

## Preface

The main topic addressed in this book is that intelligent tutoring systems (ITS) models and architectures can be designed and implemented based on the agent paradigm. More specifically, this book aims at addressing three main questions related to agents paradigm: (1) which agents technologies and methodologies can be used to design and implement ITS; (2) how this paradigm can be useful for creating affective and cognitive students models; and (3) how the agents approach can be employed to model and implement distributed and open architectures of ITS. Hence, this chapter introduces the matter in order to provide a general context to the reader: a background on cognitive artificial intelligence, BDI (beliefs, desires and intentions) architecture, cognitive models, and architectures for ITS. It is important to point out that the purpose of this chapter is not to provide the reader with cues on how to apply theses methods or theories. The application examples will be presented in the following chapters that compose this book. Finally, this chapter presents a brief description of each chapter in this book, relating them to the topics that this book is concerned.

Artificial intelligence (AI) is probably the most anthropomorphic of all computer science research areas and this fact should not be considered as a weak point; on the contrary, it is an advantage when handling with certain complex tasks or domains. It is a common implicit assumption from natural and applied sciences (and also from most parts of the computer science) that any theoretical analysis or experimentation about concrete phenomena must avoid anthropomorphic, subjective, or emotional aspects, focusing only on objective and measurable characteristics that are not subject to different interpretations by different observers of the same phenomena. On the other hand, AI, or at least the cognitive science branch of AI (Simon, 1981), here called cognitive AI (or symbolic AI), is probably the research field of computer science most akin to social sciences. The ultimate research goal of AI is to create computer systems that reproduce intelligent behaviour (Russel & Norvig, 1995). However, the only decision process, known to date, that is able to decide if some behaviour is intelligent is its comparison with some forms of intelligent human behaviour with all subjectivity and anthropomorphic features associated with this kind of behaviour. This implies in the anthropomorphic properties of AI in general and cognitive AI in particular.

Certainly, the main goal of AI must be pursued keeping the scientific parameters of accuracy, precision, and reproducibility at the highest possible levels. However, it is not possible to ignore that this task requires the analysis of the phenomena related to human cognition and to human subjective experience or emotions. In one way or another, all subfields of AI should answer how they will try to emulate this kind of behaviour. Some fields like neural networks (Haykin, 1999) and multiagent systems (MAS) composed of swarms of ant-like agents (Dorigo, Maniezzo, & Colomi, 1996) try an indirect approach to this subject, based on emergent properties of complex systems. Neural networks is a traditional research line of AI that starts from the assumption that modelling the processing cells of the human brain and simulating subsystems of these cells will accomplish some important part of it (effective and efficient

real-time vision, and voice recognition systems, for example). Ant-like agents and swarm MAS studies are recent AI research lines that, inspired by the highly effective and apparently intelligent way that colonies of social insects (bees and ants, so the “ant-like” agents) manipulate the environment, assume that some MAS emulating the behaviour of complex societies composed by less complex agents is one way to try to keep the scientific accuracy and precision.

Some drawbacks of dealing with agents have been recently identified in the research area of agent-oriented software engineering (AOSE). AOSE started with works from Wooldridge and Jennings (1999), Petrie (2000), and Jennings (2001b), and currently presents several distinct methodologies for software development (see Zambonelli & Omicini (2004) and Henderson-Sellers & Gorton (2002)). The problem is stated by DeWolf and Holvoet in the OOPLSA 2005 conference:

*Agent-oriented methodologies today are mainly focused on engineering the microscopic issues, i.e. the agents, their rules, how they interact, etc., without explicitly engineering the required macroscopic behavior [...]. Engineering the macroscopic behavior is currently done in an ad-hoc manner because there is no well-defined process that guides engineers to address this issue. (DeWolf & Holvoet, 2005, p. 145)*

The “macroscopic behaviour” cited by these authors is the desired emergent property of the MAS.

The approach carried out by cognitive AI to handle this issue is more direct and explicit. The goal is to analyse and propose cognitive models that present: (a) viable computational interpretations, (b) epistemological and psychological foundations, and (c) precise formal specifications. Each condition of the complex and difficult goal of cognitive AI has its justification. The cognitive models should be computational, at least from the theoretical point of view; otherwise, cognitive AI (or AI) cannot be considered part of computer science. In order to not reinvent the wheel, concepts used in these models should not be based only on naive intuition or common-sense psychology, but should be rooted on explicit epistemological and psychological foundations. The formal specification is the answer to avoid excessive anthropomorphism: the precise formal definition of any concept is independent of subjective belief, perception, or emotion about this concept, even when the concept that is being formalized is the concept of “subjective belief,” “perception,” or “emotion.”

Cognitive AI is far from achieving its objectives. However, starting from the foundation of this research area in 1980 (Simon, 1981), some important results have been achieved in the last years, mainly related to BDI models for agents and Multiagent systems (Rao & Georgeff, 1991a).

One active research line of cognitive AI is centred on the creation of intelligent tutoring systems (Self, 1998) or intelligent learning environments (ILE) (Fernandez-Manjon, Cigarran, Navarro, & Fernandez-Valmayor, 1998). This research field contributed with a very important multiagent paradigm, the *student model* representation. The idea is that the student model representation, and related ITS MAS concepts, together with BDI models for agents, could have a profitable application in the AOSE research area.

The motivation is that the use of weak notions of agency commonly accepted in AOSE processes eventually lead to the “macroscopic behaviour” problem (DeWolf & Holvoet, 2005). On the other hand, when the design of some MAS starts from a strong notion of agency, then the problem to achieve desired high-level properties becomes an integral part of the design process. Cognitive abstractions such as beliefs, goals, and intentions, and social abstractions like cooperation, competition, and negotiation, provide the ground where these high-level properties can be intuitively understood, and enunciated as system requirements. High-level agent architectures and models derived from ITS/ILE research, such as student models and pedagogical negotiation processes, can be used as a design framework for complex applications. Formalisms, just as modal logics, provide a way to state these requirements in a

precise and non-ambiguous form. Recent development in the field of logical programming languages, like X-BDI (Móra, Lopes, Viccari, & Coelho, 1998) or AgentSpeak(L) (Rao, 1996), can transform these requirements in prototypes, at least for certain kinds of applications. Using the spiral model for software development (Boehm, 1988), these prototypes can be used in the test and validation phases of initial stages from evolutionary development cycles of complex MAS. They are *proof-of-the-concept* systems able to exhibit the autonomous behaviour required for the final system, but not necessarily with its real-time performance.

## ITS GENERAL ARCHITECTURE

Since the 1970s, researchers in computer science for education observed the necessity to use techniques from AI in the development of educational systems. One of the AI promises for the ITS is the possibility to make a more flexible and adaptable software to the users' needs.

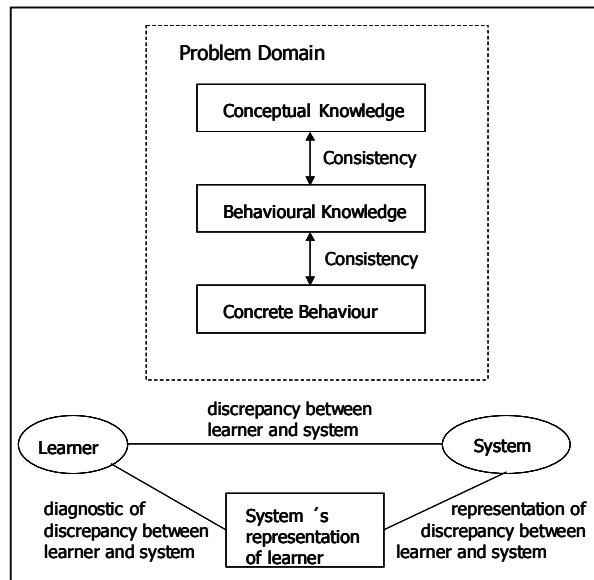
The traditional ITS architecture is composed of modules like domain knowledge (in some cases it is an expert system), user model, and pedagogical model. This traditional organization is still in use. However, in recent years, many systems for educational purposes have adopted the agent paradigm to explore the interaction and dynamic changes in teaching-learning environments. When a system has more than one agent, it is known as an MAS. Besides the change of the software engineering paradigm, the basic components (domain knowledge base, user model, and pedagogic strategies) still compose the ITS kernel. Hence, in this book, we are more interested in the use of agents technology to design and implement ITS.

To adapt the actions of a tutorial system to the student's necessities is a complex process that requires a variety of knowledge, expertise, problem resolution capacities, and strategies of man-computer interaction, evaluation, pedagogy, and presentation of multimedia information. Breaking this process into appropriate components, which are autonomous, proactive, and flexible, can reduce the complexity to construct a tutor.

Dillenbourg and Self's (1992) work describes a formal abstract framework that shows how the basic entities (modules or agents) of an ITS, as the tutor/domain module and the student model module, can be structurally organised in several abstraction layers. This framework also shows the relationships that take place between the entities of each layer and what kind of knowledge is related to each one of these relationships. The abstract layers of the framework form the "vertical" dimension. They are based upon the computational distinction among concrete behaviour, behavioural knowledge, and conceptual knowledge (see Figure 1). The basic relationship in the vertical dimension is of consistency between levels, that is, the consistency between the learner's real behaviour and the knowledge about possible behaviours, and the consistency between this knowledge and the conceptual knowledge related to the learning domain. The entity subclassification, used in the framework, form the "horizontal" dimension. It assumes the existence of three entities: the "system," "the learner," and "the system's representation of the learner" (the student model). The identification of discrepancies between these entities forms the basic relationship among them (see Figure 1). The interaction between the learner and the system is clearly contextualized as a search space problem. Methods for establishing the search space for learner models and for carrying out the search process were also reviewed in their work.

In effect, these interactions of pedagogical nature are the most important elementary units for the analysis of the teaching and learning process. The challenge is in the search of symmetry between man and machine. Such symmetry provides the same possibilities to the user and the system as to actions, and symmetric rights for decision taking. In an asymmetric mode, an agent has always the decisive point,

*Figure 1 Vertical and horizontal dimensions of Dillenbourg and Self abstract framework*



as there is no space for a real negotiation. In the symmetric mode, there is no predefined winner, as conflicts need to be solved through negotiation. Precisely, the cognitive processes that trigger an explicit declaration that justify an argument or refuse the partner's point of view are most likely to explain the reasons why collaborative learning is more efficient than solitary learning.

The main functions of ITS (explanation, education, and diagnosis) are traditionally implemented as a one-way mechanism, which means that the system has total control. Recent work (Flores, Seixas, Gluz, Patrício, Giacomel, Gonçalves, & Vicari, 2005), however, tries to treat them as bilateral processes. The model is built collaboratively, and there are some moments of negotiation. It is clear that for a negotiation to take place, there must be a level of latitude available to agents, otherwise anything can be negotiated. Discussions about the use of negotiation mechanisms in learning environments are not recent. According to Self (1992), there are two major motivations for the use of negotiation in ITS (i) make it possible to foster discussions about how to proceed, which strategy to follow, and which example to look for in an attempt to decrease the control that is typical of ITS, and (ii) give room for discussions that yield different viewpoints (different beliefs), providing that the agent (tutor) is not infallible. Dillenbourg and Self (1992) say that human partners do not negotiate a single shared representation, but they really develop several shared representations, that is, they move in a mosaic of different negotiation spaces.

## **DISCUSSIONS AND CURRENT WORK ABOUT ITS DEVELOPMENT USING AGENT ARCHITECTURE**

Following the approach taken by object oriented design and development processes, where the computational notion of "object" is a central concept, we will start our terminology by the computational notion of an "agent." Agents compose the basic element of computation in the domains we are aiming at, and MAS are simply systems formed by several agents working together. Adapting the most commonly used definition of agent (Jennings, 2001b) to incorporate the notion of design purpose, we assume that an

agent is a *computational process situated in an environment* that is *designed* to achieve a *purpose* in this environment through an *autonomous and flexible behavior*. From this point of view, the environment *is* the application domain where the agent will work to achieve its purposes.

An individual agent is an autonomous entity that has metalevel knowledge about itself and about other agents in the society and, therefore, can collaborate with each other to reach a common goal in the same environment.

Agents detect properties in the environment, or more commonly, changes in these properties through perceptions. These changes happen regardless of the agent, or may occur in response to some actions executed by the agent or by other agents, but the only way the agent has to detect them is through perceptions. Its characteristic is very important for ITS, in particular, for development of the student model. As presented before, the student model represents in the computer the human student cognitive (and also affective) state about the subject. The teaching actions intend to transform the student cognitive state in order to promote learning. Then, if the ITS is modeled using agents technology, the sensors can detect these changes and the student model will be updated.

## Pedagogical Agents and Multiagent ITS

Within the ITS area, we can consider agents as pedagogical agents. These kinds of agents incorporate multiple features of entertainment agents and artificial life. They have a set of normative teaching goals and plans to achieve these goals (e.g., teaching strategies), and associated resources in the learning environment. Pedagogical agents can be divided into goal driven (tutor, mentor, and assistance), and utility driven (MOO - virtual environments on the network where people can meet and communicate - and Web agents, for example). The goal-based agents decide their actions (to achieve the goal) based on information described by desirable situations. The utility-driven agents are used for pedagogical purposes such as labor agents and to help students find things.

As mentioned before, some researchers who use agents to build ITS adopt a mentalistic approach (Bratman, 1990) where the term agent means a computer system that can be viewed as consisting of mental states such as beliefs, intentions, motives, expectations, obligations, and so on. In others cases, authors adopt a generic organization to build each agent. In this organization, the project of each agent is divided into three distinct levels:

- **Decision level:** It models the agent's decision processes according to evidence received, either from artificial agents or students.
- **Operational level:** It is where decisions taken in the previous level are transformed into real actions and operations in the environment. This level is also responsible for gathering/organizing evidences and information necessary for decision taking.
- **Communication level:** It is responsible for the interaction among agents, and between agents and the external world. This level is responsible for sending/receiving messages among agents, and also for the interaction with users.

The levels compose the plan of construction (or functioning) of each agent, since in each level there are specific functionalities that must be modelled and implemented (modules, functions, processes, or software components). On the other hand, there must be the possibility for agents of sharing knowledge and information because of the social issues of the multiagent environment that must be handled. This implies the need of an ontology that defines meanings of messages, information, and decisions taken by each agent.

Pedagogical agents can be modelled as:

1. Cooperative agents who work in the background as part of the architecture of the educational system.
2. Personal and animated agents that interact with the user.

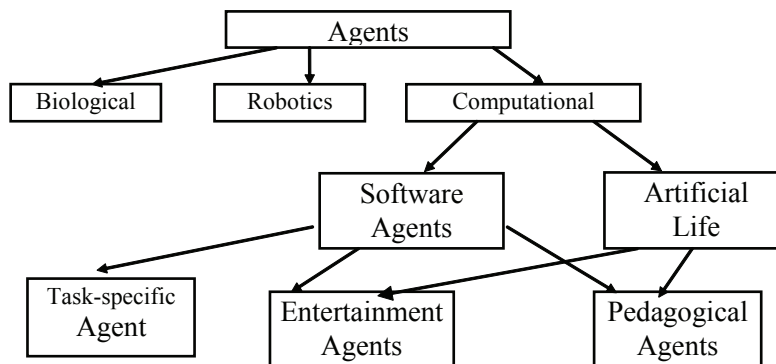
In the first case, the educational system is modelled and implemented using a multiagent approach, where each agent has a specific function in the system. These agents act in the background, they are transparent to the user, and exchange information among themselves in order to carry out actions that are appropriate to improve learning. Towards this approach we highlight the works by Silveira and Viccari (1999); Frasson, et al. (Frasson, Chaffar, Abdel Razek & Ochs, 2005); and Bica et al. (Bica, Verdin, & Viccari, 2006). According to Giraffa and Vicari (1998), architectures based on this approach are variations of the traditional and functional architecture of an ITS (domain knowledge base, student model, and teaching strategies), where one or more agents implement each function of the tutor. The control is distributed among agents. However, the user sees the system as a single entity while internally, it is composed of a society of agents.

In the second case, the animated pedagogical agents are human-like and animated agents, represented by a character that interacts with the student through voice and emotional attitudes and gestures. Some examples of animated pedagogical agents are Vincent (Paiva & Machado, 2002), Steve (Rickel & Johnson, 1998), and Cosmos (Lester et al., 1999).

With the purpose of understanding agents and pedagogical agent's classification, Figure 2 illustrates the agents' taxonomy in the context of ITS (Giraffa & Viccari, 1998).

The two major advantages of using agents in the conception of educational systems are modularity and openness. As the agents are independent, they are a powerful tool to make the tutorial system a modular system. Some efforts have been carried out towards the construction of components of tutors as agents that can be joined to form an ITS (Silveira & Viccari, 1999). Moreover, since each agent is an exclusive module independent from the others, it is easier to add other agents to these systems that will carry out new functionalities. They do not need to know the specific details to communicate with every agent, making the society more flexible and extensible. As agents are autonomous, they only need to know information about how to interact with other agents (what type of new information the system expects the agent sends, for example).

Figure 2. Agents' taxonomy



The multiagent system's modularity also makes possible to handle bigger and more complex problems: each agent can be specialized in its own tasks in the space problem (in terms of knowledge and abilities to solve problems). This modularity simplifies the project and development of the educational system. The developer can concentrate on knowledge representation, on the granularity analysis and on ways of reasoning that are different for the functionality of each agent. This modularity also permits the reuse of the components in different systems.

Besides, the distributed nature of the multiagent architectures lets the functionality of an educational system to be distributed in a computer network and in different platforms. This distribution allows the tutorial system to be constructed from several components that are in different platforms, allowing the use of appropriate tools without worrying about the platform. The distributed nature of these architectures also allows partial parallel processing.

## The Use of BDI to Model and Develop ITS

If the ITS is modelled using agents technology and in particular, the BDI model, we can assume that the purpose of an agent can be fully specified by the definition of its *beliefs* and *desires*, and that the behaviour of this agent is implied by its *intentions*, that is, we are considering the BDI cognitive model for our agents.

The use of BDI models in AI is not a simple application of naive concepts of belief, desire, and intention in software programming. Cognitive AI makes use of precise formal concepts that, in some cases, have the possibility to pass directly from formal design to software implementation by using logical programming languages like AgentSpeak(L) and X-BDI. The BDI-models can also be implemented by several successful and powerful agent architectures (Bordini, Hubner, & Vieira, 2005; Mora et al., 1998). The use of agent and Multiagent technologies represents a new way to design and implement ITS, this is the reason why we will dedicate an important space to discuss this subject in this book.

New abstractions are being added to the traditional BDI model, allowing that “outward looking” social concepts are integrated to the traditional “inward looking” mental state concepts commonly associated to BDI models. Mental states related to expectations, confidence, planning, and emotions have made the BDI modelling closer to human behaviour. Social concepts like cooperation, competition, and negotiation (among others), are fundamental to define complex coordinated behaviours, coalition work, and to define the role and scope of organizations, institutions, and societies.

It is possible to see how these concepts and abstractions are interrelated. Traditional works in Theories of Agency from Rao and Georgeff (1991a, 1991b), Cohen and Levesque (1990, 1997), Sadek (1992), and others show on how knowledge is related to beliefs (provisional knowledge), on how beliefs are related to commitments and intentions, and on how they are related to objectives, goals, actions, and communication. A more recent work by Bordini and colleagues (2002) shows how goals are related to planning in the case of resource-limited agents, and how reasoning and planning can be related to search heuristics. In our research, we are starting to integrate, in a single conceptual and formal framework (probabilistic modal logics), probabilistic subjective beliefs (*bayesian beliefs*) and Bayesian networks (BN) with the other concepts of the BDI model (Gluz, Vicari, Flores, & Seixas, 2006a, 2006b). This is a small step to decrease the gap between probabilistic and purely logical knowledge representation and reasoning methods.

Such research and its recent results are bringing out a computationally reproducible understanding of human cognitive and psychological behaviour. In these fields, a multiagent modelling paradigm was developed with interesting applications and implications to software engineering (SE) AOSE research areas: the *student model* paradigm.

The main idea behind this paradigm is that ITS agents must create internal models of subjects with whom they interact. Together with these subjects, there are purposeful and intelligent entities that may be artificial agents or human beings. This is the difference between ITS agents and other kinds of agents. ITS agents not only are continuously trying to understand the environment where they “live,” but they also need to know, theorize, plan, and understand several aspects of the other intelligent entities that inhabit the same environment. Without this psychological subjective knowledge, they are unable to actually teach anything. This is not a requirement for all sorts of MAS or agents, but it is the specialty of ITS agents.

The interaction of the agent with its environment is done by *actions* and *perceptions*. An action is an alteration in the external environment caused directly by the agent. From an intentional point of view, it also represents a way to attain an end (intention). Therefore, internally the agent should know (believe) the basic effects produced by possible actions and what are the relations of these actions to their intentions.

As mentioned, a particular intention is to be pursued by a plan of action that is composed of a set of actions structured by sequence, iteration, and test/choice order relations (operators). Plans are specified *through planning inference processes* that use a base of beliefs about which kind of *strategies* and *tactics* should be applied to achieve an intention. Plans of action do not need to be fully specified from the beginning. They can be partial and the agent can start to follow the plan and reassess or complete it during execution.

Agent’s perceptions produce updating in the base of the agent’s beliefs. However, the exact update produced by a particular perception depends on the current state of beliefs of the agent. Indeed, perceptions can be classified as expected or non-expected and, in a similar way of actions, should be related to the current set of intentions and its correlated plans.

The notions presented, up until now, provide a common set of concepts necessary to understand what are the internal (mental) states of *individual* agents; the relationships between these states and the agent’s behaviour; and, the effects of such behaviour in the interaction of the agent with its environment. These notions are established in a well-known theory of agency,<sup>1</sup> and can be thought as a conceptual framework that can be used to understand the kind of agent we are intending to work on.

Those concepts are enough to analyse and describe applications based on single autonomous and deliberative agents. An application aimed at helping users to accomplish some particular task or activity can be thought in terms of a help agent that will aid users to conclude this task, for example. The purpose of this agent is to be helpful to users on how to accomplish the task, and be able to detect when is necessary to interfere (and when is not, to not be boring). The belief base of this agent will be composed of knowledge related to how to represent the task, to what perceptions and actions are related to the tasks and to the users, and similar matters. Problem-solving knowledge will be composed of methods that can be used to accomplish the task, inference processes that can identify how the user is proceeding in the resolution of the task, and detect when to interfere or not. The BDI machinery will be responsible, then, for combining these purposes, belief bases, and problem-solving knowledge in appropriate intentions that will result in the behaviour of the agent.

This conceptual framework for individual agents can be extended in several ways, depending on the properties of the environment where the agent will live in or, equivalently, the characteristics of the application domain where the agent will work.

One basic extension is to consider that all planning reasoning processes of the agent are necessarily limited to *bounded rationality* that forces the agent to plan and act rationally with respect to its purposes, as its knowledge and abilities (its resources) will permit. Another possibility is to include some set of assumptions in the planning reasoning processes of the agent to solve the classical *frame problem* in

planning reasoning, and allow the agent to solve planning problems without having to worry about all the elements of the environment that are not directly related to the solution search process.

On the other hand, if the domain (environment) requires *non-monotonical* changes in the belief state that can produce contradictions, then it is important to explicitly handle these contradictions and the non-monotonical reasoning behind these contradictions. The work carried out with *Extended Logic Programming* (ELP) with *Well Founded Semantics with eXplicit negation* (WFSX) (Mora et al., 1998) had shown on how to handle these kinds of situations in certain domains through the use of logic programming.

Another usual extension is to add concepts related to time in order to allow the agent to reason/understand things like timeouts, instantaneous events, duration, and similar notions, which are very common in a wide range of applications, in particular for ITS development. These notions of time are already implicit in the framework because the definitions of intentions and plans are intended to achieve *possible future* goals. The idea here is to allow the explicit use of these notions in other kinds of reasoning.

An interesting new possibility is to consider domains that require degrees of belief. These kinds of beliefs are usually represented by subjective probabilities (Bayesian probabilities), and the conditional relations between these beliefs are expressed by *Bayesian networks* (Cowell, Dawid, Lauritzen, & Spiegelhalter, 1999; Pearl, 1986, 1993). When this kind of knowledge representation is used, then usually a single Bayesian network represents the entire belief base of the agent. Decision and planning problems are represented by extensions of Bayesian network diagrams, like influence diagrams (Shachter, 1986). If the uncertainty were to pervade even the actions and perceptions (observations) of the agent, then the entire decision problem of planning agent's actions becomes a partial observable markov decision problem (*POMPD*). In this case, recent works (Hui & Boutilier, 2006) have shown that dynamic Bayesian networks (DBN) can be successfully applied to represent these problems and specify the belief base and inference processes of the agent.

## The Use of Multiagent System to Model and Develop ITS

In terms of agent communication and multiagent system research, the study and application of social concepts to understand and model the communication and coordination of groups (teams or societies) of agents is a relatively recent field of research. The work of Singh (1998), which identifies limits that mental states concepts present when used to give the semantics of agent communication, and Castelfranchi and Falconi (1998, 1999), which study the characteristics of social relationships based on trustfulness notion, were important starting points in this new research line. Some examples of this line are the works by Colombetti and Verdicchio (2002), which models social interactions through commitment acts executed by agents through a new logic of social interaction based on principles of cooperation; the work by Grosz, Kraus, and others (Grosz, Kraus, Sullivan, & Das, 2002); the SPIRE system, which studies how rational and collaborative agents can adapt their goals and intentions to work in a group; and the work of Fischer and Ghidini (2002), which proposes a logic of abilities, beliefs, and confidence to model the behaviour and interaction among agents.

The concepts presented here should be regarded as open research topics as other concepts associated to individual and communicative agents. They are subject to discussion and revision as the research evolves. Anyway, the social concepts presented here are based on research literature and adjust very well to the experience we have accumulated with our ITS/ILE systems.

The most basic social concept is of *social relationship*, which is a relationship that occurs among two (or more) members of a group, team, or society. In our case, these members are agents, and societies, groups, or teams are systems of agents. Therefore, social relationships are simply any possible relation-

ship that can be formed among two (or more) agents of the system. In general, these relationships can be based on any perceived element of the environment or any noticeable event occurring in the environment that can be related to the action of other agents. Nevertheless, taking into account that our agents are purposeful and intentional agents, these relationships are more frequently based on some issue or matter that happens in the environment, but which depends on the purposes and intentions of the agents.

From this point of view, one dividing social issue is if the agents have common *conflicting purposes* or not. If the answer is true, then they can be considered as possible *competitive agents*. Otherwise, they can be held as *cooperative agents* if they have common non-conflicting purposes, or as *indifferent agents* if they share no common purpose at all. The conflicting purposes issue defines the *competitive/cooperative* social relationship among agents. It is important to note that these social issues are profoundly related to the individual and communicative aspects of the agents, in the sense, for example, that agents are really competitive agents only if they know they have common conflicting purposes based on previous communication or interactions, and they know that this conflict affects their current intentions (or already known future intentions).

Two or more agents are not always entirely competitive or cooperative due to the fact that they can share some common non-conflicting purposes and, at the same time, have other conflicting purposes. Recalling that purposes are represented by high-priority desires of the agent, this relationship can change with the passing of time because it is possible to temporarily align the desires or beliefs of agents. Following the model of Jennings et al. (Jennings, Faratin, Lomuscio, Parsons, Wooldridge, & Sierra, 2001a) about automatic agent negotiation, it is possible to consider the process of discovery and establishment of common purposes and desires among two or more agents as a *negotiation* interaction process when the desires are not subject to change, or as an *argumentation* interaction process when it is possible to change the desires of the agents.

Besides basic competition/cooperation relationships, agents can establish other sorts of social relationships. The agent can establish *dependence/independence* relationships with environment elements necessary/not necessary to achieve its goals, for example. The social version of this relationship is established in terms of the requirement or not of other agents to execute particular actions (or hold particular intentions) to allow the agent to achieve its goals. Similar to competitive relationships, dependence relationships also give birth to specific forms of interaction among agents. Particularly, the existence of dependence relationships among cooperative agents makes necessary the creation of *coordination interaction mechanisms* between these agents, to allow them to achieve their purposes.

Another example of social relationship is the *confidence/doubt* relationship that can be established between one agent and another in terms of the expected behaviour of this other agent in some future situation or state. In this perspective, one agent trusts or establishes a confidence relationship with another agent in respect to certain situations if it believes that the other agent will behave properly (as expected) if these situations were to occur. The relationship of confidence is the basis for any kind of *social mediated agreement* or *social mediated commitment*, which can be established among agents based on socially constructed sets of customs, practices, and behaviours.

The list of social relationships presented earlier is not intended to be an exhaustive list of possible social relationships among agents. Although incomplete, this list depends on several open research issues that imply in different assumptions that can be made about agent societies and multiagent systems.

Probably the most important open question related to social relationships and interaction mechanisms in agent societies is if these relationships and mechanisms are preexistent to the system, and if their corresponding rules and beliefs need to be incorporated in the agents before the system starts operation, or if they arouse and evolve during the lifetime of the system based on the knowledge and behaviour of individual agents not necessarily related to social issues. These approaches cannot be considered

necessarily as mutually excluding. They compose the spectrum of possible solutions on how to design and implement social interactions in multiagent systems. Related to this issue are the problems on how the agents will know if they will compete, cooperate, or be indifferent with other agents. Also, what kind of dependencies the agent had with the other agents and if the agent will trust, or not, other agents in particular situations.

The approach taken by some particular multiagent system to solve these matters is expressed by a set of *social and communication principles* to be adopted by the agents of the system. It is a common assumption that the agents of ITS multiagent systems are cooperative agents. This is a *design principle* for these agents. The purposes of these agents are not conflicting by design. This same assumption is held by coordinated teams of agents seeking to solve a distributed problem, which is a common assumption in multiagent systems aimed at distributed problem solving. The problem in these teams of agents is to identify the dependence relationships among agents. Several solutions can be used to solve this problem. Simple systems can assume that this knowledge must be incorporated by design in the belief base of each agent. Complex systems can use interaction protocols (like blackboards and contract nets) to solve the matter. In this case, only the knowledge about how to use these protocols needs to be incorporated by design in the agents.

Summarising the discussion for modelling and developing ITS, we will need a methodology with a strong and powerful set of high-level agent abstractions. It is also important that these abstractions are considered from the beginning of the software engineering process, including the requirements engineering phase. The definition of agent used in the initial works of Wooldridge and Jennings on AOSE as an “...encapsulated computer system situated in some environment and capable of flexible, autonomous action in that environment in order to meet its design objectives.” (Jennings, 2001b, p.36) may induce the reader to adopt the weaker, non-mentalist or cognitive notion of agency. It is an easier notion to be adopted by someone without cognitive AI background, but that was the point of Jennings paper: to sell the idea of the applicability of agents in software engineering processes with minimum AI support.

## COGNITIVE AI MODELING APPLIED TO ITS

In computational cognitive modelling, we hypothesize internal mental processes of human cognitive activities and express such activities by computer programs. Such computational models often consist of many components and aspects. Claims are often made that certain aspects play a key role in modelling, but such claims are sometimes not well justified or explored.

Computational cognitive modelling is an important aspect in cognitive science because it plays a central role in the computational understanding of the mind. A common methodology of cognitive science is to express a theory about human cognition in a computer program and compare its behaviour with the human cognitive behaviour.

Cognitive models can be of two major types: One type consists of the computational process models, and the other consists of (behavioural) mathematical models.

The former seeks to capture internal computational processes that are generated from cognitive behaviour (Corbett & Anderson, 1992). The latter works through measuring a number of behavioural parameters (such as recall rates, response time, or learning curves) in a precise way; through mathematical equations is the work developed by (Coombs, 1970). If we can find suitable measures and relations of these measures that reveal fundamental regularities, mathematical models can be used as tools to gain insights into cognitive processes. They can also be useful in serving as a kind of abstract ideal that computational models try to match at the outcome level.

Computational models complement mathematical models by providing detailed internal process descriptions that reveal underlying mechanisms of behaviour. Computational modelling opens up the black box, although it is usually done in a highly descriptive way. As in most cases, we do not have enough cognitive data that can lead directly to a computational model; numerous assumptions need to be made, and parameters need to be set. Thus, models are often under constraint from data, even with the methodologies of protocol analysis and other stylized procedures. Another shortcoming of computational models is that they often fail to account for individual differences and, thus, serve only as an “average subject,” which is nonexistent and ultimately meaningless. Given the shortcomings of each of the two approaches, it is clear that some combination of the two approaches might be useful or even necessary (Anderson, 1993).

Abstract models can serve as a compromise between mathematical and computational models. In abstract models, assumptions and details not essential to the phenomena are omitted, yet computer programs are still a good way to generate model behaviour.

There is still a huge gap in our knowledge on how the natural mind works. However, by looking at the structure of mental states, we may build replicas of these systems to help us understand how these biological systems work. Such computer models are unlocking the secrets of a variety of artificial reactions. This can be the breakthrough we have been waiting for to design cognitive agents focused on particular problem domains.

The Mental States approach allows us to trace more precisely the intrinsic dynamics of an interaction between tutor and students. These results can be used to improve future modelling and help us to build better students and tutor models. The Mental States approach has much more to provide, and it will demand more research to improve properties and develop ways to represent teaching/learning situations.

The BDI modelling, that is a Mental States approach, allows for a more granulated representation of the student’s mental states during the problem-solving activities. For this reason, it is appropriate for student model representation. Also, the BDI approach is composed of proactive mental states (desires and intentions). Therefore, the student model can better represent the student’s actions in the ITS environment, and the student’s actions guide the tutor’s pedagogic tactics in order to achieve a more effective teaching. This fact provides knowledge for actions for the student model and allows that this part of the ITS be modelled and developed as a knowledge-intense agent, and also communicate this knowledge with the other agents in the environment. In some cases, as in MCOE, PAT, and AMPLIA applications (see the next chapters of this book), the student model can also make decisions about the student actions. It is an important aspect that differs our work from the others. Indeed, the use of BDI architectures appears to be adequate to build teaching and learning agents because the proactive mental states potentially lead agents into action. Another important feature is that desires may be contradictory (with other desires and agent’s beliefs) while beliefs allow the agents to constantly update its view of the world. These characteristics of mental states are very important to better representing the choreography that occurs during a teaching and learning interaction.

We have been using the concepts of agency and mental states as an abstraction to describe design and to build systems. As usual, architectures are quite implementation-oriented. They provide schemas to build agents and systems. Nevertheless, they are insufficient as an analysis tool, especially in the case of pedagogical agents. It is not easy to respect the pedagogical theoretical foundations that are the basis for such agents while building them. Formal models come into place as we are interested in both describing and analysing the autonomous behaviour of (artificial and natural) agents in an ITS.

The agent’s behaviour is based on the fact that the only thing the programmer has to do is to specify the agent’s mental states as a high-level form of program development. The inclusion of expectations in traditional BDI architectures enables greater flexibility and more complex behaviours. The ideas behind

this social approach introduce promising concepts to understand how to construct agents with special abilities to learn and teach, for instance.

Trying to go beyond the notion of behaviour and to know which mental states allow us to model knowledge and reasoning, we have been updating these architectures to represent some affective aspects of learners, such as effort, confidence, and independence. The affective information improves the system, and allows it to provide a more adequate help for the student. In other words, we believe that the use of both mental states and affective aspects permits more accurate selection of pedagogical strategies (see Chapters VI and IX).

The long-term purpose of our research is to define a multiagent environment where there is no explicit tutor or learner, but only a set of mental states and affective aspects that generate learning and teaching attitudes, which can be assumed by all agents that compose the environment. The acquisition of knowledge by the agents in this Multiagent society emerges from their interactions in which each agent can behave as a tutor or as a learner. For this much, the agent needs only to change its mental states (sometimes an agent assumes the learner mental states and another time assumes the tutor mental states set).

## STUDENT MODEL

The student model remains the weak part of such systems. This situation imposes a great restriction for designing better tutoring systems to aid the student to build his own knowledge. However, the strong restriction comes from our imprecise knowledge about mental activities of the students during the teaching/learning process. In order to build an ITS with a good student model, we need to better understand what is happening in the student's mind during the interaction. We need to understand the process and reproduce it into the machine. Much has been done to understand such process, according to different points of view: psychological, educational, and from computer science.

The simplest student model covers basic profile characteristics, normally informed by the student himself, through a form, and information on its level of knowledge in the domain. The construction of the student model is, therefore, an important aspect to take care in the pedagogical profile of the ITS.

Some models as Conati et al. (1997) and Gertner et al. (Gertner, Conati, & VanLehn, 1998) works are based on the student's previous knowledge, and brought up-to-date in accordance with the detected progress and its probable future action. Others, as Mayo and Mitrovic (2001), consider the model directed toward the accompaniment of the student answering the questions. Studies carried out by Bull and Pain (1995a) consider the importance of the student participation, in a collaborative way with the ITS, in the student model construction (open student model). The environment developed by Collins (Bull, Pain, & Brna, 1995b), is a good example of these ideas. Kay (2001) presented a student model controlled by the student and also considered the social context, like the collaborative work made among students. Indeed, the construction of the student model is a complex task, involving a diversity of variables that, moreover, must constantly be brought up-to-date, in accordance with the software development process. Hence, a good approach to deal with this complex task is to use formal specifications to design the student model.

The application of formal approaches to understand or conceptualise aspects of educational processes brought ITS research closer to cognitive agent models. Works from Self (1990, 1994) and Dillenbourg and Self (1992) present solid foundation on the application of formal methods to the analysis of student models. Particularly, the formal analysis presented in (Self, 1994) clearly shows that there is a profound relationship among several areas of AI, like machine learning and cognitive-agent modelling, and ITS research. The formal model defined by Self is derived from various areas of theoretical artificial intel-

ligence, particularly from the epistemic/doxastic modal logics (logics of knowledge/beliefs), and from BDI logic's used for agent cognitive and communication modelling.

The student model can be defined as the representation of some characteristics and attitudes of the learners, which are used to achieve individualised and appropriate interaction between computational and student environments. Its objective is to understand the exploratory behaviour of the learner in order to offer any necessary support whilst also maintaining a sense of control and freedom.

This model constitutes a description of student knowledge, learning skills, strengths, and weaknesses. In addition, the model can also take into consideration the domain of the problem that was taught and the student's learning process. It should also be updated when more information on the student as affective aspects, is obtained.

In the systems developed at present in our group (see Chapters VI, VII and IX), the general cognitive agent structure is formed by  $\langle B, D, I \rangle^T$  *tuples* where  $B$  is the set of agent's beliefs,  $D$  is the set of agent's desires,  $I$  are the set of agent's intentions, and  $T$  is the set of time axioms.

The *desires* of the agent are a set of sentences  $DES(Ag, P, Atr)$ , where  $Ag$  is an agent identification,  $P$  is a property, and  $Atr$  is a list of attributes. Desires are related to the state of affairs that the agent eventually wants to bring about; but desires, in the sense usually presented, do not necessarily drive the agent to act. That is, the fact of an agent having a desire does not mean it will act to satisfy it. It means, instead, that before such an agent decides what to do, it will be engaged in a reasoning process, confronting its desires (the state of affairs it wants to bring about) with its beliefs (the current circumstances and constraints the world imposes). The agent will choose those desires that are possible, according to some criteria.

Beliefs constitute the agent's information attitude. They represent the information agents have about the environment and about themselves. Set  $B$  contains sentences describing the problem domain using ELP. An agent  $A$  believes a property  $P$  holds at a time  $T$  if, from  $B$  and  $T$ , the agent can deduce  $BEL(Ag, P)$  for the time  $T$ . We assume that the agent continuously updates its beliefs to reflect changes it detects in the environment. We assume that, whenever a new belief is added to the beliefs set, consistency is maintained.

Intentions are characterised by a *choice* of a state of affairs to achieve, and a *commitment* to this choice. Thus, intentions are viewed as a compromise that the agent assumes with a specific possible future. This means that, differently from desires, an intention may not be contradictory with other intentions, as it would not be rational for an agent to act in order to achieve incompatible states. Intentions should also be supported by the agent's beliefs. That is, it would not be rational for an agent to intend something it does not believe possible. Once an intention is adopted, the agent will pursue that intention, planning actions to accomplish it, replanning when a failure occurs, and so forth. Agents must also adopt these actions, as means used to achieve intentions, as intentions.

The definition of intentions enforces its rationality constraints: an agent should not intend something at a time that has already passed; an agent should not intend something it believes is already satisfied or that will be satisfied with no efforts by the agent; an agent only intends something that it believes possible to be achieved, that is, if it believes there is a course of actions that lead to the intended state of affairs. When designing an agent, we specify only the agent's beliefs and desires. It is up to the agent to choose its intentions appropriately from its desires. Such rationality constraints must also be guaranteed during this selection process.

Agents choose their intentions from two different sources: from its desires and as a refinement from other intentions. By definition, there are no constraints on the agent's desires. Therefore, an agent may have conflicting desires, that is, desires that are not jointly achievable. Intentions, on the other hand, are restricted by rationality constraints. Thus, agents must select only the desires that are related to those constraints. Firstly, it is necessary to determine the subsets of the desires that are relevant according to the

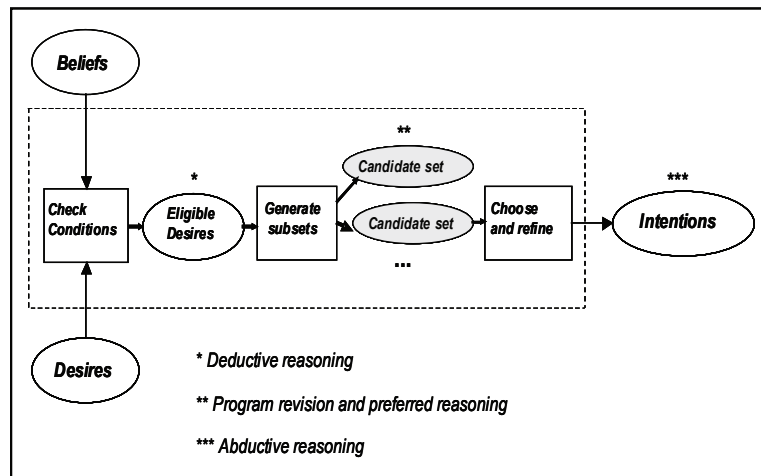
current beliefs of the agent. Afterwards, it is necessary to determine desires that are jointly achievable. In general, there may be more than one subset of the relevant desires that are jointly achievable. Therefore, we should somehow indicate which of these subsets are in preference to be adopted as intentions. This is done through a preference relation defined on the attributes of desires. The agent in our applications should prefer to first satisfy the most important desires. Additionally in preferring the most important ones, the agent adopts as much desires as possible. The selection is made by combining the different forms of non-monotonic reasoning provided by the logical formalism.

Once the agent adopts its intentions, it will start planning on how to achieve those intentions. During planning, the agent will form intentions that are relative to preexisting ones. That is, they “refine” their existing intentions. This can be done in various ways as, for instance, a plan that includes an action that is not directly executable, but can be elaborated by specifying a particular way of carrying out that action, or a plan that includes a set of actions that can be elaborated by imposing a temporal order on that set. Since the agent commits to the adopted intentions, these previously adopted intentions constrain the adoption of new ones. That is, during the elaboration of plans, a potential new intention is only adopted if it is not contradictory with the existing intentions and beliefs.

The next step is to define *when* the agent should perform all this reasoning about intentions. We argue that it is not enough to state that an agent should revise its intentions when it believes a certain condition holds, just like to believe that an intention has been satisfied or that it is no longer possible to satisfy it, as this suggests that the agent needs to verify its beliefs constantly. Instead, we take the stance that it is necessary to define, along with those conditions, a mechanism that triggers the reasoning process without imposing a significant additional burden on the agent. Our approach is to define those conditions that make the agent start reasoning about intentions as constraints over its beliefs. Recall that we assume that an agent constantly has to maintain its beliefs consistent whenever new facts are incorporated. When the agent reviews its beliefs and one of the conditions for revising intentions hold, a contradiction is raised. The intention revision process is triggered when one of these constraints is violated (see Figure 3).

Figure 3 presents all the steps involved in the decision process for a classical BDI agent.

Figure 3. BDI cognitive structure



## ORGANIZATION OF THE BOOK

In this book, we intend to present a modern view of ITS, focusing mainly on the conception of these systems according to a multiagent approach, and on the affective and cognitive modeling of the student also based on agent's technologies. In this way, this book consists of 14 chapters organized into three major sections. In the first section, we present some notions about intelligent agents and multiagent systems, which are necessary to understand some learning environments modelled according to the agents approach, as well as some agent technologies, and examples of their application, related to educational purposes as animated pedagogical agents, embodied conversational agents and intelligent learning objects. This first part is composed of four chapters. In the second section, we present some works on affective and cognitive modeling of students. This part consists of six chapters that give to the reader a vision of the state-of-art in the area, and also describe some works in order to exemplify the concepts showed. The next four chapters, which compose the third section, aim at presenting some works that profit of modularity and openness advantages of multiagent systems in order to implement decentralized and distributed architectures of ITS, also known as intelligent learning environments (ILEs).

A brief description of each of the chapters follows.

**Chapter I** aims at providing to the reader an overview of agent's paradigm. The chapter introduces the agent's concept, as also describes reactive (weak notion of agent, according to Wooldridge & Jennings, 1999) and cognitive (strong notion of AI) architecture of agents, focusing on the cognitive BDI model. Some challenges in the matter are also addressed by the chapter.

**Chapter II** addresses the reuse of learning objects to create ILEs. The authors are mainly concerned in the interoperability improvement among learning objects in agent-based learning environments by integrating learning objects technology and the multiagent systems approach. Aiming at achieving this goal, they propose the development of learning objects based on agent architectures: the intelligent learning objects approach.

**Chapter III** presents some basic guidelines to design and evaluate animated pedagogical agents. Some background in the matter is also introduced. Besides, the authors discuss some limitations and problems in the area that motivate their proposed guidelines.

**Chapter IV** introduces MagaVitta, an embodied conversational agent inserted into a multiuser intelligent learning environment, called CIVITAS, oriented to the construction and simulation of virtual cities. The authors also present the educational theories that ground CIVITAS modelling, as well as the pedagogical methodology developed for training teachers and for working with students in the educational environment.

**Chapter V** reviews the Bayesian-related technologies for modelling students in learning environments. The author states that Bayesian networks offer a mathematically sound mechanism for representing and reasoning about students under uncertainty, a topic that is also addressed in the chapter. An example of application of Bayesian network for modelling uncertainty information about the student in ITS is also presented in order to illustrate the concepts presented in the chapter.

**Chapter VI** addresses the modelling of collaborative ITS by using a multiagent architecture and a mental states approach. In order to exemplify this idea, the authors introduce MCOE, an ecological collaborative educational game that was implemented according to these notions. The authors also present a comparison of MCOE with Ecologist, a non-agent earlier version of MCOE, aiming at illustrating the benefits of using a multiagent and mental states approach for modelling and implementing ITS.

**Chapter VII** introduces Pat, an animated pedagogical agent that infers student's emotions in order to adapt the system to student's affective states. The authors issue the use of a BDI approach for the inference of emotions by student's actions in the ILE interface using an appraisal-based psychological model of emotion. The chapter also lists Pat affective tactics that are expressed through emotional

behaviour and encouragement messages of the agent lifelike character. An evaluation of Pat's affective tactics with pedagogues and psychologist is also described in the chapter.

**Chapter VIII** is concerned with modelling student's self-efficacy. The authors present a fuzzy-logic approach for student's self-efficacy inference and modelling. The authors illustrate their work by presenting Intelliweb, an intelligent e-learning system that covers the domain of vegetal anatomy for undergraduate students of a biological science course. This environment is composed by two agents: the (i) SEM, which was conceived for modelling student's self-efficacy; (ii) and PAT, the animated pedagogical agent presented in Chapter VII that was introduced in IntelliWeb to apply affective tactics in order to increase student's self-efficacy. An evaluation of IntelliWeb is also presented by the authors.

**Chapter IX** discusses the use of affective information about students to support the decision of an ITS on establishing pedagogical actions. The chapter is mainly concerned with the inference of the student's motivation to learn. A prototype developed to test some of those ideas was implemented as an agent that is modelled through mental states and is responsible for the inference of the student's affective states and the choice of the pedagogical actions. This agent was introduced in the ILE Eletrotutor III for the evaluation that is also described in the chapter.

**Chapter X** presents a proposal for using probabilistic networks in order to model affective information about students. This model is used by an affective agent, Social Agent, in AMPLIA ILE in order to promote collaboration between students. AMPLIA, which is a probabilistic multiagent environment for the medical area, is also described in this chapter. The authors state that the probabilistic networks, that is, Bayesian networks, are a powerful tool to deal with complex and uncertain domains, like the medical domain. The authors also present some experiments and results obtained in considering emotions to promote collaboration in AMPLIA.

**Chapter XI** presents a set of tools for constructing multiagent based ITS, and describes a methodology for guiding the development of ITS. The main goal is to make multiagent based ITS development more efficient and useful for both developers and authors. This chapter also describes Mathema ILE as a reference model for the construction of ITS. Two study cases are also presented in order to illustrate the advantages in using their proposal.

**Chapter XII** describes Leibniz and E-M@T. Leibniz is a pedagogical agent that was inserted in E-M@T, an ILE developed to give support for calculus classes of engineering courses. The authors also review the Solidarity Assimilation Group theory that grounded E-M@AT design. Initial results obtained from the application of the prototype to real classrooms and future perspectives of this work are also presented by the authors.

**Chapter XIII** discusses some guidelines for modelling and designing ITS for distributed learning. The author also identifies trends that are influencing the development of intelligent tutoring systems and suggests some topics for future research and development in the matter.

**Chapter XIV** presents a multiagent approach for implementing ILEs. The proposed architecture exploits the assumption that each teaching subject can be regarded as the synthesis of elementary pieces of knowledge, each of which can be presented by an independent expert. The authors also present two implementations of the proposed architecture and discuss the advantages in using it.

## REFERENCES

- Anderson, J. R. (1993). Problem solving and learning. *American Psychologist*, 48, 35-44.
- Bica, F., Verdin, R. & Viccari, R. M. (2006). Using the beliefs of self-efficacy to improve the effectiveness of ITS: An empirical study. In Alexander F. Gelbukh & Carlos A. Reyes García (Eds.), *MICAI 2006*:

*Advances in Artificial Intelligence, 5th Mexican International Conference on Artificial Intelligence*, Apizaco, Mexico, November 13-17, 2006, Proceedings (pp. 248-258). Lecture Notes in Computer Science 4293. Berlin: Springer.

Boehm, B. W. (1988). A spiral model of software development and enhancement, *IEEE Computer*, 21(5), 61-72.

Bordini, R. H., Hubner, J. F., & Vieira, R. (2005). JASON and the golden fleece of agent-oriented programming, In R. H. Bordini, M. Dastani, J. Dix, & A. E. F. Seghrouchni, (Org.). *Multi-agent programming: Languages, platforms and applications* (pp. 3-37), v. 15. Berlin: Springer.

Bordini, R. H., Bazzan, A. L., Janione, R. O., Basso, D. M., Viccari, R. M., & Lesser, V. (2002). AgentSpeak(XL): Efficient intention selection in BDI agents via Decision-Theoretic Task Scheduling. In *Proceedings of AAMAS2002*, Bologna, Italy. Bringing people and agents together: proceedings of AAMAS (pp.1294-1302), v. 3. New York: ACM.

Bratman, M. E. (1990). What is intention? In P. Cohen, J. Morgan, & M. Pollack (Eds.), *Intentions in communication* (pp. 15-31). Cambridge, MA: MIT Press.

Bull, S., & Pain, H. (1995a). Did I say what I think I said, and do you agree with me?: Inspecting and questioning the student model. In J. Greer (ed), *Proceedings of World Conference on Artificial Intelligence in Education* (pp. 501-508). Charlottesville, VA: Association for the Advancement of Computing in Education (AACE).

Bull, S., Pain, H., & Brna, P. (1995b). Mr. Collins: A collaboratively constructed, inspectable student model for intelligent computer assisted language learning. *Instructional Science*, 23, 65-87.

Castelfranchi, C., & Falconi, R. (1998). Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. *Proceedings of the 3rd international Conference on Multi Agent Systems* (pp. 72). Washington, DC, IEEE Computer Society.

Castelfranchi, C. & Falconi, R. (1999). Basic mental attitudes of a collaborating agent: Cognitive primitives for MAS. In F. J. Garijo and M. Boman, Eds., *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in A Multi-Agent World: Multiagent System Engineering* (June 30 - July 02, 1999). Lecture Notes In Computer Science (188-209), vol. 1647. London: Springer-Verlag.

Cohen, P., & Levesque, H. (1990). Intention is choice with commitment. *Artificial Intelligence*, 42, 213-261.

Cohen, P., & Levesque, H. (1997). Actions for artificial agents. In J. M. Bradshaw (Ed.), *Software Agents* (419-436). Cambridge, MA: MIT Press.

Colombetti, M., & Verdicchio, M. (2002). An analysis of agent speech acts as institutional actions. In *Proceedings of AAMAS2002*, Bologna, Italy. Bringing people and agents together: Proceedings of AAMAS (pp. 1157-1164), v. 3. New York: ACM .

Coombs, C. H. (1970). *Mathematical psychology: An elementary introduction*. Prentice-Hall.

Conati, C. et al. (1997). *On-line student modeling for coached problem solving using Bayesian networks*. Paper presented at User Modeling.

Corbett A., & Anderson, J. R. (1992). *Student modeling and mastery learning in a computer-based programming tutor source*. Lecture Notes in Computer Science, vol. 608, Proceedings of the Second International Conference on Intelligent Tutoring Systems (pp. 413-420). London, UK: Springer-Verlag.

- Cowel, R., Dawid, P. R., Lauritzen, S. L., & Spiegelhalter, D. J. (1999). *Probabilistic networks and expert systems*. New York: Springer-Verlag.
- DeWolf, T., & Holvoet, T. (2005). *Towards a full life-cycle methodology for engineering decentralised multi-agent systems*. Paper presented at *SIGPLAN OOPSLA 2005 Conference*.
- Dillenbourg, P., & Self, J. A. (1992). A framework for learner modelling. *Interactive Learning Environments*, 2, 111–137.
- Dorigo, M., Maniezzo, V., & Coloni, A. (1996). The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26(2), 29–41.
- Fernandez-Manjon, B., Cigarran, J. M., Navarro, A., & Fernandez-Valmayor, A. (1998). Using automatic methods for structuring conceptual knowledge in intelligent learning environments. In B. P. Goettl, H. M. Halff, C. L. Redfield, and V. J. Shute. (Eds.), *Proceedings of the 4th international Conference on intelligent Tutoring Systems* (August 16 - 19, 1998). Lecture Notes In Computer Science, vol. 1452 (pp. 264–273). London: Springer-Verlag.
- Fischer, M., & Ghidini, C. (2002). The ABC of rational agent modelling. In *Proceedings of AAMAS2002*, Bologna, Italy. Bringing people and agents together: proceedings of AAMAS (pp. 849–856), v. 3. New York: ACM.
- Flores, C., Seixas, L., Gluz, J. C., Patrício, D., Giacomel, F., Gonçalves, L., & Vicari, R. M. (2005). *AMPLIA learning environment architecture*. *Proceedings of ICCE2005*, Cingapura, 2005. Towards Sustainable and Scalable Educational Innovations Informed by the Learning Sciences (pp. 662–665). Tokyo: IOS Press.
- Frasson, C., Chaffar, S., Abdel Razek, M., & Ochs, M. (2005). *Emotion recognition agent for on-line learning systems*. Paper presented at FLAIRS 2005 Conference.
- Gertner, A. S., Conati, C., & VanLehn, K. (1998). Procedural help in Andes: Generating hints using a Bayesian network student model. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial intelligence/innovative Applications of Artificial intelligence* (Madison, Wisconsin, United States) (pp. 106–111). Menlo Park, CA: American Association for Artificial Intelligence.
- Giraffa, L., & Vicari, R. (1998). *The use of agents techniques on intelligent tutoring systems*. Paper presented at International Conference of the Chilean Computer Science Society, Antofagasta, Chile.
- Gluz, J. C., Vicari, R. M., Flores, C. D., & Seixas, L. (2006a). *Applications of a probabilistic knowledge communication framework*. Paper presented at Hybrid Artificial Intelligent Systems Workshop, Ribeirão Preto.
- Gluz, J. C., Vicari, R. M., Flores, C. D., & Seixas, L. (2006b). Formal analysis of the communication of probabilistic knowledge. In M. Bramer (Ed.), *Artificial Intelligence in Theory and Practice, IFIP 19th World Computer Congress*, TC 12: IFIP AI 2006 Stream, August 21–24, 2006, Santiago, Chile (pp. 31–40). London, UK: Springer.
- Grosz, B., Kraus, S., Sullivan, D., & Das, S. (2002). The influence of social norms and social consciousness on intention reconciliation. *Artificial Intelligence*, 142(2), 147–177.
- Haykin, S. (1999). *Neural networks: A comprehensive foundation*. Prentice-Hall.
- Henderson-Sellers, B. & Gorton, I. (2002). *Agent-based software development methodologies* (white-paper). Paper presented at *SIGPLAN OOPSLA 2002 Conference*.

- Hui, B., & Boutilier, C. (2006). Who's asking for help?: A Bayesian approach to intelligent assistance. *Proceedings of IUI'06*, Sydney, Australia.
- Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Wooldridge, M. J., & Sierra, C. (2001a). Automated negotiation: Prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2), 199-215.
- Jennings, N. R. (2001b). An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4), 35-41.
- Kay, J. (2001). *Learner control. User modelling and user-adapted interaction*, 11, 111-127.
- Lester, J. et al. (1999). Deictic believability: Coordinated gesture, locomotion and speech in lifelike pedagogical agents. *Applied Artificial Intelligence*, 13(4-5), 383-414.
- Mayo, M., & Mitrovic, A. (2001). Optimising ITS behaviour with Bayesian networks and decision theory. *International Journal of Artificial Intelligence in Education*, 12, 293-309.
- Móra, M. C., Lopes, J. G., Vicari, R. M., & Coelho, H. (1998). *BDI models and systems: Reducing the gap*. Paper presented at Agents Theory, Architecture and Languages Workshop, Canarias.
- Paiva, A., & Machado, I. (2002). Life-long training with Vincent, a Web-based pedagogical agent. *International Journal of Continuing Engineering Education and Life-Long Learning*, 12(1), 254-266.
- Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29, 241-288.
- Pearl, J. (1993). Belief networks revisited. *Artificial Intelligence*, 59, 49-56.
- Petrie, C. J. (2000). Agent-based software engineering. *Proceedings of the 5th International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM 2000)*. Manchester: The Practical Application Company Ltd.
- Rao, A. S., & Georgeff, M. P. (1991a). Modeling rational agents within a BDI-architecture. *Proceedings of Knowledge Representation and Reasoning (KR&R-91)*, pp. 473-484. San Mateo, CA: Morgan Kaufmann Publishers.
- Rao, A. S. & Georgeff, M. P. (1991b) Deliberation and its role in the formation of intentions. *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI'91)*, San Mateo: Morgan Kaufmann.
- Rao, A. S. (1996). AgentSpeak(L): BDI agents speak out in a logical computable language. In W. Van de Velde & J. W. Perram (Eds.), *Agents breaking away: Procs. of the 7th 24 European Workshop on Modeling Autonomous Agents in a Multi-agent World, (LNAI Volume 1038)*, pp. 42-55. Springer-Verlag.
- Rickel, J., & Johnson, L. (1998). Steve: A pedagogical agent for virtual reality. *Proceedings of 2nd International Conference on Autonomous Agents*, Minneapolis (pp. 165-172). New York: ACM Press.
- Russel, S., & Norvig, P. (1995) *Artificial intelligence: A modern approach*. Prentice Hall.
- Sadek, M. D. (1992). *A study in the logic of intention*. Paper presented at *Knowledge Reasoning*, Cambridge.
- Shachter, R. (1986). Evaluating influence diagrams. *Operations Research*, 34(6), 871-882.

- Self, J. A. (1990). Theoretical foundations for intelligent tutoring systems. *Journal of Artificial Intelligence in Education*, 1(4), 3-14.
- Self, J. A. (1992). *Computational viewpoints*. In R. Moyse & M. Elsom-Cook (Eds.), *Knowledge negotiation* (pp. 21-40). London: Paul Chapman.
- Self, J. A. (1994). Formal approaches to student modelling. In G. I. Mccalla & J. Greer (Eds.), *Student modelling: The key to individualized knowledge-based instruction* (pp. 295-352). Berlin: Springer.
- Self, J. A. (1998). Hanging by two threads: The evolution of intelligent tutoring systems research. In B. P. Goettl, H. M. Half, C. L. Redfield, & V. J. Shute (Eds.), *Proceedings of the 4th international Conference on intelligent Tutoring Systems (August 16 - 19, 1998)*. Eds. *Lecture Notes In Computer Science*, vol. 1452. London: Springer-Verlag.
- Silveira, R., & Viccari, R. (1999). *Eletrotutor- A distance learning multiagents architecture*. Paper presented at International Conference on Web Based Modeling and Simulation, San Francisco.
- Simon, H. A. (1981). Cognitive science: The newest science of the artificial. In Norman, P. A. (Ed.), *Perspectives on cognitive science* (pp. 13-26). Ablex, Norwood.
- Singh, M. (1998). Agent communication languages: Rethinking the principles. *IEEE Computer*, 31(12), 40-47.
- Wooldridge, M. J., & Jennings, N. R. (1999). Software engineering with agents: Pitfalls and pratfalls. *IEEE Internet Computing*, 3(6), 20-27.
- Zambonelli, F., & Omicini, (2004). A. challenges and research directions in agent-oriented software engineering. *Autonomous Agents and Multi-Agent Systems*, 9, 253-283.

## ENDNOTE

- <sup>1</sup> Agency theory is directed at the agency relationship, in which one party (the principal) delegates work to another (the agent), who performs that work. Agency theory is concerned with resolving two problems that can occur in agency relationships. The first is the agency problem that arises when (a) the desires or goals of the principal and agent conflict, and (b) it is difficult or expensive for the principle to verify what the agent is actually doing.