, T_0 a predefined constant on allocated time for a student to perform a task, and ε is a fixed increment to the values returned by the emotion-specific functions.

The Pedagogical Agent will be subject to five independent emotions: joy, distress, pity, boredom and fear. These are relatively easy to portray using the IF-THEN-ELSE rules and are sufficiently distinct from each other.

- Rules for joy:
  IF $D(s; e; t) > 0$
  THEN set $P_j(s; e; t) = f_j\,(D(s; e; t)\ G; L)$
  f_j returns ($T_j(s; t)+\varepsilon$) IF student s has collected at least y credits AND $Time(t) < T_0$
  IF $P_j(s; e; t) > T_j(s; t)$
  THEN set $I_j(s; e; t) = P_j(s; e; t) - T_j(s; t)$
  ELSE set $I_j(s; e; t) = 0$

- Rules for distress:
  IF $D(s; e; t) < 0$
  THEN set $P_d(s; e; t) = f_d(D(s; e; t); G; L)$
  f_d returns ($T_d(s; t)+\varepsilon$) IF student s has not collected at least y credits AND for at least x-2 team-mates (other students) agent feels distress AND $Time(t) < T_0$
  IF $P_d(s; e; t) > T_d(s; t)$
  THEN set $I_d(s; e; t) = P_d(s; e; t) - T_d(s; t)$
  ELSE set $I_d(s; e; t) = 0$

- Rules for pity:
  IF $D(s; e; t) < 0$
  THEN set $P_i(s; e; t) = f_i(D(s; e; t); G)$
  f_i returns ($T_i(s; t)+\varepsilon$) IF student s has not collected at least y credits AND $Time(t) > T_0$
  IF $P_i(s; e; t) > T_i(s; t)$
  THEN set $I_i(s; e; t) = P_i(s; e; t) - T_i(s; t)$
  ELSE set $I_i\,(s; e; t) = 0$

- Rules for boredom:

  IF $W(s; a; t) > 0$
  THEN set $P_b(s; a; t) = f_b(W(s; a; t); L)$
  f_b returns ($T_b(s; t) + \varepsilon$) IF for at least x-1 students agent has identical type of emotion AND $Time(t) < T_0$
  IF $P_b(s; a; t) > T_b(s; t)$
  THEN set $I_b(s; a; t) = P_b(s; a; t) - T_b(s; t)$
  ELSE set $I_b(s; a; t) = 0$

- Rules for fear:

IF $D(s; e; t) < 0$

THEN set $P_f(s; e; t) = f_f(D(s; e; t); L)$

f_f returns ($T_f(s; t)$+ ε) IF student s has not collected at least y credits AND $Time(t) < T_0$

IF $P_f(s; e; t) > T_f(s; t)$

THEN set $I_f(s; e; t) = P_f(s; e; t) - T_f(s; t)$

ELSE set $I_f(s; e; t) = 0$

Figure 6-5 Emotions' display: distress and joy

Agent would experience joy (see Figure 6-5) when a student reaches its minimum number of credits for a session. Distress is "felt" when there are at least two students who couldn't reach the minimum value of credits within the given time. Similarly, agent feels pity for the student who couldn't reach the number of credits. Boredom is felt when for at least a certain number of students agent has the same feelings for example fear and the students do not perform in the agent's expected way (e.g. this turns agent bored and it tries to provoke its students by offering turns, providing more hints or helpful questions, making small theory demonstrations). Finally, fear is experienced when agent observes a student whose credits are so few and the allocated time may finish soon.

6.3. Modelling Emotions using Dynamic Belief Networks

One can easily notice that these rules are domain-oriented and the emotions formulas are not goal oriented. However Picard [Pic97] argues that many emotions do have complicated rules, therefore this basic framework can be used as a starting point by anyone who wants to generalize the rules, at least for similar scenarios. Another issue may be seen in calculating the intensity value for each emotion. Nevertheless, the major inconvenient that can be easily noticed is that the presented framework doesn't support emotion mixing.

In the 'Pure vs. Mixed Emotions' Section of her book on 'Affective Computing', Rosalind Picard introduces the following example:

> "After winning a Marathon, a professional runner described *feeling tremendously* ***happy*** *for winning the race,* ***surprised*** *because she believed she would not win, somewhat* ***sad*** *that the race was over and a bit* ***fearful*** *because during the race she had acute abdominal pain* (Picard, [Pic97], p. 171). A runner's friend, when assisting to that Marathon, was probably ***"happy for her*** *because she won the race, although a bit* ***envious*** *for not being able to participate to it and* ***sorry for*** *seeing her so tired, at the end."*

How is it that these two persons referred to feel this different mixture of emotions? Clearly, the main source of difference is due to the different structure of beliefs and goals of their minds. In the runner, the *intensity* of ***fear*** during the race was probably related, at the same time, to the *importance* she assigned to her goal of winning it and to variations in the *probability* of achieving this goal, that she dynamically revised during the race. The importance of this goal also affected the intensity of ***happiness*** (or ***satisfaction)*** of achieving it, while ***surprise*** was probably due to a difference between the likelihood she attached to achieving the goal at the beginning of the race and the final result. The ***sadness*** that the event was over might be a mixed emotion in its turn, some combination of ***nostalgia*** for a pleasant past event and ***hope*** to be again in a similar situation, in the future. The mixing of emotions in the runner's friend was probably due to a mixing of goals of approximately equal weight: ***happy -for*** is due to his desire of achieving "the good of his friend" (conditioned to her winning the race) ***sorry -for*** to his desire of "preserving her from bad" (illness, in this case) and ***envy*** to his desire of "dominating her", in a way.

So: differences between the two persons, in the example, are due to differences in the beliefs, the goals they want to achieve, the weights they assign to achieving them and the structure of links between beliefs and goals. Variations of these measures with time seem to govern cognitively generated emotions. Picard evokes the *generative mechanism* as the key factor for distinguishing between emotions that may *coexist* (by mixing according to a 'tub of water' metaphor) and emotions that *switch* from each other in time (by mixing according to the "microwave oven" metaphor). She suggests that co-existence may be due, first of all, to the differences in these *generative mechanisms.* But they may be due, as well, to differences in the decay speed between emotions that were generated by the same mechanism in two distinct time instants: for instance, "primary" emotions, like *fear,* and cognitively-generated ones, like *anticipation.*

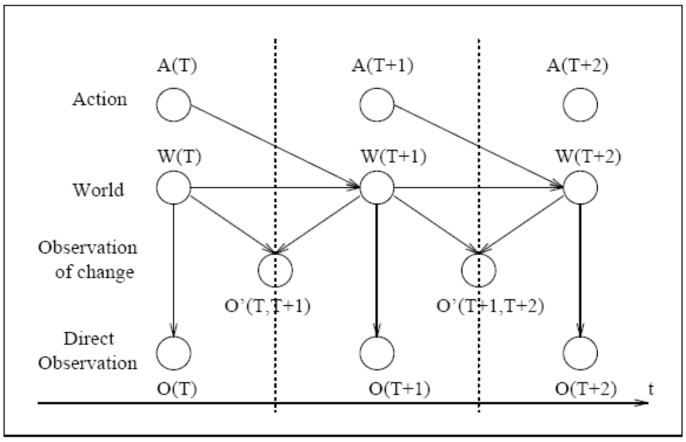

Figure 6-6 Generic Structure for a typical DBN [Nic94]

To represent the two ways emotions may mix up, the modelling formalism adopted should therefore enable representing their generative mechanism, the intensity with which they are triggered and the way this intensity decays with time. We claim that *dynamic belief networks* (DBNs) are an appropriate formalism to achieve the mentioned goals.

Dynamic belief Networks (DBNs), also called Dynamic causal Probabilistic Networks, have been of interest of interest as modelling tools for environments that change over time[Nic94]. DBNs have the following general characteristics. The nodes can be divided into three general categories: *world nodes*, which describe the central domain variables; *observation nodes*, which represent direct observation of world nodes, or the observable effects of a change in the state of a world node; *action nodes*, which represent the cause of a change in the state of a world node. Time is discretized; each time slice within the network represents the environment during that interval, and the structure within time slices is usually very regular. Figure 6-6 highlights a generic DBN structure for these node types. We enrich the proposed framework with the usage of DBNs for emotion triggering.

6.3.1. Emotion Triggering with Dynamic Belief Networks

As previously said, our departure point is that emotions are activated by the belief that a particular important goal may be achieved or threatened. So, our simulation is focused on the *change* in the belief about the achievement (or threatening) of goals of an agent A, over time. In our monitoring system, the cognitive state of A is modelled at the time instants { T, T+1, T+2, …}. Events occurred during the time interval (T,T+1) are observed, to construct a probabilistic model of the new state and reason about emotions that might be triggered by these events. A well known formalism for representing dynamic phenomena in conditions of uncertainty is that of DBNs.

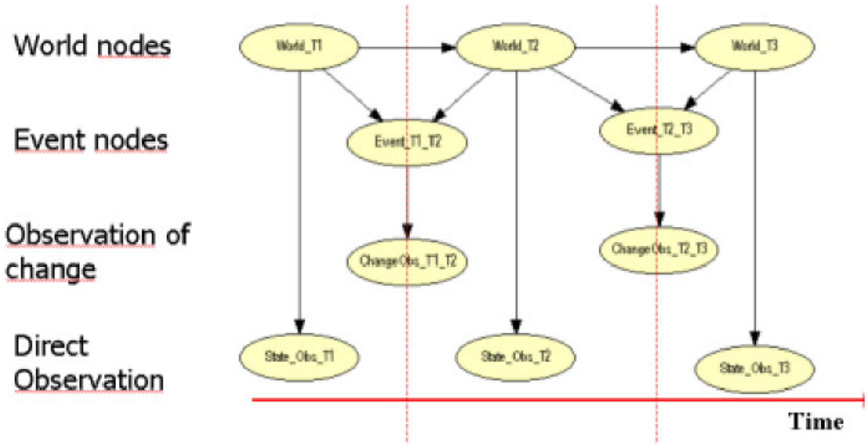

Figure 6-7 DBN in monitoring system [Nic94]

As shown in Figure 6-6 and Figure 6-7, DBNs are based on the idea that time is divided into time slices, each representing the state of the modelled world at a particular instant Ti. This state is described by means of a static belief network: World_Ti in the figure, with its observable state variables State_Obs_Ti. When DBNs are employed for monitoring purposes, two consecutive time slices are linked by arrows between the domain variables that have to be monitored. When something changes in the world, an event Event_Ti_Ti+1 occurs, that is observed through the variables in Change_Obs_Ti_Ti+1. The network is then extended for an additional time slice Ti+1. As a consequence, its structure and the probabilities of its nodes usually change [Pea00]. To avoid explosion in the complexity of the network (and therefore in the uncertainty propagation algorithm), pruning of time slices and of network parts is performed after a new observation is added to the model, with a mechanism of roll-up [Nic94].

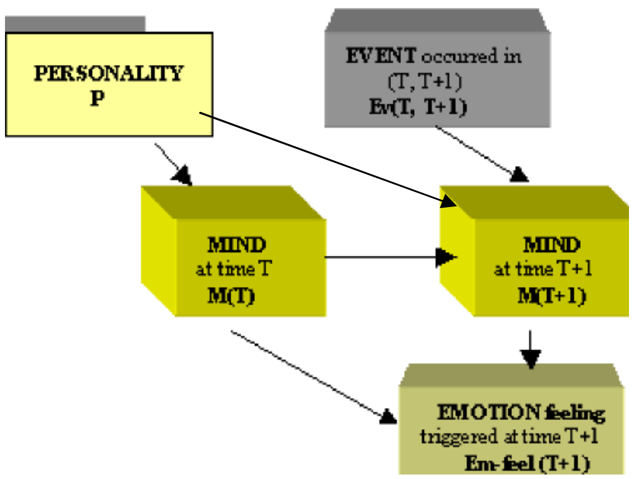

Figure 6-8 Mental Model of Agent

Figure 6-8 shows the general structure of our model of emotion activation, that includes the following static components:

1. M(T) represents the agent's Mind at time T, with its beliefs about the world and its goals;
2. Ev(T,T+1) represents the event occurred in the time interval (T, T+1), with its causes and consequences;
3. M(T+1) represents the agent's Mind at time T+1;
4. Em-feel(T+1) represents the fact that a particular emotion is activated, in the agent, at time T+1.

M(T+1) depends on M(T) and the event occurred in the interval (T,T+1). The feeling of emotions depends on both M(T) and M(T+1). We calculate the intensity of emotions as a function of two parameters: (1) the *uncertainty* in the agent's beliefs about the world and, in particular, about the possibility that some important goal is achieved or threatened, and (2) the *utility* assigned to achievement of this goal. More in depth, if:

a. A denotes the agent; G_i a high-level goal and Ach-G_i the achievement of this goal;

b. Bel A Ach-G_i denotes A's belief that the goal G_i will be achieved;

c. P(Bel A Ach-G_i) and P*(Bel A Ach-G_i) denote, respectively, the probabilities that A attaches to this belief, before and after the event Ev occurred;

d. W_A (Ach-G_i) denotes the weight that A attaches to achieving G_i,

e. Γ_A denotes the personality of an agent using Millon's[Mil97] parameters which are affected by the socio-cultural context of the tutor [Hof80]

then, according to the utility theory of Pearl [Pea00], the *variation of intensity in the emotion* (ΔIe) may be calculated as follows:

$$\Delta Ie=[P^*(Bel\ A\ Ach\text{-}G_i)-P(Bel\ A\ Ach\text{-}G_i)] * W_A(Ach\text{-}G_i)*\Gamma_A$$

In other words, ΔIe is the product of the change in the probability that G_i will be achieved, times the weight of this goal. In negatively-valenced emotions (such as fear, sorry for etc), the probability that a goal G_i will be threatened (Thr-G_i) comes into play, instead of its achievement, and for the Γ_A we use the bipolar value according to Millon's theory.

"Fortune-of-others" emotions *(sorry -for, happy -for, envy and gloating)* may be represented as points in the two-dimensional space ("desirability of the event", "empathic attitude"). Happy-for and envy apply to "desirable" events while "sorry-for" and "gloating" apply to "undesirable" ones; happy-for and sorry-for are driven by an empathic attitude, while gloating and envy are driven by a non empathic (or even contrasting) one.

Figure 6-9 shows the dynamic belief network that models how *happy -for* and *sorry-for* may be activated in an agent A who assists a student S solving a task T. This model shows that *happy -for* is triggered after believing that S will be able to solve the task with or without help in a good time. If this event occurs, the probability of the belief that the high-level goal of '*desiring the good of others*' will be achieved (S Achieve goal) increases. The intensity of this emotion depends on the variation of this probability, that is produced when evidence about this 'desirable' event is propagated in the network. It depends, as well, on the weight the agent attaches to achieving that goal; this weight is, in its turn, a function of the agent's personality. It is high for *altruistic* people, low for *egoistic* ones. This examples shows also how *sorry-for* is triggered using the opposite values of the same parameters used for triggering *happy-for*.

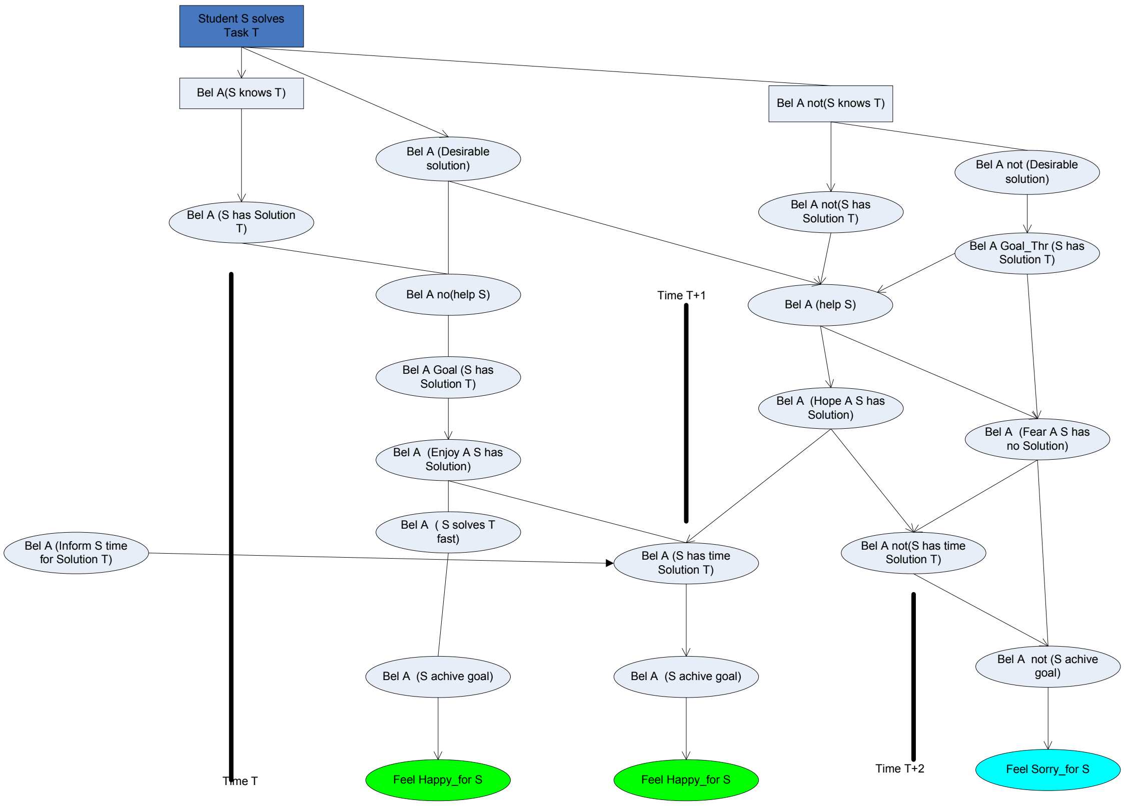

Figure 6-9 DBN for Happy-for and Sorry-for

Let us consider the DBN for activation of *happy-for* that is shown in above. As stated, the involved goal is, in this case, '*achieving the good of others'* (in particular, of S): happy-for is triggered by the belief that achieving this goal (Bel A (S achieve goal) increases over a given threshold. The root nodes of this sub-network, that may influence variation of this probability and are directly influenced by the considered event, are the following:

Bel A (S knows T)*"the agent believes that S can solve T"*

Bel A (Desirable solution) *"that solution is desirable"*.

If opposite values are considered instead, the threatened goal is 'dominating others' and the root nodes are the same as for happy-for, plus the additional belief:

> Bel A not (S has solution T) that *"the agent believes that the student cannot solve the task"*.
>
> Bel A Goal_thr(S has solution T) *"the agent believes that the student cannot achieve his/her learning goal"*.

The cognitive generation mechanism may therefore be directly represented, in DBNs, in *"the root nodes of the subnets affecting goal achievement or threatening, in the Agent 's mental state "*. In this representation, two emotions can coexist if and only if their cognitive generation mechanisms are compatible: that is, if all the root nodes in their activation subnets take compatible values. According to this model, *happy-for* and *envy* are examples of potentially coexisting emotions. As we saw in Figure 6-9, *the set of root nodes that their generation subnets have in common take compatible values when evidence about some observed event is propagated in the net:* in particular, the nodes *Bel A (Desirable Solution)* and *Bel A not (S has solution T)* must both be true. So, agents who are moderately altruistic and moderately dominant may be moderately envious and moderately happy-for at the same time, when they come to know that a desirable event occurred to a friend. For similar reasons, *sorry –for* is triggered using the opposite values.

Summary: within this chapter we highlighted a framework for emotion mixing. More or less fast switching in time may occur from ***Prospect-based*** emotions *(fear, hope)* to ***Well being*** *(distress, joy)* or ***Confirmation*** emotions *(disappointment, relief)*. Beliefs about achievement or threatening of high-level goals *("Desiring the Good of Others*" or "*Preserving Others from Bad*") are involved in this framework. Switching from one emotion to another is due to a change in the probability of the belief that a (desirable or undesirable) event will occur, is occurring or occurred. This change may be due to observation of different evidences originating from this event at different times. In our learning scenario, switching from ***fear*** to ***joy*** is closely related to the probability of the belief that the high-level goal of '*desiring the good of others'* will be achieved (S Achieve goal) increases. The intensity of this emotion depends on the variation of this probability, that is produced when evidence about this 'desirable' event is propagated in the network

Nevertheless, our framework cannot provide a concrete answer for questions like '*Which "fortune of others" emotions may mix up and how?*' We believe that if a cognitive model of emotion activation is taken (as in our case), a correspondence may be established between cognitive generation of emotions and the set of beliefs and goals that influence the variation of the probability of achievement (or threatening) of the goals that govern their activation.

Chapter 7: Architectural Model of the Pedagogical Agents

The architect should be equipped with knowledge of many branches of study and varied kinds of learning, for it is by his judgement that all work done by other arts is put to the test. This knowledge is the child of practice and theory.
Marcus Vitruvius Polio, Roman architect and engineer of the first century BC

There are many architecture proposals for ITSs. Carvalho [Car00] used a student model based on expert overlay model observing only pedagogic issues. The architecture showed in Figure 7-1 is a similar approach with Carvalho's architecture and it includes emotional and motivational modelling.

The system consists of a number of tutor-agents, representing workflow participants, several mobile agents that are responsible for the intercommunication process between the tutor-agents and a coordinator agent which provides directory service to tutoring-agents. Within this chapter we propose an architectural approach for our model based on the results of previous chapter.

In Carvalho's proposal the cognitive student model is elaborated by the Diagnostic Module, which consults the Student Background, the Student History and the Student Knowledge. The tutoring system also consults the Knowledge Tree, the Teaching Strategy and the Subject Domain, to prescribe teaching activities for the student. The Teaching Module accepts indications from the Diagnostic Module and interacts with the student presenting the activities selected.

While interacting with the student, the Teaching Module captures the primitive variables which are stored by the Emotional Module in the Interaction History database. The Emotional Module also consolidates the primitive variables after examining the Interaction History, the Behavioural and Temperament Structures database. The Emotional Module assigns the current values for the primitive variables and stores the averages and gradients in the student model. After that, the Emotional Module infers the behavioural and temperament records.

In the Emotional Strategy there are sets of primitive variables pointing for the increasing or decreasing of tertiary behaviours. There is also information about how tertiary behaviours affect the secondary behaviours and how these last affect primary behaviours. The Temperament Structure is also inferred based on the set of behavioural patterns contained in the Emotional Strategy.

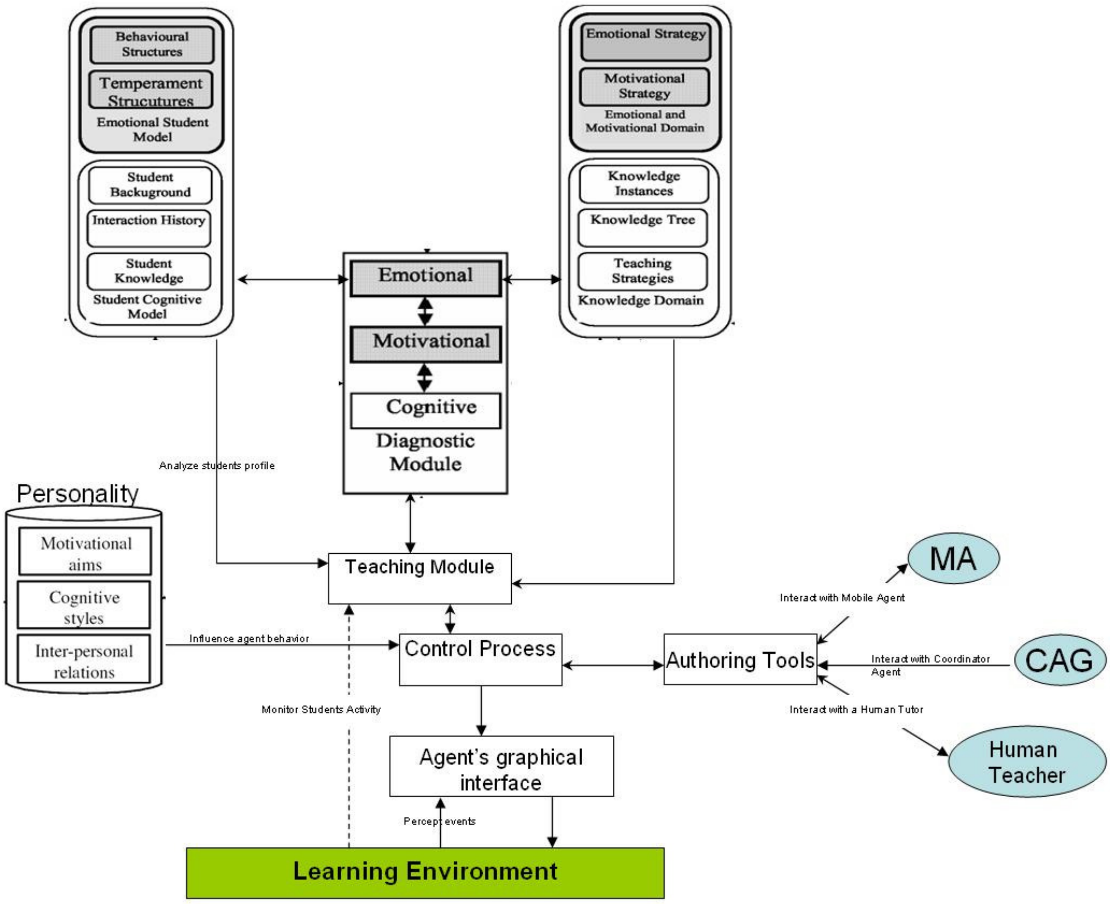

Figure 7-1 Architecture for an emotional pedagogical agent

The Motivational Module evaluates the presumed emotions of the student based on the Motivational Strategy, the Temperament Structure and the primary variables contained in the Motivational Domain and in the Student Emotional Model. Then it makes its prescriptions in order to deal with motivational diversions or even just to keep the student motivated. For instance, the Motivational Module may look for the motivational activity that seems more convenient for the student in case of miss-motivation perception. If this condition persists, the Motivational Module exchanges information with the Diagnostic Module to influence the choice of the next activities, and try a different Teaching Strategy.

The importance of the participation of experts from different areas is justified by ITS's multi-disciplinary. Disciplinary and pedagogy experts are required to describe the *Knowledge Tree,* the Teaching Strategy, the Instances of Knowledge and the Teaching Activities, using specific authoring tools.

In our case, the learning material for one semester is organized like a *Knowledge Tree* by a set of topics, $T = \{T_i | 1 \leq i < n\}$, where T_i represents a topic for a learning-session and n is the number of topics/lectures course per semester. Topics are designed to attract participants into an interactive dialogue and to avoid the "silence" during a learning session. Therefore each topic has a tree structure, with nodes that are: first question for the participants, possible answers by participants, and agent's response to each of these answers. Each topic T can be defined as T= $\{Q_k, P_j \mid 1 \leq k \leq n, 1 \leq k \leq m\}$, where Q_k represents a question and P_j represents possible answers by participants to this question, and also agent's plan/reply to each of these answers, *n* is the number of questions per topic, *m* is the number of possible plans per question. In other words we can define $P_j=\{M_j, A_j, \mid 1 \leq i \leq n\}$, where n is the number of possible answers/solutions for the question/task Q_i, M_j represents students reply to the question, while A_i is agents' response to these replies according to its goals.

There is a binary relationship between very two different topics T_i and T_j called *Precedence(T_i,T_j)* to highlight the fact that T_i is a prerequisite for T_j, where i<j. A student can learn a new topic only after finishing all its prerequisites. Also, there is a unary relationship *Time(t), t є T*, which represents the time allocated for each topic. The value of *Time(t)* is a number in time units and it differs from a topic to other.

Also, there is a unary relationship *C(t), t є T*, which represents the credits allocated for each topic. The value of *C(t)* is a number in time units and it differs from a topic to other. We defined $C_{student}(t)$ as the relationship which defines the credits won by a *student X* during a *t*–topic learning session. It is obvious $C(t)=\Sigma\ C_{student}(t)$. It is worth mentioning that for each topic there is a predefined constant value *Y(t)* (called also minimum value) for the number of accumulated credits. In order to successfully pass a learning session, this relationship must be satisfied $Y(t) \leq C_{student}(t)$.

The *Motivational Strategy* models sets of behaviour patterns that may indicate student's lack of motivation, to allow actions to keep motivation according to the expert recommendations. The expression "strategy" refers to a set of rules to orient the tutor in the decision taking process. The Emotional Strategy orients the tutoring system on identifying which primary variables are related to which behaviours in order to assemble the Behavioural Structures. The tutoring system also models the Temperament Structure based on these strategies and on the student's performance information. In our case we choose three strategies based on the progress of the student. .

i. ***Learning by Doing.*** In this strategy the tutor is very active. Within the context of the scenario it coaches the student step-by-step to perform the appropriate activity. At each step, the student can inquire about the purpose of the actions and activities performed. The tutor uses the structure of the activity trees to provide explanations (Figure 7-2).

The tutorial goals (activities) in Figure 7-2 give rise to a contextualized dialogue in the following ways:

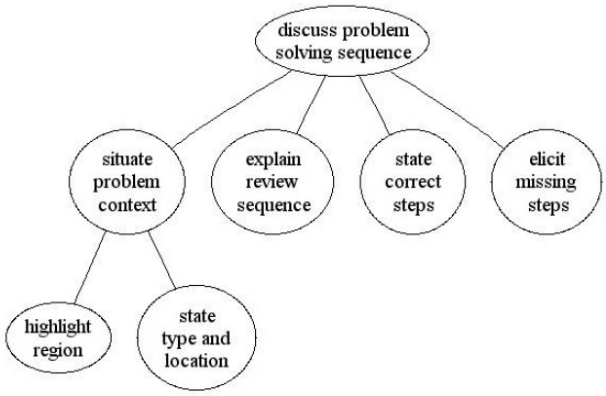

Figure 7-2 Sample activity tree

In turn (1) of the dialogue, it is the tutor's first mention of this problem, so the *situate_problem_context* activity is added to the activity tree, and the tutor describes the type of problem while highlighting its location in the ship display (regions are colored according to the type of crisis, e.g. red for fire, grey for smoke).

In turn (2) of the dialogue, the tutor tells the student why it chose to review this sequence so that the student will understand the tutor's subsequent turns. This corresponds to the activity *explain_review_sequence.*

In turn (3) of the dialogue, the tutor contextualizes the problem by reminding the student what they did (they sent repair 3 to set fire boundaries). This corresponds to the activity *state_correct_steps.*

Also in turn (3), the tutor asks the student what step of the sequence they omitted. Since the student does not provide the information the tutor is looking for (in turn (4)), the tutor provides further information about the context (turn (5)), and re-asks the question (turn (7)). This interaction is specified in the decomposition of the *elicit_missing_steps* activity (not shown in Figure 7-2).

ii. ***Practicing with Feedback.*** The tutor is less active in this strategy. The student performs activities without prompting by the tutor. When the tutor detects an error or missed action, it provides immediate remediation of the problem.

iii. ***Practicing with Checkpoints.*** Now the tutor intervenes minimally, only providing feedback at critical points (called checkpoints) of the scenario. Checkpoints are situations where error recovery is no longer possible. This strategy provides the widest possible latitude for performance of required activities and opportunities for correction of errors without tutor intervention. Here, the tutor is very similar to an operational aid.

Personality: As already mentioned, the behaviour of an individual can be modelled based on three main topics of the human personality (motivational aims, cognitive modes and inter-personal relations). These topics have a strong influence in the following aspects:

- How an individual can have perceptions of the environment in order to define his/her goals
- How an individual plan strategies to reach his/her goals;
- The way an individual perform these goals

As it can be observed, the proposed architecture (Figure 7-1) contains not only modules commonly used in a common agent architecture, such as perception, planning, goal definition and action, but also it is compose of a base, defining the agent's artificial psychological profile. As personality can be seen as the way in which an individual interacts with the environment, it is represented as an information base, or personality base. It is composed by the three aforementioned topics of the human personality (motivational aims, cognitive modes and interpersonal relations). This personality base has strong influence in the definition of the goals of an agent (goal definition module) as well as in the choice of the actions to perform the chosen goal (planning module).

In considering the personality base as a common knowledge base, this agent can be seen as a common BDI (Beliefs, Desires and Intentions), in which based on its beliefs, an agent can choose its goals as well as the needed actions to reach this goal. However, using common BDI agents for simulating human organizations could treat all members of this organization as having the same way of making decisions. This can be seen as an unnatural way of modelling humans. The use of personality will help to transform this modelling in a more natural one, in which the personality of an individual should and will be taken into account.

The way in which personality can influence in the decision making process of an agent is through the values associated to all bipolarities included in the three aforementioned areas. All bipolarities can be seen as attributes, considering both features in a complementary way (for instance, 90% of extroversion and 10% of introversion). These values will be used to define the probability of activation of rules (goals or actions) for an agent. It is important to emphasize that all polarities will be taken into account in this definition. These probabilities can be seen as weights that are taken into account in the decision making process. For instance, in order to reach a goal, an agent can have three possible set of actions to perform, the one which has the highest probability (weight) will be chosen. In this sense, it means that the lowest the probability is, the more distant the rule is from the personality of an agent. However, even rules with low probabilities can be chosen. This allows representing unexpected actions of an individual, which is a characteristic fact of the human beings. In case two or more rules have the same priority (it is a rare case, since twelve bipolar attributes are taken into account in the calculation of this value), an agent can choose one of them either based on its previous experiences or randomly

7.1. Control Process for Pedagogical Agents

The procedural system corresponds to the cognitive functioning of the agent and is based on BDI architecture. First, the agent perceives its environment. The perception is filtered according to the personal state (a relaxed agent will be less sensitive to malicious remarks) and physical state (an agent with a little deafness will less perceive a loud noise). Then, the agent updates its beliefs in consequence of the filtered message, decides what to do and then acts.

Finally, a base of psychological rules (established with experts) links all these parts. These rules manage personality evolution and influence, choice of actions and kind of communications. For instance, rules can be expressed in the following manner: "If I am very stressed and very anxious and angry and my neighbour is speaking to me harshly, therefore I will shout at him".

The control process of the agent is its functioning. It enables the agent to act, talk or react. Talking and acting are parallel processes. In our case, communication is not a means for the agent to collaborate or to negotiate with other agents. We only use communication as a vector to have our agents exchange their feelings. In this way, we cannot really consider exchanged messages as speech act, they are not a mean for the agent to achieve its goals. It is the reason why we have separated the two processes. This choice simplifies protocol management and thus, an agent can act and speak at the same time, which increases its reactivity. In this part, we will just describe the control of acting (Figure 7-3) since speaking is a very simple process (get the message, update the personal state and generate possible answers).

First, all the perceptions (messages) of an agent are stocked in a buffer and analyzed one by one. The first message of the buffer is analyzed (filter and update of beliefs, desires and internal state). Then, the strongest desire becomes the intention of the agent. At this moment, it checks if its current goal is still in adequation with its intention. These two steps are important because they enable an agent to change its intention before completing it (the change is between two elementary actions of the plan). We talk about a BDI agent with open-minded commitment [Rao95] – i.e. it will maintain an intention as long as it is still believed possible. Thus, the agent is very reactive to the environment evolution. After that, if the goal corresponds to the intention, the agent acts. Otherwise, it changes its goal to fit its intention.

We can see in Figure 7-3 that in plan search the agent has the ability to construct individual plans and collaborative plans to reach its goal. We do not yet work on this part which implies planning and requires a lot of investigations to have emotions impact on it.

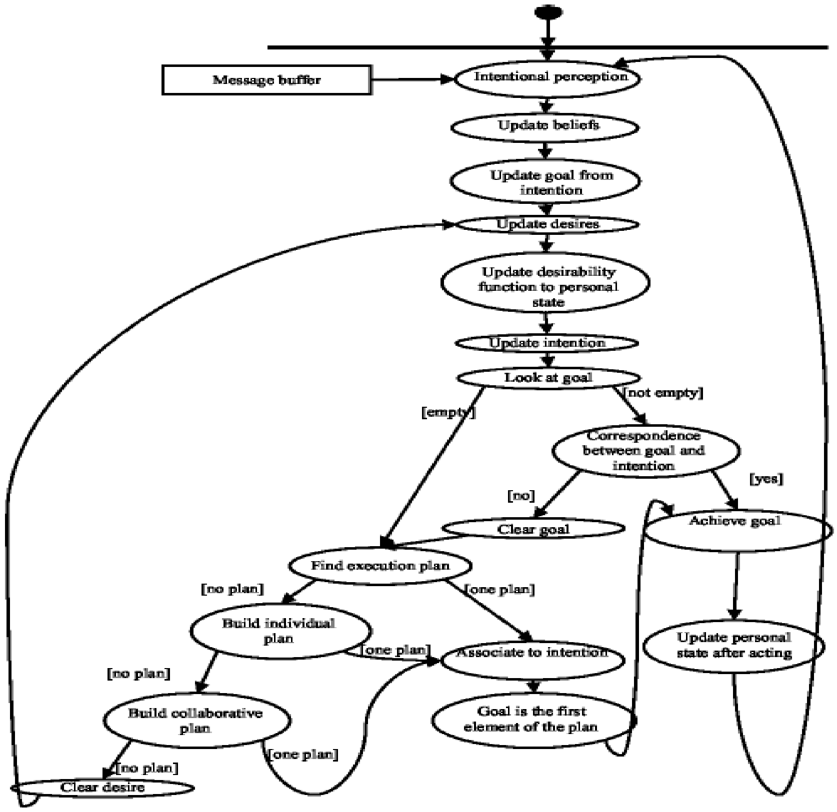

Figure 7-3 Control process of the pedagogical agent

We will now insist on how emotions and more generally how internal state take place in this cognitive architecture. The main influence is on reasoning. Indeed, emotional variables affect the action chosen by the agent. Following its personal state, an agent will not have the same choice of possible actions. On the other hand, relationships are memorized thanks to feelings. Therefore communication is also influenced by the internal state of an agent. Temperament is necessary to enable each agent to react differently to the same event. Finally, emotional state is strongly linked to the physical state which enhances the possible relational situations

7.2. Student Model

In general, student modeling entails developing models based on student behaviors and background knowledge to attain personalization and adaptation of learning environments. Apparently, the goal of student modeling is to build a model for assisting an intelligent learning environment in adapting to specific aspects of student behavior.

By definition, student modeling contains three essential elements: student behavior, background knowledge, and student model. In student modeling, student behavior is defined as the student's observable response to a particular stimulus in a given domain. Student behaviors are the primary input to a student modeling system and can be classified into simple and complex behaviors. Background knowledge in student modeling mainly contains the theory of the domain, such as concepts, principles, procedures, strategies, and so on, and the bug library, which includes misconceptions and other errors made by the students in the same domain. Naturally, background knowledge also consists of historical knowledge of a particular student and stereotypical knowledge in that domain. The student model can be an approximate, possibly partial, primarily qualitative representation of student knowledge about a particular domain that fully accounts for specific aspects of student behavior [Sis98]. With both student behavior and background knowledge, the student modeling system can construct the student model by using modeling approaches.

Each learner may have a different profile (preferred learning style, knowledge), portfolio (learning goals, tasks, assignments) and attitudes (behaviour) during the learning session:

- *Cautious:* responds or performs a modification on the common learning artefact only if he/she was asked to.
- *Leader:* it learns firmly and tries to stimulate the other participants to adapt themselves.
- *Inflexible:* persistent on its solution even if the solution is wrong or more time consumer than another one.
- *Flexible:* accepts any new ideas received from his/her team-mates.
- *Persuasive:* gradually tries to convince the others about its “ideal” solution
- *Improvising:* sometimes can be creative and propose new good changes other times can be because of lack of knowledge, and students simply try to guess the solution.
- *Selfish:* doesn’t care about the rest of the group and tries sometimes to prove that he/she is “the best” even if actions like this can harm the progress of the group.

7.2.1. Introducing Fuzzy Cognitive Maps

Despite the numerous classifications on the learner’s style which have been proposed so far, one easily recognize the difficulty to classify many learners as of a certain kind in any given classification. No matter the classification in hands, the subsets should rather be characterized by their fuzziness than by their compactness. To the purpose of the presentation of a classification by Fuzzy Cognitive Map (FCM), we consider as learning style model, introduced by Kolb [Kol84], according to whom “we learn by conceiving and transforming our experiences”. The proposed method can be easily applied to other classifications of learner’s style in an analogous manner. According to Kolb’s classification, conception and elaboration of information are the two dimensions of learning process. It has also been pointed out that each dimension of the learning process presents us with a choice. For example, it is virtually impossible to drive a car (Concrete Experience) and at the same time to analyze a driver's manual about the car's function (Abstract Conceptualization). Therefore, we resolve the conflict by choosing. Hence, in order to conceive information one has to choose between Concrete Experience and Abstract Conceptualization. As a matter of information elaboration one has to choose among Reflective Observation or Active Experimentation. Such choices determine the learning style. According to Kolb's model, the four learning styles and the corresponding per learning dimension choices are presented at the following table.

Table 7-1 Middle layer of basic learner [Kol84]

Active			x	x
Abstract Conceptualization		x	x	
Reflective	x	x		
Concrete Experience	x			x

Fuzzy Cognitive Maps is a soft computing tool which can be considered as a combination of fuzzy logic and neural networks techniques. FCM representation is as simple as an oriented and weighted compact graph. Each vertex of the graph represents a concept which express explicitly or implicitly certain characteristics a learner has, or the main learner styles according to Kolb [Kol84] classification. The vertices of the graph connect pairs of the user characteristics if and only if there is a certain relationship among them. Each concept is characterized by an integer indicating the significance of the characteristic in the model. So, an integer of great value indicates the importance of the concept, as an integer of low value indicates a concept of minor meaning. In order to transform this values of concept significance into the scale of [0,1], which is in use by the fuzzy logic methods, we introduce an appropriate simple linear transformation. As a matter of fact, the labels which stand for the weights of the graph's oriented edges, should also be de-fuzzified and transformed to values in [-1,1]. The final graph is designed in a way that easily its observer can see the significance of a concept and the influence each concept has on another. As of the simplicity of its structure, an expert can easily add more vertices and edges in case new concepts should be introduced or more experts are asked to be represented in the model.

7.2.2. Modeling Student Profile using Fuzzy Cognitive Maps

Each class conforms a layer of concepts, coloured to a particular colour. The inner layer is conformed by the four learner's profiles according to Kolb's classification. The middle layer is the layer of basic learner's characteristics, according to Table 1.

The outer layer has the measurable learning activity factors (LAF) which are subjects to be diagnosed by the machine. Such factors influence directly the learner's characteristics. The oriented connections between concepts – vertices of the graph are represented by arrows. The connections may show positive or negative influence LAF can have to LCs and LPs. A negative connection reduces the probability to diagnose a certain LP in case of strong presence of a connected LAF.

To explain these approaches the following related definitions are required:

1. the set of elements $C_i \in \Theta$, where $\Theta = \{LAF\} \cup \{LC\} \cup \{LP\}$;
2. **A,** a linguistic term of a linguistic variable (e.g. almost absolute cause) ;
3. A measurable numerical assignment compact interval $X \in (-\infty, \infty)$;
4. $V \in X$, a linguistic variable which is a label for $C_i \in \Theta$;
5. $\mu A(Ci)$, the membership value representing the degree of membership of θi to the set of elements determined by linguistic term A.

Since we do not expect that all LAFs have the same degree of effectiveness and causality on their adjacent LCs and LPs, weights must determined in order to express the degree of effectiveness and causality in case. As cognitive psychology experts mostly describe qualitative behaviour using linguistic variables, it is necessary to introduce a transforming algorithm to map the values of such linguistic variables into membership functions. Watanabe's [Wat97] membership functions direct estimation methods take an approach by asking experts to grade an event on a scale. Using such grading, we make use of the transform which appears in Georopoulos et al. [Geo03]. According to the proposed scheme, each fuzzy set corresponds to a membership function shown in the Figure 7-4, where fuzzy sets describe the degree of causality corresponding to membership functions $\mu A(Ci)$, A={ewc, wc, oc, sigc, strc, esc, aac}.

The proposed fuzzy sets and their corresponding membership functions are:

- M_{ewc}(extremely weak cause) the fuzzy set for causality around 10% with membership function μ_{ewc}.
- M_{wc}(weak cause) the fuzzy set for causality around 20% with membership function μ_{wc}.

- M_{oc}(ordinary cause) the fuzzy set for causality around 35% with membership function μ_{oc}.
- M_{sigc}(significant cause) the fuzzy set for causality around 50% with membership function μ_{sigc}.
- M_{strc}(strong cause) the fuzzy set for causality around 65% with membership function μ_{strc}.
- M_{esc}(extremely strong cause) the fuzzy set for causality around 80% with membership function μ_{esc}.
- M_{aac}(almost absolute cause) the fuzzy set for causality around 90% with membership function μ_{aac}.

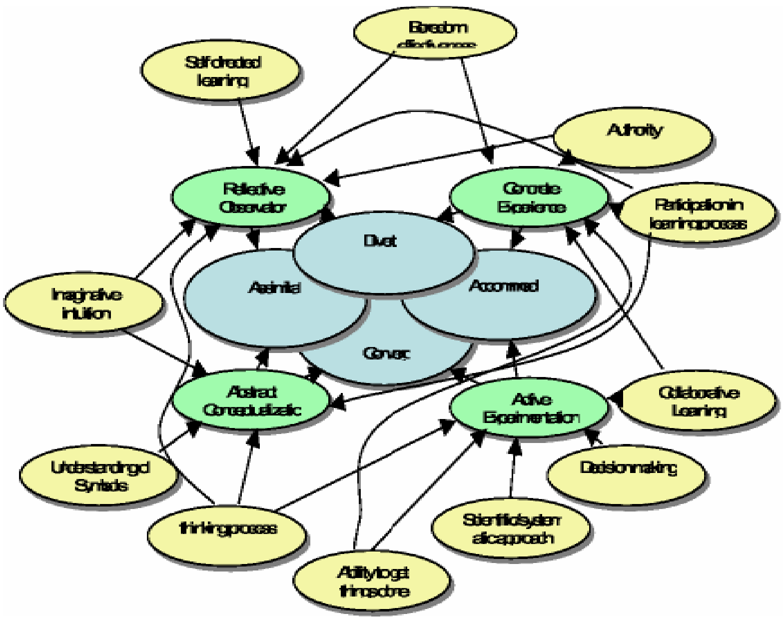

Figure 7-4 Map for detecting motivation in learner [Geo03]

For example when considering the set of learners characteristic for a given C_i (e.g. the specific learner's Synthetic Ability) in the set Θ = LC, whose linguistic value V, describes the `degree (A) of learner's characteristic', has. For example the `degree of learner's characteristic' might be in the range of evaluation units X =[0,10]. Moreover, how ` Active Experimentator' the learner's Synthetic Ability might be, is the value of the membership function μ_{strc}(Active Experimentation).

Direct rating presents randomly selected $\theta i \epsilon \Theta$, with values $V(\theta i) \epsilon X$ to subjects who answer the question "How **A** is $\theta(V(\theta))$?" . In other words the question put to the expert is: "How much reflective observator is learner θ?" (Note that ` significant causal ' is different from `causal') and they respond by using a simple indicator on a sliding scale. Then, using the experts' opinion of the range of causality, a simple calculation reveals μ_{strc}(learner θi). This experiment is repeated for other learner and, indeed, the same learner θi repeatedly to reduce error. In the late case, experimental outputs are summed up and using the defuzzification method CoA, is transformed into a numerical value which takes place for the weight $W_{\theta,i}$.

The Algorithm:

Let N be the number of concepts in the FCM

- Set the number *k* of learners
- Set initial values n=0, $V^0(C_i)$ for i = 1, 2, .., N from the learner's profile database. Data have been stored as the learner responded to certain tests. Data have been stored as linguistic values Ai , and have been turn to fuzzy degrees $V^0(C_i)$ for all concepts except those in LP. Concepts in LP are set equal to 0 for n = 0.
- Set the initial values for $w_{j,k}$ according to given information.
- For n=n+1, apply the relation (1) and set values $V^{n+1}(C_i)$. Update learner's profile database. Following the defuzzification the weights at the edges of the graph are presented as elements of the adjacent matrix W_n.
- Set $\mathbf{V}^{n+1}=W_n\mathbf{V}^n$, where
- If a C_i does not be influenced by any C_j , j≠i then $w_{j,i}$=1 at present n
- If a $V^n(C_i)$= **m** $[V^n(C_j)]^{-1}$, for a given measure of competence 0<m<1, then set $w_{i,j}$= **-m**
- Use the unipolar sigmoid function to transform the coordinates of V^{n+1} into the interval [0,1].

If max $0 < i < k|V^{n+1}(C_i)- V^n(C_i)| < \varepsilon$, ($\varepsilon$ >0) then stop and store as result the learner's profile which has the highest value $V^{n+1}(C_i)$

The above method has been developed to provide a fully computerized procedure which will be able to diagnose the learner's profile. An Adaptive Educational environment, which supports collaborative learning, will take fully advantage of the proposed algorithm, in order to "see" the learner and to tailor the learning material to his special needs. Another possible approach to the recognition of learner's profile could be the application of learning methods.

7.2.3. State of the Art: Overview of Symbolic Machine Learning

Learning is an essential component of intelligent agent systems. It is also a fundamental component in MAS-based DLEs. Without any learning abilities, DLEs cannot benefit from their past experiences or adapt to dynamic changing learning environments.

The goal of symbolic machine learning (SML) is to induce new knowledge from existing or past data for future usage, or to compile knowledge in order for existing knowledge to improve its accuracy and performance. From the viewpoint of the machine-learning approach, the SML can be classified into two main categories: supervised learning and unsupervised learning.

The two types of learning are distinct, either from required data input or the tasks that they can address. Both supervised and unsupervised SML are constructive for student modeling in DLEs.

Supervised SML: Supervised learning requires all instances (data input) to be labeled with defined classification. A labeled instance is viewed as a pair (*xi, ci*), where *xi* is the given instance for learning, and *ci* represents corresponding classifications. These classification definitions (*ci)* are given by an external "teacher" from the domain applications, hence, the name of learning supervised SML. The task of supervised SML is to learn a function *f* (it may be a description or rules), where *f(xi) = ci* for allinstances. Many machine-learning algorithms can be utilized to learn the function *f*. These algorithms (Briscore & Gaeli, [Bri96]) include but are not limited to decision tree, instance-based reasoning, case-based reasoning, Naïve Bayers, neural networks, rough set, regression, version space (also called theory revision), inductive logic programming (ILP), rule-based production system, covering algorithm, etc. To discuss these machine-learning algorithms in detail is out of the scope of this chapter. Interested readers can refer to books such as "Symbolic Machine Learning" by Briscore and Gaeli [Bri96] and "Machine Learning" by Mitchell [Mit97] for more information.

The learned function *f* or model has to be evaluated for its performance using new, previously unseen instances. There are several approaches (Efron, [Efr87]; Breiman et al., [Bre84]) for evaluating the learned function *f* or model. Such approaches include hold evaluation, cross-evaluation, bootstrapping evaluation, LOBO (leave-one-batch-out) [Kub98], and so on. After evaluation, the ideal function *f* or model will be selected for domain applications such as student modeling in DLEs.

Unsupervised SML: Unsupervised SML differs from supervised SML in that it does not require any classification information from external systems. Unsupervised SML is used to find the "commonality" or "regularity" in the given instances or examples. In this scenario, the system is not required to find a function or model from the given instances. To determine if unsupervised learning is successful, the testing set of the instances is taken to action by checking if they exhibit the same regularity as was discovered in the training set of instances.

The main technique for unsupervised SML is conceptual clustering. Conceptual clustering is the grouping of unlabeled instances into various categories, where conceptual descriptions can be formed, and the instances are with the same regularity.

The discussion above clearly demonstrated that student modeling can benefit from symbolic machine learning. By applying SML to student modeling in DLEs, we will be able to extend the background knowledge in the student's learning domain. To this end, both supervised and unsupervised SML are constructive to student modeling.

Table 7-2: The compression of some published student modeling systems

Student Modeling systems	**Developers**	**Publishing Date**	**Task for Student Modeling**	**SML Algorithm**
PIXIE	*Sleeman et al.*	*1990*	*Extend domain background knowledge*	*Rule-based production system*
ASSERT	*Baffes & Mooney*	*1996*	*Induce student model*	*Theory reversion*
MEDD	*Sison, Numao, & Shimura*	*1998*	*Extend domain background knowledge*	*Conceptual clustering*
THEMIS	*Kono et al.*	*1994*	*Induce student model*	*Decision tree (ID3)*
ACM	*Langley & Ohlsson*	*1984*	*Induce student model*	*Decision tree (ID3)*
DEBUGGY	*Burton*	*1982*	*Induce student model*	*Covering algorithm*
MLTUTOR	*Guven & Blandford*	*1998*	*Induce student model*	*Decision tree (ID3)*

Generally speaking, the procedures of applying SML to student modeling contain the following steps:

- Collect data on student behavior or background knowledge from systems.
- Represent the data with specific attributes.

- Use SML algorithms to generate the model or to extend the background knowledge.
- Evaluate the generated models and select an ideal one.
- Apply the selected model to DLEs for intelligent support.

Up to this point, in the AI and e-education research fields, there have been a larger number of research achievements for student modeling. Many student modeling systems have been developed, and their results have been published in various journals and referenced in conference proceedings. To help the reader in accessing these remarkable results, we have summarized some results of student modeling in Table 2. We are not going to describe each system in detail; however, interested readers could find their details from the relevant references.

The summary in Table 7-2. clearly shows that almost all major SML algorithms, regardless of their types, have been applied to develop student models or discover background knowledge in student modeling. Because SML is an experimental technique, it is impossible to say which algorithm or approach is better for student modeling. It depends entirely on the requirements of the domain application and the collected data. Therefore, evaluation of the learned model or knowledge is indispensable for different application domains

Chapter 8: Putting Theories to Practice: Passenger Agent Tutor

In this section we will present an implementation and critique of our design, discussing in more detail: (a) the requirements specification with respect to Passenger; (b) an implementation of the Passenger Agent Tutor (PAT) design; (c) some experimental results exploring the PAT design- and finally (d) the strengths and weaknesses of our design – i.e. how our approach contributes to the field of intelligent autonomous agency by addressing some of the problems identified with the agent architectures described in chapters and

8.1. Synchronous Groupware Passenger

Modern Software Engineering in any case signifies teamwork. The worldwide extension of the data networks and continuing globalization add another component software engineering: the development of worldwide distributed teams. The use of this forward-looking form in university education could make a special contribution to the way in which students work, and are worked with, in future [Hun99].

When comparing the professional field of Software Engineering with the Software Engineering education the following aspects have to be taken into consideration:

- working in a team, dividing up the given task into sub-tasks, discussing intermediate results.
- the usage of the new media and communication technologies also requires that students should work in a completely new scenario.

For the special case of the software engineering lab regarded here, the emphasis lies on the development of a proper educational environment for the support of the students during a meeting in the context of a Software Engineering lab. An educational environment must be constructed effectively so that students who participate into distributed teams learning process through Internet or information network could interact with other team-members timely and friendly. Namely, the collaboration is the most important subject with a view to designing an advanced computer supported educational environment. In order to attain this subject, it is necessary to discuss an architectural framework from the required resources points of view:

- Technologically mediated communication channel
- Shared workspace for a team/group

- Personal workspace
- Learning materials/ learning tools

In order to create an educational environment for the spatially distributed teams, a synchronous groupware called “PASSENGER” was developed at our university throughout the last years. The specified requirements for a groupware used in a Software-Engineering-lab give a direction how the function- and application-classes of the Passenger-Client were defined. A client-/server architecture has been chosen whereby the server is located at the university, due to the fact that the university plays a major role in this configuration.

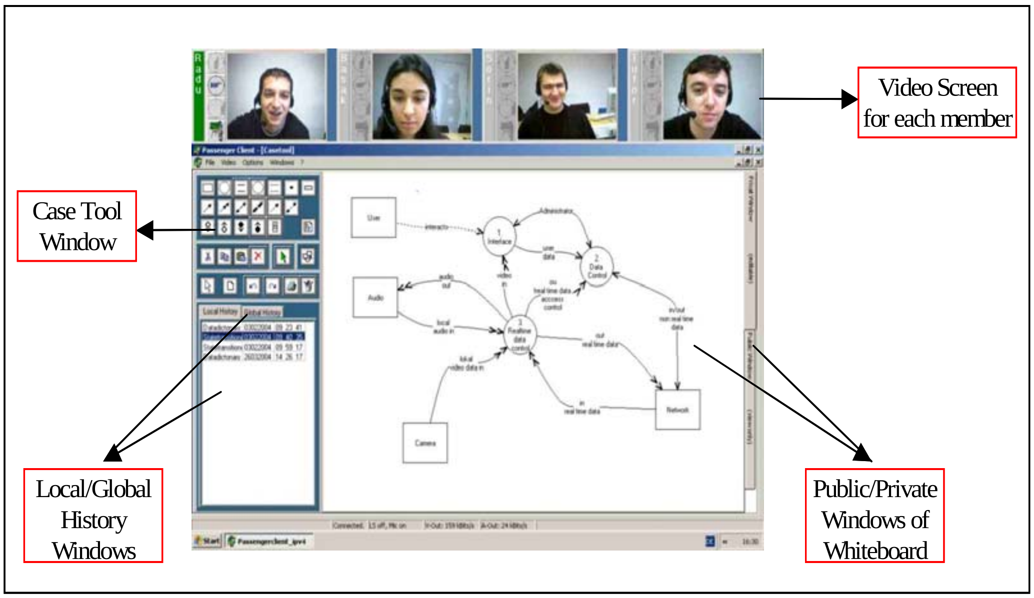

Figure 8-1 Passenger Client

This groupware (see Figure 8-1.) is composed from three modules: the communication component which contains video screens of each participant and a cooperation component which allows students to interact together on a common artefact. The participants can be three students and one tutor. Three essential differences of our tool compared to publicly available solutions can be specified:

a. *Passenger floor control*: the advantages of the developed Floor-Control [Dom99] protocol are to guarantee a defined fairness and to prevent the mutual exclusion and blocking. Thereby, the fairness definition is based on a theoretically equal distribution of the Floor-holding concerning the occurrence. Anyway this type of equal distribution is not forced. Since the defined roles in the group process model are opposed to any kind of equal distribution, the arrangements to guarantee the equal distribution of the Floor were renounced. In particular, the Passenger Floor Control does not limit Floor-holding duration.

b. *A user interface designed to support group awareness*: [Dou99] the design is based on common requirements and the special requirements from the analyzed group behaviour. For all design decisions thought has been given to the requirement for measures to increase the group-awareness. Therefore a concept for the positioning and resizing of the communication windows was developed and implemented. Especially solutions for the Floor-Control and the group-awareness were developed during the design of the user interface. The developed Floor-Control is specialized for small, closed groups. It removes the lacks of existing systems concerning the fairness of the Floor assignment and the implemented measures to mutual blockings.

c. *The whiteboard concept materialized in tools to carry out software engineering tasks*. The implemented PASSENGER-CASE-Tool for the software design features its concept for realizing the private and public work area and its process specified support for software engineering. The separated realization between the work and the display area enables the Floor-Holder firstly to simultaneously access the last two design documents. Due to this fact he is simply able to compare the two schemes by switching between the work and display window. The rest of the participants have the same possibility under condition that they transfuse the content of the display field of their work area. The transfer of arbitrary documents to the work area of the Floor-Holder and there to the display area of the others, should not be commented, in order to set the process rights and the access to the speaking channel in a shareable Floor-Control.

A more detailed description of the Passenger environment can be found in [Mar04b]. Next section will introduce the requirements for tutoring agents.

8.2. Requirements for an Agent Based Tutor for the Passenger Groupware

The traditional "Computer Supported Cooperative Work and Software Engineering" lab at UDE is conducted as a project setup of student teams, each consisting of four persons: three students and one tutor, where the same tutor can be in several virtual teams. That can cause problems in terms of tutors' availability if the virtual teams meet at the same time but also if the teams meet at times outside the tutor consultation hours. To make sure that at least a virtual tutor is always available agent technology was used.

Although there is not a predefined formula to calculate the complexity of collaborative learning in virtual environments, even if there are several attempts in this direction [Fju01] we observed from series of tests [Mar04] which took place in the laboratories of UDE, three knowledge-levels (see Figure8-2) regarding the complexity of collaborative learning:

- L_s the students can learn together without any intervention from the tutor's side,

- L_a the students can learn only with the intervention from a tutor: agent or human

- L_h students can learn only with human supervision.

As it can be noticed if the complexity of a task is not so high the students can learn together using the collaborative learning paradigm and without the intervention of a tutor. The role of an agent – tutor in this case is just to monitor the collective and individual activities of the participants.

If the level of complexity of a given task is increased then besides the follow-up role the agent – tutor must also be able to provide individual help to those participants who cannot keep up progress with their groupmates or collective help: the students might reach a deadlock point: no one is able to continue or to provide any ideas in order to accomplish the common learning task.

Given the fact that learning is a complex cognitive activity, learners cannot rely solely on machines when capturing and mastering knowledge of a certain domain. Still tutoring agents are useful tools in the sense that in many cases improve the learning effectiveness but it can happen that the agent-tutor might not be able to provide help: answers, solutions or hints on future steps to students if the task/exercise is too complex.

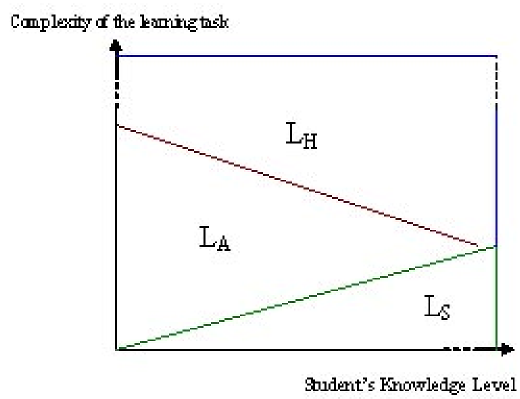

Figure 8-2. Knowledge Levels [Mar04]

Therefore the key challenge for us was to design and develop an intelligent agent-based system that should replace the human tutor only to a certain level and to integrate it within the Passenger-environment. The main important roles which our prototype should have are:

- Tutoring role: it can present new topics during a Passenger-session, ask questions and understand the students' responses, and the students can ask clarification questions and receive appropriate explanations but it should be able also to provide individual help to a participant if it is necessary.

- Follow-up role: it can monitor the whole collaborative process and also individual activities. After each session the agent provides to a human-tutor a full report concerning the activities of each participant within that session.

8.3. A MAS for the Passenger Groupware

Choosing an environment is an important decision for agent researchers and developers. A key issue in this decision is whether the environment will provide realistic problems for the agent to solve, in the sense that the problems are true issues that arise in addressing a particular research question.

Due to the fact that a large number of Passenger sessions can run simultaneous at the same time, it appeared the need of a system to monitor and manage all these sessions. The first step in this direction was a web-based system called Watchdog which monitors the activities of all Passenger Servers (see Figure 8-3).

The following scenario can be assumed: three students would like to meet on-line using Passenger and learn together or discuss about the topics learned from their previous lecture. Also, besides the fact that the students might need supervision the need of a tutor may appear if there is a disagreement between the students. Therefore, during that session if help of a tutor is needed, one of the students can press “Call Tutor” button. This request will be send to the Watchdog and the system will provide this request also with the exact parameters of the session (IP address of the Passenger server where that session is hosted) to any available human-tutors.

If there are no human-tutors available, then a tutor-agent will connect to the current session and it will try to provide necessary help to the participants. In Passenger learning environment the presence of an intelligent-agent, that can perform a kind of tutoring role, could be benefic in helping the students to reach their common goal.

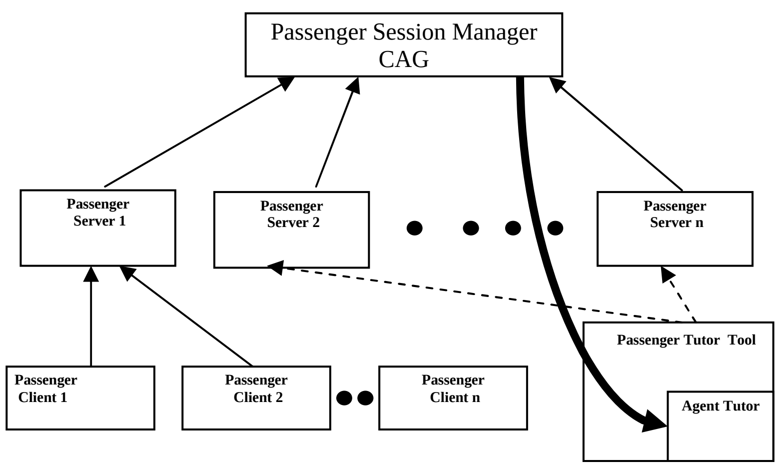

Figure 8-3. Structure of Passenger System

The MAS for the Passenger system consists of a number of Tutor Agents for Passenger running sessions, mobile agents used by the tutoring agents to communicate amongst themselves and a Coordinator Agent (CAG) responsible for coordinating all the tutoring and mobile agents. Next we are going to briefly describe

8.3.1. Tutor Agent

We chose a human character instead of an animal in order to impose learners a degree of realism due to the fact that the environment where our prototype acts is virtual environment for distance education system. To support the needs of task-oriented collaboration, Passenger-Agent Tutor (PAT) includes the following primitive actions:

Speak: PAT can produce a verbal utterance directed at a student or the whole team. PAT has a short range of utterances, all generated from text templates, ranging from a simple greetings like “Hello” or agreements “OK" or “no" to several basic descriptions of domain actions and goals.

Give tutorial feedback: To provide tutorial feedback on a student's action, PAT indicates a student error by shaking his head as he says “no” (see Figure 2) and he indicates a correct action by simply looking at the student and nodding. The motivation for shaking the head is to complement and reinforce the verbal evaluation, and the motivation for the head nod is to provide the least obtrusive possible feedback to the student.

Manipulate an object: To demonstrate domain task steps, PAT can manipulate objects in a variety of ways. Currently, this includes manipulations that can be done by grasping the object (e.g., moving, pulling, inserting, editing, deleting) or pointing at that object (using the Telepointer [Mar04b]) to guide the student's attention.

Check the status of the whiteboard: PAT can also demonstrate domain task steps that simply require checking an object (e.g., checking the control unit of a system or a simple checking whether terminator is connected or not).

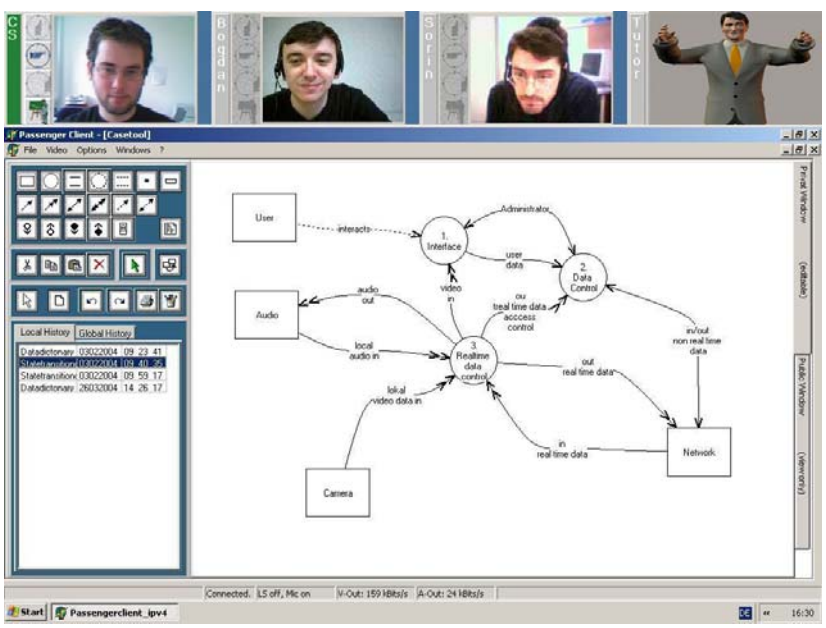

Figure 8-4 Passenger Agent Tutor

Point at an object: To draw a student's attention to an object, or connect a verbal referring expression to the object it denotes, PAT can point at the object using the Telepointer.

Offer turn: Since our goal is to make PAT's demonstrations interactive, we allow students to interrupt with questions ("What next?" and "Why?"). PAT explicitly offers the Floor Control to them after each demonstration act or when nobody is performing any action in order to achieve the learning goal. PAT's necessary skills to perform actions like this are: to monitor the owners of the Passenger Floor Control PFC, to control the entries in the PFC-list and to grant the Floor to the inactive users. Also PAT must assure a fair distribution of PFC among participants in a learning session using Passenger. A detailed description of PFC and PFC list can be found in [Mar04b].

Acknowledge an utterance: When a student or team-mate performs an action PAT can choose to explicitly acknowledge his understanding of their utterance by looking at them and nodding (see Figure 8-4).

Attend to action: When someone other than PAT manipulates an object of the whiteboard, PAT automatically shifts his gaze to the object to indicate his awareness. Unlike all the above behaviours, which are chosen deliberately by the cognition module, this behaviour is a sort of knee-jerk reaction invoked directly by the perception module.

The roles of the agent – tutor (see Figure 8-4) within the Passenger groupware are:

- *Selects a model (topic) for session/discussion*: it is through this negotiation of meaning and understanding that learning occurs. Therefore each topic has a tree structure, with nodes that are: first question for the participants, possible answers by participants, agent response to each of these answers. Topics are designed to attract participants into an interactive dialogue and to avoid the "silence" during a Passenger session.

- *Assigns roles to the students*: during the semester the student-teams will experience the entire life-cycle of Software Engineering. The students start with a requirement analysis following the Ward & Mellor approach [War85] during the modelling phase. The given problem for the practical training is chosen in such a way, that it cannot be solved by one student on its own. Therefore, each topic is divided in sub-topics which can be assigned by the tutor agent to one of the participants.

- *Provides help for toolbox buttons*: each tutor-agent is able to provide students basic help regarding the usage of the Passenger Client. Within the tutor-agent's architecture there is implemented a pattern recognition algorithm. Using this algorithm the agent can provide adequate answer to students' questions like: "<u>How</u> (1) can I <u>draw</u> (2) a <u>control transformation</u> (3)?" where (1)(2) and (3) ~How… draw… control transformation…~ represent a pattern example. After recognizing a pattern the agent will search its knowledge database for a proper answer and will provide this answer to the student. For this example the answer is "You should press the third button of the Case-Tool buttons from the first row, and then go with the mouse in your working area and click where you want to have a control transformation… ”

- *Supports and gives hints on awareness functions*: the tutor agent should be able to provide to participants proper feedback on awareness issues like: “Why can’t we see Jack? (Answer: Jack should press F3 or select send video from Video, or maybe Jack does not have a video-camera)”
- *Controls and gives hints on floor control mechanism or selects floor passing method* (adaptive): the agent can provide answer to questions like: “Why my colleagues cannot hear me? (Answer: you must be the actual floor holder in order that the others can hear you, therefore you should request the rights. There is a button on …)” or it should be able to avoid the deadlock situations like: one student which is the floor holder leaves the session but she/he forgets to pass the floor, therefore the other participants cannot modify the common artefact or they cannot communicate. One of the remaining participants can ask the floor from the agent-tutor. The agent can notice that the actual floor holder is inactive (e.g. he hasn’t made any changes to the common document for more than 10 minutes). Therefore the agent has the ability to take the floor from the inactive participant and to give it to the one that has requested for it.
- *Gives hints for next steps in modelling*: during a session it can occur that the students might reach a deadlock- the students do not know how to continue their work to fulfil their common task. The agent should be able to analyze the current state of the students’ work and to provide hint for the next steps. If the agent cannot accomplish this task then it should communicate with other tutor-agents from another Passenger sessions. If the other agents cannot provide a proper answer then the analyze evaluation should be communicated to a human-tutor. The human-tutor if he is available can replace that agent within its session or he can provide the agent the adequate answer.
- *Group Manager* - has the ability to control the coherence of the group. The necessary agents’ skills (requirements) for this role are: to monitor the owners of the Passenger Floor Control PFC, to control the entries in the PFC-list and to grant the Floor to the inactive users. Also a requirement of this role is to assure a fair distribution of PFC among participants in a learning session using Passenger. A detailed description of PFC and PFC list can be found in work of Marin *et al.* [Mar04b]. This role tries to solve one of the open problems in the collaborative virtual environments: communication issues among participants.

8.3.2. Mobile Agent

The mobile agents take the jobs from tutoring agents and travel around in the network. They are created for some particular tasks. The have clear goals orientations, and migrate to other hosts in the network just for that. They are intelligent entities, and they make decisions by themselves: according to the autonomy of mobile agents, the hosts to migrate to should be decided by mobile agents themselves.

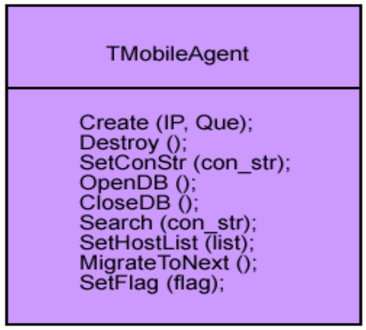

Figure 8-5 Mobile Agent UML Diagram

Figure 4 illustrates main functions or procedures of mobile agent class, which are mostly concerning about migrating and searching for information.

Create(IP,Que) is the constructor of TMobileAgent class. The IP and Que parameters are passed from tutoring agents. To the contrary of the constructor, Destroy() is the destructor of TMobileAgent. It is just used to dispose mobile agents after they completed their task. SetHostList(list) is update list received from the coordinator agent after every visit on a host. SetFlag(flag) is to set a flag to highlight the attribute visited on hosts where the mobile agent had visited. MigrateToNext() is the function that the mobile agent uses to choose the host as the next destination.

In order to realize the mobile agent's itinerary we implemented the following algorithm:

```
get_hosts_list(CAG);
search_hosts(n, P_j, A_k)
  begin
for i= 0 to n-1, i≠k do
 if KL.A_k > KL.A_i then jump to  next host;
   else if ask_help (A_i, P_j)=NULL then jump to next host;
       else return(ask_help (A_i, P_j)); break;
return(NULL);
end  search_hosts;
if search_hosts(n, P_j, A_k)=NULL then ask_human(P_j);
```

where CAG represents the coordinator agent, n is the number of hosts (we define as a host a current running Passenger learning session), A_k the tutor agent which needs help, P_j the unknown problem/ pattern, and $KL.A_k$ the knowledge level of A_k.

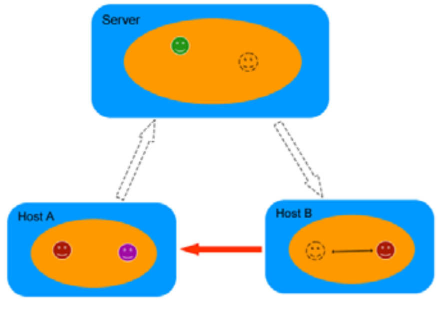

Figure 8-6. How the mobile agent travels for answers

In case that agent A_k needs help for the unknown pattern P_j, it creates a mobile agent which connects to coordinator agent and gets the list of the hosts. After this operation is successful it starts migrating from a host to another to get an answer for pattern P_j. For each agent we defined (see [13]) as a knowledge level the binary relationship between very two different topics T_i and T_j called *Precedence*(T_i,T_j) which also highlights the fact that T_i is a prerequisite for T_j, where $i<j$. Thus if between two tutoring-agent A_k and A_j the following statement $KL.A_k > KL.A_i$ is true that means that A_i is teaching a topic T_i, which is a prerequisite for T_k, topic of A_k. In other words, A_i cannot provide answers to A_k, therefore the mobile agent has to move to the next host. In case that the mobile agent cannot find adequate help from all of the tutor-agents then it has to communicate with a human-tutor and get from him/her proper answer. Figure 8-6 illustrates how the algorithm presented works on a simple example.

8.3.3. Coordinator Agent

The coordinator agent exists on the server-side, coordinating all tutoring and mobile agents working in reasonable orders in case of chaos. Its work is to receive incoming mobile agents, supply to mobile agents the host list in the network, and send them to their destinations. However, a good coordinator agent should also be able to give the suggestions about route to mobile agents. Coordinator agents are always watching the status of the whole network, and they can offer the best path for mobile agents' migrating. Also it is responsible for starting a new tutor-agent for every new session started.

The coordinator agent is used to record the statuses of the hosts and supply the host list to mobile agents. The class of the coordinator agent, TCoordinatorAgent, has only one variable, FHostList, which is supplied to mobile agents. Figure 8-7 illustrates the main functions and procedures of the coordinator agent.

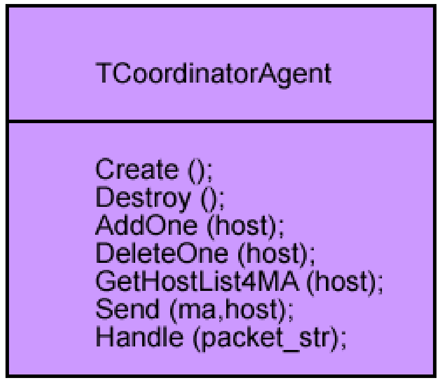

Figure 8-7 Coordinator Agent

Create() is the constructor of TCoordinatorAgent. It only initializes the variable, FHostList, and Destroy() is the destructor, to release the memory. AddOne(host) is to add one host to the host list when a client connects, and DeleteOne(host) is to delete one host from the host list when a client disconnects. Send(ma, host) has the same function as discussed in the class of TServiceAgent.

GetHostList4MA(host) is used to offer the mobile agent the host list. The only parameter is the host which the mobile agent comes from, therefore, the coordinator agent can give every mobile agent a host list which holds all the addresses of hosts in the network, except the one which the mobile comes from. This is very reasonable because it makes no sense that the mobile agent migrates to its own host to search for information.

Handle(packet_str) also needs to deserialize the mobile agent and its host list at first. And next, the coordinator agent supplies the host list to the mobile agent. If the count of the list is larger than 0, that means there are other hosts in the network, the mobile agent picks up one host and the coordinator agent sends it there, but if not, that means there is no other host available, the mobile agent sets the variable FFlag with “1” and the coordinator agent sends it back home.

8.4. Experimental Results: A Passenger Learning Scenario

The Software Engineering lab at UDE is conducted as a project setup of student teams, each consisting of four students. During the semester these teams will experience the entire life-cycle of software engineering. The students start with a requirement analysis following the Ward & Mellor [War85] approach during the modelling phase. The given problem for the practical training is chosen in such a way, that it cannot be solved by one student on its own. Therefore, the project teams have to divide up the problem amongst each other. The teams meet once a week at a certain time for two hours in a computer lab at the university. During this time, tutors are available. We first conducted a series of tests on simplified scenarios in order to prove our theory.

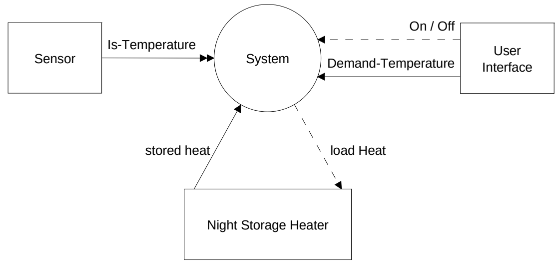

Figure 8-8. Task partial solution: Context Diagram

Here is a sample of a practical exercise: “Given is a night storage heating system. This is a heating system, which uses electrical power to store heat at night, because charge is cheaper. The system works like this: The user defines the status of the heating system: besides the decision, if the system is on or off, the user sets a temperature to the system. This temperature is compared with the current temperature stored in the night heating system and the room temperature. To measure the room temperature a separate sensor is given. Design the system according to the Ward & Mellor”.

In this exercise a night storage heating system is used as an example. Ward and Mellor introduced added some notational nuances to handle interrupts and control flows to the Data Flow Diagram; they also introduced the notion of state-transition diagrams for modelling the time-dependent behaviour of such systems.

We provide part of the solution in order to be able to explain our theory in practice. There are two different procedures for the structured system design: Top Down design and Bottom Up design.

Topic of this exercise is the design method by Ward & Mellor, a “Top Down design” which is divided into two subjects: Essential Model and Implementation Model. The Essential Model describes the demanded behaviour of the system and it is divided into two subjects: the Environment Model, and the Behavioural Model. The Environment Model has three sub-topics: Context Diagram, Information Model and Data Dictionary. The Context Diagram (see Figure8) requires two additional refinement diagrams: one for the heat control and the other for temperature control.

The Behavioural Model is the second part of the Essential Model. It describes the response behaviour from the environment of the system and it consists of State Transition Diagram, the Pre-Post-Condition and the extended Data Dictionary.

In order to successfully pass this practical training the students have to realize the Essential Model for the required system. The given credits for this task and its subtasks are as follows: choosing the right type of design method (and justifying their choice) 3 credits, 1 credit for each definition of the Essential Model, Environment Model and Behavioural Model, 15 credits for the design of the Context Diagram with the 2 additional refinements, 10 credits for the State Transition Diagram and 5 credits for each realization of the Information Model, Data Dictionary, extended Data Dictionary like also for establishing the Pre-Post-Condition. The total amount of credits is 51 and the minimum required number of credits of each student is 14 (if all the students get only their minimum credits and not more they can fulfil more than 80% of the required task which is enough to pass to the next topic). Each subtask must be accomplished in a pre-defined period of time in order to get the credits.

For instance, to check if PAT feels pity for student_1, it tests whether student_1 has not collected at least y = 14 credits AND the given period of time is not over. If student_2 has collected 18 credits and the time allocated to the task is not over PAT feels joy for him.

These emotions depend of course also on the values of the potentials and thresholds defined in the rules associated with them, which are also task dependent. Within this section we tried to provide a simplified example of our theory applied in the synchronous learning environment called: Passenger.

8.5. Evaluation

The intended evaluation study for this prototype concerns two levels:

- *Usefulness level:* the usefulness of the agent facilities within Passenger groupware needs to be evaluated by human teachers.
- *User friendliness level*: this level highlights how the agent was accepted by students.

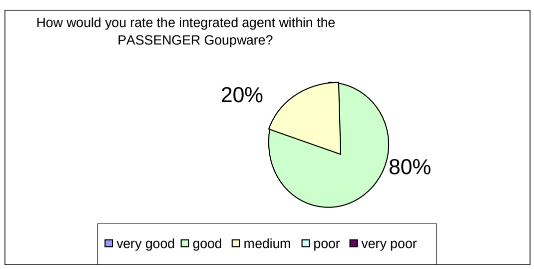

Figure 8-9. Acceptance of Agent Tutor

Several experiments took place in the local area network of our institute. Only the second part of the evaluation study was conducted among 25 first year Master-students.

Each session consisted of three students and one tutor (human or agent). The student experienced the traditional lab with the human tutor and also with the agent feature of the Passenger system. After these experiments, students had to answer to questionnaires files. A sample of questions concerning the second level that were asked to the students is the following:

1. *Do you consider the application attractive? If yes, what did you like about it?*

2. *Do you think that the "agent" features prevented you from understanding the educational process better?*
3. *Do you prefer the agent tutor instead of the human tutor? Please justify your answer.*

Based on these questionnaires several statistics could be made. Some results concerning the agent integration and acceptance are shown in the Figure 8-9.

Although the number of participants in the evaluation test was rather small for a quantitative evaluation, the trends seem to be unambiguous. It is planned to realize the full-evaluation test including an evaluation result for the first level and also to increase the number of student participants. It is also planned to extend the knowledge domains also of the emotional agents to other discipline areas and to conduct several acceptance tests among the professors.

Chapter 9: Conclusions

In this chapter, we summarise the main contributions our research makes to the ongoing task of elucidating the motivational requirements of emotional intelligent tutoring agents. We also describe some of the possible directions in which we hope our work will be extended in the future.

9.1. Main Contributions of this Research

One of the challenges faced by researchers in the construction of intelligent autonomous agents is the need to develop a systematic framework in which to answer questions about the types of control mechanisms such agents might need, and how those different control mechanisms might interact. In this work, we argued for an information-level design-based approach to the study of intelligent tutoring agents – wherein each new design gradually increases our explanatory power and allows us to account for more and more of the phenomena of interest. These broad designs help to build our understanding of the different attributes of information-level representations, their functional roles, and their causal relationships. Further, by adopting information-level descriptions, we are able to offer a rich explanatory framework for exploring human-like mental states in terms of the information-processing and control functions of the underlying architecture.

This thesis makes a number of contributions to the field of designing pedagogical agents. A briefly overview of these contributions is given below in chronological order –as they appear in this thesis.

A Systematic Method for analysing and designing Pedagogical Agent

1) Specifically, we: (a) present our design-based research methodology, and describe how it can be used to provide a powerful explanatory framework for elucidating complex systems such as intelligent pedagogical agents (b) describe how viewing the human mind as a complex control system allows the use of certain mentalistic terms and concepts to be justified by referring them to information-level descriptions of the underlying architecture (c) introduce the concept of motivational control states and describe the *functional* attributes of some of the many control states that are likely to play an important role in intelligent autonomous agency architecture, and finally (d)

describe a cognitively inspired architectural framework for elucidating the *emotional*, and *functional* attributes of these control states.

2) We argue for a concern-centric stance to autonomous agent design and provide a design-based analysis of *motivational* control states in both deliberative and behaviour-based agent methodologies – Chapter 3. We identify a number of problems with these designs and we address these problems with our design for an intelligent autonomous agent in Chapter 4-8.

3) We provide a framework for analysis of emotional agent designs based on levels of social intelligence Chapter 5. We extended this framework by adding a new level: The Cultural Background. We identified this level as a need for believability in pedagogical agents and not only.

4) We provided a metamodel of a collaborative environment for a better understanding of role requirements for tutoring agents performing activities in such environments. (Chapter 4). Chapter 5 highlights how to model these requirements using AORML for a better analysis.

5) The human emotion process can be viewed as a classic example of Dynamic Networks Beliefs. Chapter 6 adds depth to the motivated agent framework by making explicit the emotions triggering and reasoning process.

6) Finally, we present our abstract design for a motivational pedagogical agent Chapter 7 – built on the lessons learnt from Chapters 3 through Chapter 6.

An emotional model for a Pedagogical Agent

We have proposed an emotional agent model. The notion of emotion was developed along three axes: emotions, feelings and personality. The distinction of these three axes based on the duration of the evolution of each dimension is the keystone of our model. More important than the emotions, feelings and temperament parameters proposed, the idea of splitting the emotional aspect into multiple dimensions can be a good basis to develop convincing virtual humans. Then, in our model, the psychological aspect plays a role at different levels of the control process: cognitive, perceptive and expressive. It is an important stake to be able to simulate all the aspects of emotions on the behaviour.

In the short term, our objective is to complete psychological appraisal to enhance the simulation and, consequently, the complexity of potential behaviours. Then, we will have to study emergent behaviours and compare them to real-life behaviours already observed. Then, it will be interesting to extend the study to non-brain-damaged people. We have built a model that can take into account the specifities of brain-damaged people, but that is also sufficiently generic to start applying the same kind of approach to non brain-damaged people. In a second time, it would also be interesting to adopt a methodology to evaluate our emotional model as done in.

The BDI framework has been used with considerable success for human modelling, but was not designed specifically for this purpose. A framework that included more of the basic characteristics of human behaviour would be a more powerful modelling tool, because the model builders could the focus on the domain-specific aspects of the model being built, reducing their overall workload. However the BDI framework does have one particular strength for human modelling — its folk psychological roots — which in itself facilitates the model builders' task, because the subjects' knowledge maps easily to the constructs used in the framework. This paper has presented an approach to developing a framework that represents more of the basic human characteristics, but maintains this conceptual simplicity.

The approach described is an incremental one, successively adding support for additional characteristics, based upon folk psychological models of these characteristics. As was demonstrated with the implemented example, even relatively simple folk psychological explanations *may* add 'unnatural' parameters to the framework, and this needs to be taken into account when evaluating the advantages of any addition. It is also noted that folk psychological explanations of different characteristics sometimes use variations in the basic concepts, and the mapping between concepts should be carefully considered.

There are many human characteristics that could be incorporated into the framework using this approach, but some aspects of human behaviour and reasoning do not lend themselves to folk psychological explanations, and this sets the bounds of the approach. If these types of characteristics are to be included in the framework, they must be incorporated in some other way, as with the models of perception and action described in Chapter 5, Chapter 6.

This approach to extending the BDI framework should produce a framework that provides better support for human modelling, by maintaining the strength of the BDI architecture, but addressing some of its weaknesses. While it will not be possible to add models of *all* basic human characteristics using this approach, many of those that are seen as important in human modelling can be handled this way. A complementary approach can address those characteristics that do not have folk psychological explanations.

The advantages of our formalism are in the possibility of representing the following aspects of emotion elicitation:

1. *temperament and personality:* how these factors affect the Agent's propensity to feel and to show emotions, by influencing the weights it assigns to achieving its goals, the threshold for emotion activation and the resistance to change its emotional state;
2. *social context:* how the Agent's relationship with the context influences emotion triggering by influencing its beliefs and goals;
3. *dynamics of the Agent's state:* how the Agent's affective state changes over time, either as a consequence of 'endogenous' or of 'exogenous' factors;

A detailed architecture for elucidating the emotional and pedagogical requirements of our prototype

The Pedagogical Agent's Architecture integrated to the Student Emotional Model allows a better customisation of an ITS, for it considers cognitive and emotional issues increasing the possibility of truly motivating the student.

The primitive variables, captured through the communication channels between the tutoring system and the student, allow the formation of a gradual behavioural inference chain, yielding the behaviour and temperament structures for each student.

The knowledge domain and the emotional and motivational strategies compose the expert basis of the system, captured through specific authoring tools.

The Motivational Module completes the proposal, operating synchronously, to attend the momentary needs of the student, and to prescribe immediate and long term actions, based on the emotional and cognitive data.

A neuro-fuzzy model was proposed for assessing students' motivational state. A main advantage of the proposed approach is that teachers' knowledge can be elicited in linguistic form and encoded in the system. Preliminary results were encouraging, since the results of the experiment which was carried out with the assistance of experts, provided us with essential knowledge to define from experts suggestions the inputs and outputs of our model in order to evaluate student's motivational state. We are currently generating simulated students cases from the real student log files in order to be classified from the group of experts and used to train the neural networks and test our approach

9.2. Future Work

The risk of building inconsistent agents is always high. However, this risk increases considerably when adaptation to the context is included as one of the requirements of the agent's behaviour. As anticipated before, consistency in the agent's behaviour is a core aspect of believability. It is therefore important to find an answer to the following question: How may designers ensure that inconsistency is not introduced in the different phases of adaptation and in the different aspects of the agent's behaviour? For instance, how may we be sure that the agent's mental model is internally consistent? We need a psychologically grounded theory to set up the weights given to the various goals and the strengths of relations between beliefs and goals so that they correspond to a psychologically and socially plausible individual or category of individuals, like a cultural community. How may we ensure that the external behaviour of the agent is consistent, coherent across similar situations in time, and corresponds to the agent's system of norms, values, beliefs, and goals, which shape its mental model? The idea of linguists is that every message is produced to achieve a communicative goal and brings a meaning in it. When the message includes verbal and nonverbal parts, the two parts may reinforce each other or only one of them may be employed to convey a given meaning. Thus, a consistent message is one that does not include contradictions either in the same component or among different components.

Let us make some (extreme) examples of inconsistency in the two examples described in this chapter. In the information provision scenario, it would be rather implausible to attach to the same agent a highly cooperative and highly dominant attitude (although this might happen in some individuals). This would correspond to the case of a person who proactively offers help to someone who is not requesting it explicitly, but at the same time demands that his or her suggestion is followed immediately and literally [Cas98]. For the same reason, it would be inconsistent to attach an extroverted behaviour to a non-cooperative person because gestures whose meaning is generically that of invading the territory of others are unlikely in people who do not pay attention to others' viewpoints or needs.

In a learning scenario, it would be inconsistent to show empathy in a part of the message "*I'm sorry to inform you . . .*" and smile at the same time, or say it with a neutral face (although the degree of inconsistency is much lower in this case), or to not synchronize a smile with the verbal part of the expression of empathy.

These are only examples that do not provide any general solution to the problem. As far as affective factor representation is concerned, Ortony [Ort03] suggested looking at theories like the Five Factor Model [McC92] to drive the design process toward consistency. In this theory, personality traits tend to aggregate into a few factors: A consistent personality would therefore correspond to an aggregate of personality traits in the five-dimensional space. Hofstede's Five Dimensions of Culture [Hof80] might play the same role in building culturally consistent characters. It might help, for instance, in assigning consistent values to parameters that are associated with short-term versus long-term orientation, femininity versus masculinity, power-distance, collectivism versus individualism, and uncertainty avoidance. However, this is still an open problem that should be investigated more in depth.

The more general problem of our Agent's *believability* merits a particular comment. In their discussion of how Embodied Conversational Agents should be evaluated, Nass and colleagues [Nas00] mention consistency between verbal and nonverbal cues as a potentially important factor of user satisfaction and demonstrate its impact on several measures of preference [Nas00a]. Consistency is, to several authors, a key factor of believability (see, for instance, Ortony, [Ort03] and inconsistency is an impending danger for Embodied Conversational Agents, that may arise from several mistakes in their design: the content of a move may be inconsistent with its expression, verbal and nonverbal components of communication may be inconsistent, synchronisation of these components may be imperfect, personality traits may be combined in a psychologically implausible way, and so on. No one knows which of these factors merits priority in the design of an ECA.

As far as emotion activation is concerned, in particular, our main concern is to be sure that weights are assigned to goals so as to produce consistent personalities. Ortony

[Ort03] suggests applying, to this purpose, theories about clustering of personality traits such as the Big Five model [McC92]: although this is a substantial contribution to the definition of an Agent's personality as a combination of consistent traits, it still leaves open, to us, the problem of association between traits and goal weights.
To assess how 'believable' the Agent we developed is, we are designing a set of in-depth evaluation studies, there is a plan to release the prototype free. Also there is plan to conduct an evaluation study for this prototype. This study should concern two levels: *the usefulness level*- which should highlight the acceptance of the system by teachers from all over the world not only from local university and the *user friendliness level:* should highlight how the system was accepted by students, students preferences after using the system whether they would like to learn with our system instead of traditional learning

9.3. Final Remarks

Within this thesis it was shown how to corroborate affective theory with role theory with agent technology in a synchronous virtual environment in order to overcome several inconveniences of distance education systems. The aim of this research is to provide the first steps to define a method for creating a tutor agent which can partially replace human-teachers and assist the students in the process of learning.
Our tutor agent tries to replace partially the human teacher, in assisting the students at any time of their convenience and in the meantime the agent can evaluate the results of the students' activity during the learning process

The main contribution of this work in the *"Agents in CSCW"* research field is conception of a Distance Learning System in which humans and artificial agents can collaborate to achieve a common learning goal. In terms of the geographical distribution of the participants, which is one of the most publicized advantages of the Web-based education environments, there is much to gain through the use of the agent paradigm.

We try to shed upon a light on the lack of tutor problem in synchronous virtual collaborative environments and we applied our theories to practice on a synchronous groupware called Passenger. The goal of the research presented in this paper is to develop a software agent – tutor to assist students in the collaborative learning process at any time of their convenience and also to facilitate the tutor's task of perception of students' collaborative learning process.

Nevertheless, it would be a mistake to conclude that current and forthcoming developments can make intelligent tutor-agents to replace human teachers. Given the fact that learning is a complex cognitive activity, learners cannot rely solely on machines when capturing and mastering knowledge of a certain domain. Still tutoring agents are useful tools in the sense that in many cases improve the learning effectiveness.

References

A

[Alc04] Alchieri, J. C. *Models of Millon personality styles: An adaptation of Millon personality stiles*, PhD Thesis. Federal University of Rio Grande do Sul, Porto Alegre, 2004

[All02] Ally, M., & Fahy, P. Using students' learning styles to provide support in distance education. Paper published in *Proceedings of the 18th Annual Conference on Distance Teaching and Learning*, Madison, Wisconsin, August, 2002.

[And85] Anderson, J. R., Boyle, C. F., & Reiser, B. J. Intelligent tutoring systems. *Science*, *228*, 4698, pp.456–462, 1985.

[And87] Andersen, S. & Klatzky, R. "Traits and social stereotypes: Levels of categorization in person perception," *J. Personality Soc. Psychol.*, vol. 53, pp. 235–246, 1987.

[And97] Andersen, E., Conceptual Modelling of Objects: a role modelling approach. PhD Thesis, Dept. of Computer Science, University of Oslo, Oslo, 1997.

[And89] Anderson, J.R., Conrad, F.G., & Corbett, A.T. Skill acquisition and the LISP tutor, Cognitive Science, 13, p. 467-505, 1989.

[Ant99] Antunes, P., & Ho, T. Facilitation tool—A tool to assist facilitators managing group decision support systems. In *Proceedings of the Ninth Workshop on Information Technologies and Systems (WITS '99).* Charlotte, North Carolina, 1999.

[Aro99] Aroyo, L., & Kommers, P. *Preface - Intelligent Agents for Educational Computer-Aided Systems.* Journal of Interactive Learning Research, 10 (3/4), p.235-242, 1999.

[Atk83] Atkinson, R.L., Atkinson, R.C. & Hilgard, E.R. "*Introduction to Psychology*", Harcourt Brace Jovanovich Inc. 1983.

[Aug95] Augoustinos, M. & Walker, I. *Social Cognition: An Integrated Introduction.* London, U.K.: Sage, pp. 39, 1995.

B

[Bal50] Bales, R.F., Interaction process analysis: a method for the study of small groups, Addison Wesley 1950.

[Ban77] Bandura, A. "*Social Learning Theory*", Prentice-Hall, Englewood Cliffs, 1977.

[Bar76] Barr, A., Beard, M. & Atkinson, R.C. The computer as tutorial laboratory: the Stanford BIP project, International Journal on the Man-Machine Studies, 8(5), pp.567-596, 1976.

[Bar95] Baron-Cohen, S. Mind blindness: An Essay on Autism and Theory of Mind. Cambridge, MA: MIT Press, 1995

[Bat94] Bates, J., *The Role of Emotion in Believable Agents,* Communications of the ACM, 37(7), p.122-125, 1994.

[Bee90] Beer, R.D., Chiel, H.J., & Sterling, L.S., "A biological perspective on autonomous agents design", In *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back,* Elsevier Science Publishers, 1990.

[Ben99] Bennett, F., *Computers as Tutors: Solving the Crisis in Education*, Sarasota, FL: Faben Inc. Publishers, 1999.

[Ber00] Bergenti, F. & Poggi, A. "An agent-based approach to manage negotiation protocols in flexible CSCW systems", Proceedings of the 4th International Conference on Autonomous Agents, Barcelona, June 2000.

[Ber87] Berry, D. C. The problem of implicit knowledge. *Expert Systems*, August, 4(3), pp.144–151, 1987.

[Ber98] Berscheid, E., & Reis, H., *Attraction and Close Relationships*. Handbook of Social Psychology, McGraw-Hill, New York, p.193-281, 1998.

[Bid79] Biddle, B.J., Role Theory- Expectations, Identities and Behaviors. Academic Press, London, 1979.

[Bid79b] Biddle, B.J. and Thomas, E.J. *Role Theory: Concepts and Research,* K.E. Kriger Publishing Company, New York, 1979.

[Bio01] Biolluz, A., & D'Halluin, C., *Usage on an mediated environment for the cooperative learning.* Les cahiers d'études du CUEEP, 43, 2001.

[Blo84] Bloom, B. The 2 sigma problem: The search for methods of group instruction as effective as one-to one tutoring, Educational Researcher, 13(6), pp. 4-16, 1984.

[Blu96] Blumberg, P.M., Old Tricks, New Dogs: Ethology and Interactive Creatures, PhD Thesis, MIT, 1996.

[Boo86] Boose, J.H. *Expertise transfer for expert system design*. Amsterdam: Elsevier, 1986.

[Bra84] Braitenberg, V. *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MA: The MIT Press, 1984.

[Bra87] Bratman, M. E. *Intentions, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press, 1987.

[Bra04] Braubach, L., Pokahr, A., Moldt, D. & Lamersdorf, W., Goal Representation for BDI Agent Systems. In Proceedings of the Second Workshop on Programming Multiagent Systems: Languages, frameworks, techniques, and tools (ProMAS04), 2004.

[Bre84] Breiman, L., et al. *Classification and regression tree*. Belmont, CA: Wadsworth International Group, 1984.

[Bri96] Briscoe, G., & Gaelli, T. *Symbolic machine learning*. Norwood, NJ: Ablex Publishing Corporation, 1996.

[Bru97] Brusilovsky, P. & Schwarz, E. User as student: towards an adaptive interface for advanced Web-based applications, in: Proceedings of 6th International Conference on User Modeling, Sardinia, Italy, pp. 177-188, 1997.

[Bru99] Brusilovsky, P. Adaptive and intelligent technologies for Web-based education, Special Issue on Intelligent Systems and Tele-teaching, 4, p.19-25, 1999.

[Bru02] Brusilovsky, P., & Maybury, M.T., From adaptive hypermedia to adaptive Web, Communications of the ACM, Special Issue on the Adaptive Web, 45(5), p.31-33, 2002.

[Bur88] Burns, H.L., & Capps, C.G., Foundations of intelligent tutoring systems, Foundations of Intelligent Tutoring Systems, Lawrence Erlbaum Associates, ch. 3, p.55-78, 1988.

C

[Cab04] Cabri, G., Ferrari, L. & Leonardi, L. "Rethinking Agent Roles: Extending the Role Definition in the BRAIN Framework" *Proceedings of IEEE Int. Conf. on Systems, Man and Cybernetics*, ISBN 0780385675, Hague, Netherlands, 2004.

[Can79] Cantor, N. & Mischel, W,"Prototypes in person perception," in *Advances in Experimental Psychology,* L. Berkowitz, Ed. New York: Academic, vol. 12, 1979.

[Can97] Canamero, L., "*Modeling motivations and emotions as a basis for intelligent behavior*", Proc. of the 1st Int. Conf. Autonom. Agents, W.L. Johnson Ed., NY, 1997.

[Car70] Carbonell, J. R. AI in CAI: An Artificial Intelligence Approach to Computer Assisted Instruction. IEEE Transactions on Man Machine Systems, v.11, n.4, p.190–202, 1970.

[Car80] Carbonell, J.G. Towards a process model of human personality traits. *Artificial Intelligence*, 15, pp.49-74, 1980

[Car92] Carey, R. & Strauss, P. "An Object –Oriented 3D Graphic Toolkit", Proc. of the ACM Computer Graphics Conference, pp. 341-349, 1992.

[Car99] Carley, K.M., & Gasser, L., "*Computational Organizational Theory" in Chapter 7 "Computational Organization Theory*" by KM Carley & L Gasser in "Multiagent Systems" Gerhard Weiss (Ed) MIT Press, 1999.

[Car96] Carstensen, P., & Sørensen, C., From the social to the systematic? An analysis of mechanisms supporting coordination work in design. *CSCW Journal, 5*(4), 384–413, 1996.

[Car00] Carvalho, M., Generating Intelligent Tutoring Systems for Teaching Reading: Combining Phonological Awareness and Thematic Approaches. 2000

[Car06] Carofiglio, V., de Rosis, F. & Grassano, R. Dynamic models of mixed emotion activation. In L Canamero & R.Aylett (eds.): *Animating Expressive Characters for Social Interactions.* Amsterdam: John Benjamins, 2006 (in press)

[Cas98] Castelfranchi, C., de Rosis, F., Falcone, R., & Pizzutilo, S. "Personality traits and social attitudes in multiagent cooperation." *Applied Artificial Intelligence, 12,* 7–8, 1998.

[Cas00] Cassell, J. "Nudge nudge wink wink: Elements of face-toface conversation for embodied conversational agents." In J. Cassell, J. Sullivan, S. Prevost, & E. Churchill, eds, *Embodied Conversational Agents*, pp. 1–27. The MIT Press 2000.

[Cas00b] Castelfranchi, C. Affective Appraisal Versus Cognitive Evaluation in Social Emotions and Interactions. In A. Paiva (Ed.). *Affective Interactions.* Springer LNAI 1814, pp. 76-106, 2000.

[Cas02] Castro, J., Kolp, M. & Mylopoulos, J. Towards requirements-driven information systems engineering: the TROPOS project. *Information Systems*, 27, pp.365–389, 2002.

[Chi01] Chi, M.T.H., Siller, S.A. et al. “Learning from Human Tutoring”, *Cognitive Science 25*(4) , pp. 471-533, 2001.

[Chi89] Chilberg, J. C., A review of group process designs for facilitating communication in problem-solving groups. *Management Communication Quarterly, 3*(1), 51–71, 1989.

[Cla01] Clarebout, G., Ellen, J., Johnson, W.L. & Shaw, E. “Animated Pedagogical Agents: An Opportunity to be Grasped?” *Journal of Educational Multimedia and Hypermedia*, vol. 11, pp.267-286, 2001.

[Con95] Conte, R. & Castelfranchi, C. *Cognitive and Social Action*. London : University College, 1995.

[Con02] Conati, C., Probabilistic assessment of user's emotions in educational games, Applied Artificial Intelligence, vol.16, 555-575, 2002.

[Con02a] Conati, C., Gertner , A. & van Lehn, K. Using Bayesian networks to manage uncertainty in student modeling, User Modeling and User-Adapted Interaction, 12(4), pp.371-417, 2002.

[Cor99] Corbett, A., Anderson, J., Graesser, A., Koedinger, K., & VanLehn, K. Third generation computer tutors: Learn from or ignore human tutors. *CHI ’99*, May, 1999.

[Cri00] Cristea, A. I., & Okamoto, T. Student model-based, agent-managed, adaptive distance learning environment for academic English teaching. In *Proc. International Workshop on Advanced Learning Technologies*, 2000.

D

[Das03] Dastani, M., van Riemsdijk, B., Dignum, F., & Meyer, J.-J., A Programming Language for Cognitive Agents: Goal Directed 3APL. In *Proceedings of the First Workshop on Programming Multiagent Systems: Languages, frameworks, techniques, and tools (ProMAS03),* 2003.

[Das04] Dastani, M. & van der Torre, L., Programming BOID Agents: a deliberation language for conflicts between mental attitudes and plans. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS’04),* 2004.

[Dec01] De Carolis, B., Pelachaud, C., Poggi, I. & de Rosis, F. Behavior planning for a reflexive agent. In *Proceedings of IJCAI'01.* Seattle, August 2001.

[Ded86] Dede, C. A review and synthesis of recent research in intelligent computer-assisted instruction. *International Journal of Man–Machine Studies*, 24, pp.329–353, 1986.

[DeL99] De Loach S.-A. Multiagent Systems Engineering A Methodology and Language for Designing Agent Systems. In *Proc. of Agent Oriented Information Systems,* pp. 45–57, 1999.

[DeL00] De Loach, S.-A. & Wood, M.-F. An Overview of the Multiagent Systems Engineering Methodology. The First International Workshop on Agent-Oriented Software Engineering (AOSE-2000), 2000.

[Den78] Dennett, D. C. *Brainstorms: Philosophical Essays on Mind and Psychology*. Cambridge, MA: The MIT Press, 1978.

[Den87] Dennett, D. C. *The Intentional Stance*. Cambridge, MA: The MIT Press, 1987.

[Den91] Dennett, D. C. *Consciousness Explained*. London: Penguin, 1991.

[Den95] Dennett, D. C. *Darwin's Dangerous Idea: Evolution and The Meaning of Life*. London: Allen Lane – The Penguin Press, 1995.

[Den96] Dennett, D. C. *Kinds of Minds: Towards an Understanding of Consciousness*. London: Weidenfeld and Nicolson, 1996.

[Deni03] Denis, B., *Which tutors' roles are intervening in creation of distance-education devices?* Distances et saviors, 1(1), 19-46, 2003.

[Dep01] Depke, R, Hcckel, R.& Kuster, J.M. Improving the Agent-oriented Modeling Process by Roles, in Proceedings of Autonomous Agents'01 (Montreal, Canada), ACM Press, 2001.

[DeR99] De Rosis, F., De Carolis, B., & Pizzulito, B., *Software documentation with animated agents,* Proc. of the 5th ERCIM Workshop on User Interfaces For All, 1999.

[Dig03] Dignum, V. *A Model for Organizational Interaction, based on Agents, founded in Logic*. PhD thesis, University of Utrecht, 2003.

[Div96] Divitini, M., Simone, C., & Schmidt, K., ABACO: Coordination mechanisms in a multi-agent perspective. Paper presented at the *Second International Conference on the Design of Cooperative Systems*, Antibes-Juan-les-Pins, France, 1996.

[Dom99] H.-P. Dommel and J.J. Garcia-Luna Aceves, "Group coordination support for synchronous Internet collaboration", *IEEE Internet Computing*, pp.74-80, 1999.

[Don01] Donath, J. Mediated faces. In M. Beynon, C. L. Nehaniv, & K. Dautenhahn (Eds.), *Cognitive technology: Instruments of mind. Proceedings of the 4th International Conference* (CI 2001), Warwick, United Kingdom., 2001

[Dou92] P. Dourish and V. Bellotti, "Awareness and coordination in shared workspaces", In J. Turnier and R. Kraut, (Eds.) *Proc. Of CSCW'92- Sharing Perspectives*, ACM Press Toronto Canada, pp. 107-114, 1992.

E

[Eis86] Eisenberg, N., *Altruistic Emotion, Cognition, and Behavior.* Hillsdale, NJ: L. Erlbaum, 1986.

[Efr83] Efron, B. "Estimating the error rate of a prediction rule: Improvement on cross- validation", *J. American Statistics Association*, 78, pp.316–331, 1983.

[Ekd99] Ekdal, B., & Davidsson, P.: *"A workable definition of computerized agents"*, 3rd World Multiconference on Systemic, Cybernetics and Informatics, Florida, USA 1999.

[Ekm 69] Ekman, P. & Friesen, W.V., "*The repertoire of nonverbal behavior: Categories, origins, usage, and coding*". Semiotica, vol. 1, pp.49–98, 1969.

[Ekm82] Ekman, P. *Emotion in the human face*. Cambridge: Cambridge University Press, 1982.

[Ekm92] Ekman, P. "An argument for basic emotions", Cognition and Emotion, 6(3-4), pp. 169-200, 1992.

[Ell91] Ellis, C.A., Gibbs, S.J. & Rein, G.L. "Groupware: some issues and experiences". In *Communications of the ACM*, Vol. 34, No I, pp.339-407, 1991.

[Ell92] Elliott, C. "*The Affective Reasoner. A process model of emotions in a multi-agent system.*" PhD. Thesis, Institute for the Learning Sciences, Northwestern University, 1992.

[Ell94] Ellis C. & Wainer J. A., Conceptual Model of Groupware, in *Proceedings CSCW'94*, ACM Conference on Computer Supported Cooperative Work, Furuta, R., Neuwirth, C. eds., p.79-88, 1994.

[Eng99] Engström, Y. Activity theory and individual and social transformation. In Engström et al. (Eds.), *Perspectives on activity theory*. London; New York: Cambridge University Press, 1999.

[Est02] Esteva, M., de la Cruz, D. & Sierra, C. ISLANDER: an electronic institutions editor. In *1st International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'02)*, pp.1045 – 1052. ACM Press, 2002.

F

[Fer91] Ferber, J. & Gasser, L. "*L'Inteligence Artificielle e Distribuée*", International Workshop on Expert Systems and their Applications, Avignon, France 1991.

[Fer91b] Ferber, J. & Gutknecht, O. A recta-model for the analysis and design of organizations of Multi-Agent systems, in Proceedings of ICMAS'91q (Paris, France), IEEE Press, 1991.

[Fis98] Fischer, G., & Scharff, E., "Learning technologies in support of self-directed learning", *Journal of Interactive Media in Education*, vol. 4, 1998.

[Fju01] Fjuk, A. "Complexity of distributed collaborative learning: unit of analysis" Proceedings of CSCL2001, Maastricht-Netherlands, March 2001.

[Fri86] Frijda, N. H. *The Emotions*, Cambridge: Cambridge University Press, 1986.

[Fug98] Fuggetta, A., Picco, G. P., & Vigna, G. Understanding code mobility. *IEEE Transactions on Software Engineering*, *24*(5), pp.342–361, 1998.

G

[Gen94] Genesereth, M.R. & Ketchpel, S.P. "Software agents". Communication of the ACM, 37(7): pp. 48–53, 1994.

[Geo03] Georgopoulos V., Malandraki G., & Stylios Ch., "A Fuzzy Cognitive Map Approach to Differential Diagnosis of Specific Language Impairment", *Artificial Intelligence in Medicine*, 29,pp. 261-278, 2003

[Ger98] Gertner, A., Conati, C. & van Lehn, K. Procedural help in Andes: generating hints using a Bayesian network student model, in: Proc. of 15th National Conference on Artificial Intelligence, Madison, Wisconsin, pp. 106-111, 1998.

[Gid97] Giddens, A. *Sociology*, Polity Press, 3rd edition, ISBN: 0745618030, pp.585, London, UK, 1997.

[Gil97] Gilbert, D. "*Intelligent Agents: The Right Information at the Right Time*." IBM White Paper, May 1997.

[Gla98] Glaser, R., Chi, M.T.H., and Farr, M.J. *The Nature of Expertise*, Erlbaum, Hillsdale, New Jersey, 1988

[God99] Goddard, T. & Sunderam, V.S. "ToolSpace: Web Based 3D Collaboration", Proc. 4th Symposium on the Virtual Reality Modeling Language, Germany, pp.161-165, 1999.

[Gol95] Goleman, D. Inteligência Emocional: A Teoria Revolucionária que Define o que é ser Inteligente. Bockman, Rio de Janeiro. 1995.

[Gui02] Guizzardi, R.S.S., Wagner, G. & Aroyo, L. "Agent-oriented modeling for collaborative learning environments: a peer-to-peer helpdesk case study", XIII-Brazilian Symposium on Computers in Education, November, 2002.

H

[Har93] Harris, P. "Understanding emotion," in *Handbook of Emotions*, M. Lewis *et al.*, Eds. New York: Guilford, pp. 27–46, 1993.

[Haz98] Hazemi, R., Hailes, S., & Wilbur, S., *The digital university: reinventing the academy*, Springer-Verlag, UK, 1998.

[Haw96] Hawryszkiewycz, I.T., "Support Services For Business Networking", in *Proceedings IFIP96*, Canberra, eds. E. Altman and N. Terashima, Chapman and Hall, London, 1996.

[Haw97] Hawryszkiewycz, I.T., "A Framework for Strategic Planning for Communications Support" *Proceedings of The inaugural Conference of Informatics in Multinational Enterprises*, Washington, pp. 141-151, 1997.

[Haw02] Hawryszkiewycz, I.T., "Designing Collaborative Business Systems" in IFIP 17th World Computer Congress, TC8 Stream on Information Systems: The e-Business Challenge, ed. Roland Traunmiller, Montreal, Kluwer Academic Publishers, Boston, pp. 131-146, 2002.

[Hen03] Hentea, M., Shea, M.-J., & Pennington, L., "A Perspective on Fulfilling the Expectations of Distance Educations", *Proc. of the 4th Int. Conf. on Information Technology Curriculum*, USA 2003, 160-67.

[Hef98] Heffernan, N. T. Intelligent tutoring systems have forgotten the tutor: Adding a cognitive model of human tutors. *CHI 98*, April (pp. 50–51), 1998.

[Her97] Hermans, B. "Intelligent Software Agents on the Internet: An Inventory of Currently Offered Functionality in the Information Society and a Prediction of (Near) Future Developments," *First Monday*, Vol. 2, no. 3, 1–23, 1997.

[Hir89] Hirokawa, R. Y., & Gouran, D., Facilitation of group communication: A critique of prior research and an agenda for future research. *Management Communication Quarterly*, *3*(1), 71–92, 1989.

[Hof80] Hofstede, G. *Culture 's consequences: International differences in work-related values* . Beverly Hills, CA: Sage, 1980.

[Hol87] Holland, D. & Quinn, N. Eds., *Cultural Models in Language and Thought.* Cambridge, U.K.: Cambridge Univ. Press, 1987.

[Hor98] Horn, E.M. , Collier, W.G., Oxford, J.A., Bond Jr., C.F. & Dansereau, D.F. "Individual differences in dyadic cooperative learning", *Journal of Educational Psychology*, 90, pp.153-161, 1998.

[How01] Howden, N., Ronnquist, R., Hodgson, A. & Lucas, A. Jack summary of an agent infrastructure. In *5th International Conference on Autonomous Agents*. 2001.

[Hun99] Hunger, A., Werner, S., & Schwarz, F. *Measures to improve the globalization in higher education.* Proceedings of ICCE 99, Chiba, Japan, II, 803-804, 1999.

I

[Ing96] Ingrand, F., Chatila, R., Alami, R. & Robert, F. PRS: A High Level Supervision and Control Language for Autonomous Mobile Robots. *Proc. of the IEEE Int. Conf. on Robotics and Automation,* pp. 43–49, Minneapolis, April 1996.

[ITC] Instructional Technology Council ITC's Definition of Distance Education, http://www.itcnetwork.org/definition.htm

J

[Jam95] Jameson, A. Numerical uncertainty management in user and student modelling: an overview of systems and issues, User Modelling and User-Adapted Interaction, 5(3- 4), pp.193-251, 1995.

[Jen95] Jennings, N., & Wooldridge, M.,: "*Intelligent Agents: Theory and Practice*" in Knowledge Engineering Review, vol.10, no.2, 1995.

[Jen98] Jennings, N., Sycara, K., & Wooldridge, M.: A Roadmap of Agent Research and Development, Autonomous Agents and Multi-Agent Systems, 1, Kluwer Academic Publishers, pp.7-38, 1998.

[Jen00] Jennings, N.: "*On Agent-based Software Engineering, Artificial Intelligence*" 117**,** pp.277-296, 2000.

[Joh87] Johnson, M., The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason. Univ. Chicago Press, 1987.

[Joh90] Johnson, D., Johnson, R., & Holubec, E. J.: Circles of learning: Cooperation in the classroom 3rd edn. Edina, MN: Interaction Book Company, 1990.

[Joh97] Johnson, W. L., & Rickel, J. "Steve: An animated pedagogical agent for procedural training in virtual environments". *SIGART Bulletin*, *8*(1–4), pp.16–21, 1997.

[Joh98] R. Johansen, “Groupware: Computer Support for Business Teams”, *The Free Press-Macmillan*, New York, 1988.

[Joh00] Johnson, W. L., Rickel, J. W., & Lester, J. C. “Animated pedagogical agents: Face-to-face interaction in interactive learning environments”. *International Journal of AI in Education*, (11), 47–78, 2000.

[Joh01] Johnson, W.L. Pedagogical agents for Web-based learning, in: Proceedings of First Asia-Pacific Conference on Web Intelligence, Maebashi City, Japan, pp. 43-44, 2001.

K

[Kay01] Kay, J., *Learner Control. User modeling and User Adapted Interaction*. Netherlands, Kluwer Academic Publishers,11, p.111-127, 2001.

[Kea87] Kearsley, G. P. Artificial intelligence and education: Applications and methods. Reading, MA: Addison-Wesley, 1987.

[Ken99] Kendall, E.A., Role models - patterns of agent system analysis and design. BT Tech. Journal, 17(4), 46-57, 1999.

[Kin96] Kinny, D., Georgeff, M. & Rao, A. A methodology and modelling technique for systems of bdi agents. In *Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-96)*, LNCS 1038. Springer Verlag, 1996.

[Kin97] Kinshuk & Ashok, P., “A conceptual framework for Internet-based Intelligent Tutoring Systems”, *Knowledge Transfer*, vol.2, A.. Behrooz, London, UK, pp.117-124, 1997.

[Kit02] Kitamura, Y., Tsujimoto, H., Yamada, T. & Yamamoto T., Multiple character – agents interface: an information integration platform where multiple agents and human users collaborate, Proceedings of AAMAS-02, New York, 2002.

[Kle86] Kleinke, C. “Gaze and eye contact: A research review,” *Psych. Bull.*, vol. 100, no. 1, pp. 78–100, 1986

[Koe97] Koedinger, K.R., Anderson, J.R., Hadley, W., & Mark, M. Intelligent tutoring goes to school in the big city, International Journal of Artificial Intelligence in Education, 8, pp. 30-43, 1997.

[Kod96] Koda T. Agents with faces: a study on the effect of the personification of software agents, Master thesis, MIT Media Lab, Cambridge, MA, 1996.

[Koj98] Kojiri, T. & Watanabe, T. “Cooperative learning support mechanism based on scenario of specifying solving process”, In *Proceedings of ICCE’98*, Vol.1, pp. 133-140, 1998.

[Koj99] Kojiri, T. & Watanabe, T. “Adaptable learning environment for supporting a group of unspecified participants in web” In *Proceedings of STIE’99*, pp. 1937-1942, 1999.

[Kol84] Kolb, D. A. “Experiential learning: Experience as the source of learning and development.” New Jersey: Prentice-Hall 1984.

[Koy01] Koyama, A., Barolli, L., Tsuda, A., & Cheng, Z. An agent-based personalized distance learning system. In *Proc. 15th International Conference on Information Networking*, 2001.

[Kub98] Kubat, M., Holte, R. C., & Matwin, S. “Machine learning for the detection of oil spills in the satellite radar images”. *Machine Learning*, *30*(2), pp.195–215, 1998.

L

[Lak99] Lakoff, G., & Johnson, M., Philosophy in the Flesh: The Emboddied Mind and its Challenge to Western Thoughts. New York: Basic, ch. 12-13, 1999.

[Lam83] Lampson, B.W., Hints for computer systems design, *ACM Operating System Review*, vol. **17**(5), pp. 33–48, 1983.

[Lap04] Laperrousaz, C., Leroux, P. &Teutsch, P. *Tutor follow-up of a Distance Collective Activity*, Proceedings of the 5th IEEE International Conference on Information Technology Based Higher Education and Training, Turkey, 2004.

[Lee01] Lees, B. & Ye, Y., *Preface of the Proceedings of ASCW01 – Workshop of Agent-Supported Cooperative Work*. 5th International Conference on Autonomous Agents, 2001.

[Leh95] van Lehn, K., & Martin, J., *A Bayesian approach to cognitive assessment*, Cognitively Diagnostic Assessment, 141-165, 1995.

[Lep93] Lepper, M.R., Motivational techniques of human-tutors: Lessons for the design of computer-based tutors, Lawrence Elbaum Associates, 1993.

[Les97] Lester, J.C., Converse, S.A., Kahler, S.E., Barlow, S.T., Stone, B.A., & Bhogal, R.S. ,*The Persona effect: Affective impact of animated pedagogical agents*, In Proceedings of CHI-97, 359-366, 1997.

[Les99] Lester, J., Stone, B., & Stelling, G. "Lifelike pedagogical agents for mixed-initiative problem solving in constructivist learning environments." *User Modeling and User-Adapted Interaction, 9*(1–2), pp.1–44, 1999.

M

[Mae90] Maes, P., "Guest Editorial: Designing Autonomous Agents", In *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, Elsevier Science Publishers, 1990.

[Mae95] Maes, P., "Artificial Intelligence Meets Entertainment: Lifelike Autonomous Agents." *Communications of the ACM*, Vol. 38, no. 11, 108–114, November 1995.

[Mae97] Maes, P. "Agents that reduce work and information overload." In J. M. Bradshaw (Ed.), *Software agents* (pp. 145–164). Menlo Park, CA: AAAI Press, 1997.

[Maj95] Major, N. Modelling Teaching Strategies. Journal of Artificial Intelligence in Education, 6 (2/3): 117-152. 1995.

[Mal94] Malone, T., & Crowston, K., The interdisciplinary study of coordination. *ACM Computing Surveys, 26*(3), 87–119, 1994.

[Mal97] Malone, T., Grant, K., & Lai, K.-W., Agents for information sharing and coordination: A history and some reflections. In J. M. Bradshaw (Ed.), *Software agents* (pp. 109–143). Menlo Park, CA: AAAI Press, 1997.

[Mar04] Marin, B., Hunger, A., Werner, A., Meila, S., & Schuetz, C. "An Intelligent Tutor-Agent to Support Collaborative Learning within a Virtual Environment", *Proceedings of IEEE Int. Conf. on Systems, Man and Cybernetics*, ISBN 0-7803-8567-5, Hague, Netherlands, Oct. 2004.

[Mar04b] Marin, B., Hunger, A., Werner, A., Meila, S., & Schuetz, C. *A synchronous groupware tool to conduct a spatially distributed collaborative learning process.* Proceedings of the 5th IEEE International Conference on Information Technology Based Higher Education and Training, Istanbul, Turkey, 2004, pp. 269-274, 2004.

[Mar05] Marin, B., Hunger, A., Werner, A., Meila, S., & Schuetz, C. *Roles of an Intelligent Tutor-Agent within a Virtual Society*. Proceedings of IEEE International Symposium on Applications and the Internet, 2005.

[McC78] McCarthy, J. "*Ascribing mental qualities to machines*", Technical report, Stanford AI Lab., 1978.

[McC92] McCrae, R., & John, D. P. An introduction to the Five Factor Model and its applications. *Journal of Personality, 60,* pp.175–215, 1992

[McC00] McCalla, G., Vassileva, J., Greer, J., & Bull, S.: "Active Learner Modeling", In: *Gautier, Frasson & Vanlehn (Ed.). Proceedings of ITS'2000*, Springer LNCS 1839, pp. 53- 62, 2000.

[Mic89] Miceli, M. & Castelfranchi, C. A cognitive approach to values. Journal for the Theory of Social Behavior, 19, pp. 169-193, 1989.

[Mil97] Millon, T. "The MIPS: Gauging the Dimensions of Normality. In *The Millon inventories: clinical and personality assessment.* New York. Guilford, pp. 498-522, 1997.

[Mof97] Moffat, D. "Personality parameters and programs", In R. Trappl and P. Petta, editors, *Creating Personalities for Synthetic Actors*, Springer, pp. 120-165, 1997.

[Mor98] Morin, J. F., & Lelouche, R. Agent-oriented tutoring knowledge modeling in a problem-solving ITS. In *Proceedings of the ACMSIGART Workshop on Interaction Agents (IA 98)* (pp. 26–32). May 24, L'Aquila, Italy, 1998.

[Mus04] Mustapha- Syed, S.M.F.D. "Agent mediated for intelligent conversational channel for social knowledge-building in educational environment", *Proceedings of the 5th Int. Conf. on Information Technology Based Higher Education and Training*, Istanbul, pp.533-538, 2004.

[Mur99] Murray, T. Authoring intelligent tutoring systems: an analysis of the state of the art, International Journal of Artificial Intelligence in Education, 10, p.98-129, 1999.

[Mye93] Myers, B.A., Chuang, Y.S.A., Tjandra, M., Chen, M.-C. & Lee, C.-K.: "Floor control in a highly collaborative co-located task", In *Interact '93 and CHI'93 Conference Companion on Human Factors in Computing Systems*. Amsterdam, Netherlands 1993.

N

[Nas96] Nass, C., & Reeves, B. The Media Equation: how people treat computers, televisions and new media like real people and places, Cambridge University Press, Cambridge, 1996.

[Nas00] Nass, C., Steuer, J. & Tauber, E., „*Truth is the beauty: Researching embodied conversational agents*“, Embodied Conversational Agents, eds. Cassell, J., Prevost, S., Sullivan, J., Churchill, E., MIT Press, Cambridge, pp.374-402, 2000.

[Nas00a] Nass, C., Isbister, K., & Lee, E.-J. Truth is beauty: Researching embodied conversational agents. In S. Prevost, J. Cassell, J. Sullivan, & E. Churchill (Eds.), *Embodied conversational agents* (pp. 374–402). Cambridge, MA: MIT Press., 2000.

[Nic94] Nicholson, A.E. & Brady, J.M. Dynamic belief networks for discrete monitoring. *IEEE Transactions on Systems, Men and Cybernetics,* 24(11), pp. 1593-1610, 1994

O

[Ohl87] Ohlsson, S. Same principles of intelligent tutoring. In R. W. Lawler, & N. Yazdani (Eds.), *Artificial intelligence and education: Learning environments and tutoring systems*. Norwood: Ablex Pub, 1987.

[Ols87] Olson. J. R., & Rueter, H. H. Extracting expertise from experts: Methods for knowledge acquisition. *Expert Systems*, *4*(3), pp.152–168, 1987.

[Omi00] Omicini, A. SODA: Societies and infrastructures in the analysis and design of agent-based systems. In *AOSE*, pp. 185–193, 2000.

[Omi02] Omicini, A. Towards a notion of agent coordination context. In D. Marinescu and C. Lee, editors, *Process Coordination and Ubiquitous Computing*, pp. 187-200. CRC Press, 2002.

[Ort88] Ortony, A. , Clore, G. L. & Collins, A.. *The Cognitive Structure of Emotions*. Cambridge University Press, Cambridge, UK, 1988.

[Ort03] Ortony, A. On making believable emotional agents believable. In R. Trappl, P. Petta, & S. Payr (Eds.), *Emotions in humans and artifacts* (pp. 189–211). Cambridge, MA: MIT Press, 2003

P

[Pai01] Paiva, A., Affective Interactions: Towards a New Generation of Computer Interfaces, New York Springer-Verlag, 2001.

[Par87] Park, O., Perez, R. S., & Seidel, R. J. Intelligent CAI: Old wine in new bottles, or a near vintage? In G. P. Kearsley (Ed.), *Artificial intelligence and education: Applications and methods*. Reading, MA: Addison- Wesley, 1987.

[Pea00] Pearl, J. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufman Publishers, 2000.

[Per01] Persson, P., Laaksolahti, J., & Lönnqvist, P. "Understanding Socially Intelligent Agents —A Multilayered Phenomenon", *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS*, Vol. 31, no. 5, 2001.

[Pic97] Picard, R. W. *Affective Computing*. The MIT Press, Cambridge, MA, 1997.

[Pin01] Pinto, M., Amor, M., Fuentes, L. & Troya, J.M., "Collaborative virtual environment development – an aspect-oriented approach", *21st International Conference on Distributed Computing Systems Workshops (ICDCSW '01)*, 2001.

[Plu80] Plutchik, R. *A General Psychoevolutionary Theory of Emotion, Emotion: Theory, Research, and Experience* 1:3–33, New York: Academic, 1980.

[Pol91] Pollard, C., & Vogel, D. Group support system product comparisons. In *Proceedings of Hawaiian International Conference on System Sciences,* pp. 771–778, 1991.

[Pog00] Poggi, I,& Pelachaud, C. Performative Facial expression in Animated Faces. In J.Cassell, J.Sullinvan, S. Prevost & E.Churchill (eds.). Embodied Conversational Agents. Cambridge, MA: MIT press, 2000.

[Pre01] Prendinger, H. & Ishizuka, M. "Let's Talk! Socially Intelligent Agents for Language Conversation Training", *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,* vol.31, no. 5, pp. 465-471, September 2001.

R

[Ram93] Ramstein, C. "COOPDRAW: A Multi-Agent Architecture for a Shared Graphical editor", Proceedings of the 1993 Conference of the Centre for Advanced Studies on Collaborative Research: Distributed Computing – vol.2, Oct. 1993.

[Rao95] Rao, A. & Georgeff, M. BDI Agents: from theory to practice. In V. Lesser, editor, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95),* pp. 312–319, MIT Press: Cambridge, MA, USA, 1995.

[Ree96] Reeves, B. & Nass, C., The Media Equation: How People Treat Computers, Television and New Media Like Real People and Places, Cambridge Univ. Press, 1996.

[Ren03] Rene-Boullier, L. "Pedagogical coordination: a global vision of the personalized accompaniment of DESS DICIT", *Revue Sticef*, vol.10, 2003.

[Ris91] Ristau, C. A. "Before mind reading: Attention, purposes and deception in birds?," in *Natural Theories of Mind. Evolution, Development and Simulation of Everyday Mind reading,* A. Whiten, Ed. Oxford, U.K.: Basil Blackwell, pp. 209–222, 1991.

[Ros86] Rosenschein, S., & Kaelbling, L., The synthesis of digital machines with provable epistemic properties. *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge,* 83–98, Morgan Kaufmann Publishers, Inc., 1986.

[Ros02] Rosié, M., Stankov, S., & Glavinié, V. Application of Semantic Web and personal agents in distance education system. In *Proceedings, IEEE MELECON 2002,* Cairo, Egypt, May 7–9, 2002.

[Ros05] De Rosis, F., Poggi, I., Pelachaud, C., Carofiglio, V. & De Carolis, B.: "*Greta: A believable emobodied conversational agent*", In: "Multimodal Intelligent Information Presentation". Springer Series on "Text, speech and language technology", Vol 27, 2005.

[Rus89] Russell, J.A., Levicka, M. & Niit, T. A cross-cultural study of circumplex model of affect. *Journal of Personality and Social Psychology*, 57, pp.848-856, 1989.

[Rus95] Russell, J. & Norvig, P. *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall, 1995.

S

[San02] Santos, J. M., & Rodriguez, J. S. Towards an agent architecture to provide knowledge-based facilities for distance education. *Proceedings, 24th Int. Conf. Information Technology Interfaces (ITI2002)*, Cavtat, Croatia, June, 2002.

[Sco92] Scott, W.R., Organizations: Rational, Natural and Open Systems. Prentice Hall International, New York, 1992.

[Sch77] Schank. R.C. & Abelson, R.P. *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures.* Hillsdale, NJ: L. Erlbaum, 1977.

[Sch99] Schmidt, C., "A remote laboratory using virtual reality on the web", *Simulation*, vol.73, no.1, 13-21, 1999.

[See89] Seel, N. "*Agent Theories and Architectures"*, PhD thesis, Surrey University, Guildford, UK, 1989.

[Sen98] Sengers, P., *Anti-boxology: Agent Design in Cultural Context*, PhD Thesis, Carnegie Mellon Univ., Pittsburg, PA, 1998.

[Sha90] Shardlow, N. "*Action and agency in cognitive science*", Master's thesis, Department of Psychology, University of Manchester, Oxford Rd., UK, 1990.

[She92] Shelli, D., & Hayne, S., Distributed facilitation: A concept whose time has come? In *Proceedings of ACM CSCW'92*, pp. 314–321, Toronto, Canada, 1992.

[Shi03] Shih, T.K. et al., A Survey of Distance Education Challenges and Technologies, *International Journal of Distance Education Technologies*, Vol. 1, No. 1, January-March 2003, 1-21.

[Sho90] Shoham, Y., "*Agent-oriented programming*", Technical Report STAN–CS–1335–90, Department of Computer Science, Stanford University, 1990.

[Sho93] Shoham, Y. "Agent-oriented programming", Artificial Intelligence, 60: pp. 51–92, 1993.

[Shu96] Shute, V.J., Psotka, J. Intelligent tutoring systems: past, present, and future, Handbook of Research on Educational Communications and Technology, Macmillan, New York, pp.570-600, 1996.

[Sim94] Sims, R. *"Interactivity: A Forgotten Art?*" , 1994, http://itech1.coe.uga.edu/itforum/paper10/paper10.html

[Sis98] Sison, R., & Shimura, M. Student modeling and machine learning. *International Journal of Artificial Intelligence and Education*, *9*, pp. 128–158, 1998.

[Sla96] Slavin, R.E. *Research on Cooperative Learning and Achievement: What We Know, What We Need to Know.* Contemp. Educ. Psychol., vol21, Jan 1996.

[Sle82] Sleeman, D., & Brown, J.-S., *Introduction: Intelligent tutoring systems*, Intelligent Tutoring Systems, 1-10, 1982.

[Slo85] Sloman, A., *Real time multiple-motive expert systems*, Expert Systems 83, pp. 1-13, 1985.

[Slo90] Sloman, A., *Motives mechanisms and emotions*, The Philosophy of Artificial Intelligence, OUP, 1990.

[Slo95] Sloman, A., *Musings on the roles of logical and no-logical representations in intelligence*, AAAI Press, 1995.

[Sol95] Soldato, T. & Boulay, B. *Implementation of Motivational Tactics in Tutoring Systems.* In: Journal of Artificial Intelligence in Education vol.(6) no.4, pp.337-378, 1995.

[Sow84] Sowa, J. F. *Conceptual structures: Information processing in mind and machine*. Reading, MA: Addison-Wesley, 1984.

[Str99] Strack, S. “Millon’s normal personality styles and dimensions”, *Journal of Personality Assessment*, 72 (3), pp.426-436, 1999

[Sve99] Svennevig, J. *Getting Acquainted in Conversation,* John Benjamins, Philadelphia, 1999.

[Syk03] Sykes, E.R. and Franek, F. A Prototype for an Intelligent Tutoring System for Students Learning to Program in Java, in: Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies, Athens, Greece, pp. 485-486, 2003.

T

[Ten87] Tennyson, R. D., & Park, O. C. Artificial intelligence and computerbased learning. In R. M. Gagne (Ed.), *Instructional technology: Foundations*. Mahweh, NJ: Lawrence Erlbaum, 1987.

[Tha97] Thalmann, D., Noser, H., & Huang, Z. “Autonomous virtual actors based on virtual sensors”, *Creating Personalities for Synthetic Actors,* ed Trappl, R., Petta, P., Springer, Berlin New York, 25-42., 1997.

[Tra97] Trappl, R., and Petta, P., Creating Personalities for Synthetic Actors: toward Autonomous Personality Agents. New York, Springer-Verlag, 1997.

U

[Urb04] Urban-Lurain, M., Intelligent tutoring systems: An historic review in the context of the development of artificial intelligence and educational psychology, August 3, 2004, http://www.cse.msu.edu/rgroups/cse101/ITS/its.htm

V

[Vic98] Viccari, R.M., Martins-Giraffa, L. M.: "*The use of Agents techniques on Intelligent Tutoring Systems*", IV Congresso RIBIE, Brasil 1998.

[Vic98b] Vicente, A. Pain, H. *Motivation Diagnosis in Intelligent Tutoring Systems.* In: Proceedings 4th International Conference. ITS'98 San Antonio. Springer-Verlag, pp. 86-95.1998.

W

[Wag03] Wagner, G. The Agent-Object-Relationship Metamodel: Towards a Unified View of State and Behavior, Information Systems, vol. 28:5, 2003.

[War85] Ward, P.T. & Mellor, S.J. Structured development for real time systems. Prentice-Hall Internet, 1985.

[Wat79] Watanabe N. Statistical Methods for Estimating Membership Functions. *Japanese Journal of Fuzzy Theory and Systems, 5(4),* 1979

[Wei01] Weiss, G. Agent orientation in software engineering. *The knowledge engineering review*, 16(4), pp.349–373, 2001.

[Wen87] Wenger, E. *Artificial Intelligence and Tutoring Systems*, Morgan Kaufmann Publishers, Los Altos, California, 1987.

[Wen98] Wenger, E. *Communities of practice: Learning, meaning and identity*. London; New York: Cambridge University Press, 1998.

[Whi91] Whiten, A. Ed., Natural Theories ofMind Evolution, Development and Simulation of Everyday Mindreading. Oxford, U.K.: Basil Blackwell, 1991.

[Whi95] White,P. The Understanding of Causation and the Production of Action. From Infancy to Childhood, Hillsdale, NJ: L. Erlbaum, 1995

[Woo87] Woolf, B. P., & Cunningham, P. A. Multiple knowledge sources in intelligent tutoring systems. *IEEE Expert*, Summer, pp.41–54, 1987.

[Woo87a] Woolf, B. P. Theoretical frontiers in building a machine tutor. In G. P. Kearsley (Ed.), *Artificial intelligence and education: Applications and methods.* Reading, MA: Addison-Wesley, 1987.

[Woo91] Woo, C.W. Instructional Planning in an Intelligent Tutoring System: Combining Global Lesson Plans with Local Discourse Control, Ph.D. Thesis, Illinois Institute of Technology, Chicago, IL, 1991.

[Woo95] Wooldridge M. J. & Jennings N. R. Intelligent Agents: Theory and Practice. *The Knowledge En-gineeringReview,* 2(10):115–152, 1995.

[Woo96] Woolf, B. P. Intelligent multimedia tutoring systems. *Communications of the ACM*, April, *39*(4), pp.30–31, 1996.

[Woo00] Wooldridge, M.J., Jennings, N. R., & Kinny, D. The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3), pp.285–312, 2000.

Y

[Yao03] Yao, J.T., Yao, Y.Y., Web-based support systems, in: Proceedings of the WI/IAT Workshop on Applications, Products and Services ofWeb-based Support Systems, pp. 1-5, 2003.

[Yu97] Yu, E. Towards modelling and reasoning support for early phase requirements engineering. In *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (RE'97)*, pp.226–235, 1997

Z

[Zam00] Zambonelli, F., Jennings, N. & Wooldridge, M. Organizational abstractions in the analysis and design of multi-agent systems. In *First International Workshop on Agent-Oriented Software Engineering at ICSE*. 2000.

Appendix 1: List of Publications

Publications: Book Chapter

- **B. Marin,** A. Hunger, S. Werner "*A Framework for Designing Emotional Agents as Tutoring Entities".* in "Affective and Emotional Aspects of Human-Computer Interaction: Game-Based and Innovative Learning Approaches, vol. 1 The Future of Learning" a book edited by: Dr Maja Pivec, ISBN: 158603572x, IOS Press 2006.
- **B. Marin**, A. Hunger "*A Framework for Building Emotional-Motivational Agents as Intelligent Tutoring Entities*", In "Technology Enhanced Learning: Best Practices", a book edited by Dr. Miltiadis D. Lytras, to be published in 2008.

Publications: Journal

- **B. Marin**, A. Hunger "*Intelligent Agents: a New Paradigm to Support Collaborative Learning in Distance Education Systems*", **IEEE Journal of Learning Technologies,** ISSN 1438-0625, vol. 7 (2), 2005.
- **B. Marin**, A. Hunger, S. Werner "*Corroborating Emotion Theory with Role Theory and Intelligent Agents: a Framework for Designing Emotional Agents as Tutoring Entities,* in: **JOURNAL of NETWORKS,** vol.4 2006, by ACADEMY PUBLISHER
- A. Hunger, S. Werner, **B. Marin**, A. Tanuatmadja: "*Computergestützte Gruppenarbeit im kulturellen Kontext*" in **Essener Unikate –Bereichte aus Forschung und Lehre**-, Band 28: Neue Medien –Interaktivität und Ubiquität, ISSN 0944-6060

Publications: Conference Proceedings

- **B. Marin** "*The benefits of using emotional agents as motivational tutoring entities in distance educations environments*", Proceedings of the 5th IEEE International Conference on Information Technology Based Higher Education and Training, Kumamoto, Japan, 2007
- **B. Marin**, A. Hunger, S. Werner, "*A Framework for Designing Emotional Agents as Tutoring Entities".* Proceedings of IEEE International Symposium on Applications and the Internet (SAINT 2006), ISBN 0-7695-2508-3, Phoenix, USA, 2006.
- **B. Marin**, A. Hunger, S. Werner, "*Socially Animated Agents: Future of Intelligent Tutoring Systems".* Proceedings of International Symposium on System Theory SINTES 12 (SINTES 12), ISBN 973-742-148-5, Craiova, Romania, 2005.

- **B. Marin**, A. Hunger, S. Werner, "*Corroborating Role Theory and Intelligent Agents: a New Paradigm to Support Collaborative Learning".* Proceedings of IEEE/ WIC/ ACM International Conference on Intelligent Agent Technology (IAT 2005), ISBN 0-7695-2416-8, Compiégne, France, 2005.
- **B. Marin**, A. Hunger, S. Werner, S. Meila and C. Schuetz, "*Roles of an Intelligent Tutor-Agent within a Virtual Society".* Proceedings of IEEE International Symposium on Applications and the Internet (SAINT 2005), ISBN 0769522629, Trento, Italy, 2005.
- **B. Marin**, A. Hunger, S. Werner, S. Meila and C. Schuetz, "*A synchronous groupware tool to conduct a spatially distributed collaborative learning process*", Proceedings of the 5th IEEE International Conference on Information Technology Based Higher Education and Training, pp.269-274, ISBN 0-7803-8597-7, Istanbul, 2004.
- **B. Marin**, A. Hunger, S. Werner, S. Meila and C. Schuetz, "*A Generic Framework for an Interface Tutor Agent within a Virtual Collaborative Learning Environment".* Proceedings of 4th IEEE International Conference on Advanced Learning Technologies, pp.31-35, ISBN 0-7695-2181-9, Joensuu, Finland, 2004.
- **B. Marin**, A. Hunger, S. Werner, S. Meila and C. Schuetz, "*An Architectural Design of a Tutor Agent in a Collaborative Virtual Learning Environment".* Proceedings of The Symposium on Professional Practice in AI, a stream within IFIP World Computer Congress, pp.41-50, ISBN 2-907801-05-8, Toulouse, France, 2004.
- **B. Marin**, A. Hunger, S. Werner, S. Meila and C. Schuetz, "*An Intelligent Agent to Support Collaboration within a Distributed Environment".* Poster Proceedings of 27th German Conference on Artificial Intelligence, pp.122-136, ISSN 0939-5091, Ulm, Germany, 2004.
- **B. Marin**, A. Hunger, S. Werner, S. Meila and C. Schuetz, "*An Intelligent Tutor-Agent to Support Collaborative Learning within a Virtual Environment".* Proceedings of IEEE International Conference on Systems, Man and Cybernetics, ISBN 0-7803-8567-5, Hague, Netherlands, 2004.