Chapter 8
Decision-Making in Cognitive Tutoring Systems

Daniel Dubois[1], Roger Nkambou[1], Jean-François Quintal[1], and François Savard[2]

[1] Computer Science Department, University of Quebec in Montreal, P.O. Box 8888, Centre-Ville Station, Montréal, Québec, H3C 3P8, Canada
{dubois.daniel,Nkambou.roger}@uqam.ca
[2] École Polytechnique de Montréal, C.P. 6079 Succursale Centre Ville Montreal, PQ, Montréal, QC, H3C 3A7, Canada
francois@fsavard.com

Abstract. Human teachers have capabilities that are still not completely uncovered and reproduced into artificial tutoring systems. Researchers have nevertheless developed many ingenious decision mechanisms which obtain valuable results. Some inroads into *natural* artificial intelligence have even been made, then abandoned for tutoring systems because of the complexity involved and the computational cost. These efforts toward naturalistic systems are noteworthy and still in general use. In this chapter, we describe how some of this AI is put to work in artificial tutoring systems to reach decisions on when and how to intervene. We then take a particular interest in pursuing the path of "*natural*" AI for tutoring systems, using human cognition as a model for artificial general intelligence. One tutoring agent built over a *cognitive architecture*, CTS, illustrates this direction. The chapter concludes on a brief look into what might be the future for artificial tutoring systems, *biologically-inspired cognitive architectures*.

8.1 Introduction

Human teachers have features and capabilities that remain beyond the reach of current computer-based tutoring systems, but artificial systems continue to improve. Many approaches have been explored and a number of artificial intelligence algorithms were developed which yield interesting results and often mimic human actions, even attitudes, quite well. But they still lag on the communicative aspects (perception, language), in transfer of abilities across domains, in sociability and emotions, and even autonomy. If the goal is to offer human-level capabilities, why not have systems work the way humans do? Well, for one thing, understanding humans is not so easy, as the multiplicity of psychological theories demonstrates. Also, at the neuronal level, the computational brute force required is impressive, with the brain's hundred billion neurons, each with thousands of connections to other neurons. We have yet to decipher how the neural "modules" operate and interact, but research delivers new insights by the week. Theories about

R. Nkambou et al. (Eds.): Advances in Intelligent Tutoring Systems, SCI 308, pp. 145–179.
springerlink.com

the mind are being proposed and refined, and incorporate more and more of the biological aspects.

In his 1987 book about artificial intelligence and tutoring systems, Wenger (1987) presents a surprising number of early but nonetheless advanced research efforts aimed at going beyond action-reaction tutoring. From the outset, they explored various aspects of the goal of rendering human cognition: presenting the appropriate content, conversing, storing factual knowledge, using procedural knowledge, diagnosing, and so on. Initial steps were taken by Carbonell in 1970 when he introduced the idea of *information-structure-oriented CAI* (Carbonell 1970). He applied Quillian's (1968) semantic networks to knowledge modeling in learning systems as an accurate model of how people store and access information. From this seminal idea, he derived SCHOLAR, in which domain knowledge was separate from tutorial capabilities. Collins collaborated with Carbonell to extend SCHOLAR with better inference and dialog capabilities (Collins et al. 1975), yielding WHY (Stevens and Collins 1977). That system implemented the human Socratic strategy (orienting the learner through a series of questions leading to general principles and contradictions) with production rules as a succession of simple decisions. Self, for his part, addressed the need for explicit modeling of the learner, and more importantly, the need to implement the model as a set of programs that would than play an active role in the tutor's decision-making (Self 1974). The two psychologists Donald Gentner and Donald Norman set out to build artificial tutors to further investigate the ramifications of their theories about memory organization as a network of schemata or prototype concepts (Norman et al. 1976). In 1979, O'Shea produced the QUADRATIC tutor (O'Shea 1979), after observing with Sleeman (O'Shea and Sleeman 1973) that intelligent tutoring systems should be able to self-improve their set tutorial strategies.

The stake was to achieve systems capable of adapted actions without having to prepare ahead of time for all combinations of constraints. Indeed, adapting to the student's inferred state of knowledge and presentation preferences, rapidly becomes a gigantic undertaking in frame-based systems, even without trying to adapt the interaction in line with a chosen pedagogical theory. Yet, it remains true that having learners build lasting knowledge can best be achieved when the context is appropriate, including high motivation. The teacher needs sensitivity to various factors, and the ability to decode, weigh and infer, and prioritize. Reproducing this requires much more than what is traditionally implemented in artificial systems. This is where human-inspired processing may really shine. Anderson, in his Adaptive Control of Thought – Rational (ACT-R) theory of cognition (Anderson 1976, 1993) and Newell with SOAR, his own version of the mind as a production-system (Newell 1973, 1990), went a step further with regard to cognition and production rules, both proposing unified theories of how the mind works. In their proposals, at a conceptual level and even at some physical level, things get processed and moved via the action of a large collection of simple rules. This is an instantiation of Fodor's physical symbol system hypothesis (Fodor 1975; 1983), which states that every general intelligence must be realized by a symbolic system. This principle offers a unified framework upon which to build complete cognitive systems. However, arbitrary processes and algorithms can result from these rule-based system, since rules are constrained only by practical considerations. But at least the fundamental unifying principle remains: rules do it all. At a high level of abstraction, they may correspond to the simple processes that form cognition (Baars 1997; Dennett 2001; Dennett and Akins 2008; Dehaene and Naccache 2001; Hofstadter 1995; Jackson 1987; Minsky 1985).

What justification is there for seeking to create cognitive architectures rather than being content with using just any artificial intelligence technique that works? Langley, Laird and Rogers (2009) express the opinion that "Research on cognitive architectures is important because it supports a central goal of artificial intelligence and cognitive science: the creation and understanding of synthetic agents that support the same capabilities as humans." In contrast to AI techniques such as *expert systems* that solve specific problems, cognitive architectures aim for breadth of coverage across a diverse set of tasks and domains. In the words of Newell (1990), "A cognitive architecture specifies the infrastructure for an intelligent system that remains constant across different domains and knowledge bases." The appropriate fundamental principle should sustain scalability and generalization – a seductive holy grail.

The word *architecture* implies that the system attempts to model not only the behavior, but also the structural properties of the modeled original. This definition goes deeper than Taatgen, Lebiere and Anderson's (2006) view that "A cognitive architecture is a computational modeling platform for cognitive tasks." The latter definition sees a "cognitive architecture" as a system that includes, and supports the creation of, a cognitive model of the task one wants to teach to a learner (able to reflect the learner's cognition to some extent); whereas the former requires that the structural organization, and its processing (not just the observable behaviours), mimic the original.

A quick lexical digression may be helpful here, to clarify a term that underlies this whole section: *cognition*. Like many words (such as intelligence, emotions, consciousness), this one is plagued with multiple definitions and excessive breadth of coverage. It may refer to knowledge building, decision making, information processing (the "cognitivist" view of mental processing), or all of these and more (Neisser 1967). In tutoring systems, one finds that a *cognitive* tutor may includes a model of the cognitive steps a learner may follow, or a tutor may adopt the label based on the "cognitive *architecture*" view, in which the underlying mechanisms mimic biological ones, at least at some level of abstraction.

We propose the following definition of an ideal artificial cognitive tutoring agent: *an agent built on an architecture that offers structures, features and functioning comparable to the human model so that it is similarly capable of adaptation, learning, generalization within and across domains, and action in complex situations encountered in tutoring learners*. This definition points in the direction of *general intelligence*, to which we will return in the final section.

8.2 Various Levels of Tutoring Decisions

Traditional computer-aided instruction (CAI) is not doomed. Thanks to simple knowledge structures, it is easily implemented and maintained by almost anyone.

It will keep being used in classrooms and niche applications for a long time. But such instruction isn't sufficient for complex domains, in higher education and for learners that are not highly autonomous. It cannot sustain strong adaptation and yet provide effective tutoring. Plain frame-oriented systems are not sufficient because flexible, adaptive tutoring systems have to *build* decisions, all the time, about lots of things. Some of their *intelligence* is geared toward the aim of adapting to the user, by means that may be simple or more elaborate. Ideally, as the user of an intelligent tutoring system (ITS), I would like human-level support with all its richness: seeing the tutor understanding what I really mean when I answer, interacting with me in the way I prefer, sensing from my voice and my face when I am struggling, showing empathy, adapting to my affective state, joking with me, guiding my efforts to resolve problems, finding out what keeps me motivated, inventing creative ways of helping me to learn. I would like the artificial agent not to respond solely on the basis of observed events, but to try and understand what is happening in my mind. These are goals for intelligent tutoring systems. And some of these call for human-inspired decision mechanisms.

But here again, the term "*intelligent* tutoring system" is widely applicable. As long as a system adapts in some significant way to the learner, to fall into this category. And any that use human-*inspired* mechanisms can be called *cognitive*. Real-world observation of tutors does not exert much pressure toward changing this "all-inclusive" categorization: looking at how some human tutors tackle their task, one may be surprised to discover how little inexperienced tutors use complex and involved approaches while still making a difference (Graesser et al. 2005). However, accomplished tutors perform better than naive ones (Merrill et al. 1995; Cohen et al. 1982). This observation underlies the subject of this chapter: architectures that can yield results on a par with those of skilled human tutors.

Although the idea of cognitive architectures may seem novel and strange to most people in the computer science field, quite a few research efforts for mingling cognition and AI have been ongoing at the same time as the work on ACT-R and SOAR , although they may not have achieved the same level of recognition (Wenger 1987). Recent years have seen the development of more such systems, often with common features, and each with its own merits. In an updated and extended version of their 2002 article (Langley et al. 2009), Langley, Laird and Rogers offer a recent and well documented account of the field. Wikipedia (http://en.wikipedia.org/wiki/Cognitive_architecture) adds some interesting items to their list. Giving an account of their similarities and specificities would exceed the scope of this chapter. We will limit ourselves to offering an illustration of such an architecture. This chapter proposes a view of the way some recent intelligent tutoring systems, cognitive and otherwise, make tutoring decisions, and how human-like architectures may support these processes. A sample system, CTS, is described in some detail. We also offer a glimpse of the emerging trend of biologically-inspired architectures that may close the gap between so-called intelligent tutoring systems and their human mentors.

8.3 An Overview of Some Tutoring Decision Mechanisms

ACT-R cognitive tutors and their student modeling techniques have shown their potential, and there has been an opportunity to explore their limitations. For example, a model-tracing mechanism cannot engage students in multi-turn dialogs and cannot be successfully applied to ill-defined domains. Researchers have started from these observations to work on extensions of the original ideas or pursue other paths. One such extension is exemplified by the *Ms. Lindquist* ITS, which adds a separate tutorial model to the central decision engine. *Ill-defined domains* (see Lynch et al. 2006), by definition, cannot be described in detail and necessitate another approach, such as constraint-based modeling (CBM; Ohlsson 1994; Mitrovic et al. 2001; you may also like to look up Chapters 4 and 5 in this book). As an example of an ITS for ill-defined domains, we will present *Rashi*, which uses an expert rulebase, rather than production rules, to deal flexibly with the more relaxed procedures of an enquiry. It also shows a way of monitoring progress in ill-defined domains, where the procedural knowledge cannot be fully covered or described as absolute sequential steps. ACT-R and constraint-based modeling tutors differ in that, whereas a model-tracing tutor is designed to represent the steps of procedures and use this model to trace the learner's operations step-by-step, an expert-rules system tries to match observations to expert principles described as constraints, then relies on separate mechanisms for giving advice and tutoring (Mitrovic et al. 2003). *Andes*, a mature tutoring system, exemplifies a less stringent way of using model tracing, compared to the line of ACT-R cognitive tutors which permit little variation in steps. Andes offers multiple levels and types of help, all on-demand except for the green or red coloring of the learner's equations. Its original reliance on bayesian estimates for plan recognition and estimation of the learner's knowledge has been replaced in Andes2 by simpler mechanisms: asking the learner to indicate his current goal and what principle should apply, and evaluating his equations rather than trying to derive them from canonical ones. Indeed, creating the solution graph was a heavyweight endeavor. Solutions remain tractable only for simple problems; the same is true of validating equations through derivation. *AutoTutor*, for its part, adopts an almost provocative view of tutoring, rejecting the need for planning and relying solely on dialog-driving to implement effective coaching.

In the following section, we will briefly describe these tutoring systems, as illustrations of some of the possible decision-making mechanisms utilizing cognitive constructs and AI techniques. We will begin with a presentation of Anderson's Cognitive Tutors, which introduced two fundamental cognitive mechanisms that are still widely used: model tracing and knowledge tracing.

8.3.1 Tutors Based on the ACT-R Theory of Cognition

Backed by a rich body of research and an impressive track record of educational successes, the line of intelligent tutoring systems grounded in the ACT production system (later extended and reorganized into ACT-R theory of cognition) are prime examples of *cognitive* tutors. The distinction between declarative and procedural knowledge – between merely knowing an algebraic rule and being able to apply it in a problem – is central to ACT-R. It postulates that procedural knowledge can only be acquired with progressive integration, through problem solving, of what is initially declarative knowledge (Anderson et al. 1995). This theory inspired a line of intelligent tutoring systems based on a **model-tracing** algorithm. The core idea is to try to *trace* a student's cognitive steps by the parallel application of a series of **production rules** to facts relating to the problems to be solved. As a student works his way through the problems, his mastery of each rule is inferred by another algorithm, **knowledge tracing**. We will present these two algorithms in a little more detail in the next section.

8.3.1.1 LISP/Geometry/Algebra Tutors: Tutoring through Model Tracing

Brief description. In the 1990s, most products and research in the cognitive field were based on the ACT-R software system (grounded in the ACT-R theory of cognition). The tutors were developed using the Tutor Development Kit (TDK), itself based on the Tertl production rule engine (Choksey and Heffernan 2003). Descriptions of some artificial tutors based on ACT-R can be found in (Anderson et al. 1995). From early 2000 onward, work on most new tutors in the line has been based on the JESS production rule engine, notably in the Cognitive Tutor Authoring Tools (CTAT) environment (Aleven et al. 2009). In both systems, however, production rules and model tracing are central.

Decision mechanisms

Model tracing over production rules

The model-tracing algorithm, as mentioned above, does its best to trace a student's cognitive steps in resolving a problem. It is given the following inputs:

- The student model, further divided into 1) the current state of working memory (composed of working memory elements) and 2) a set of relevant production rules
- The student's last input in the user interface

The working memory elements are facts related to a problem. Following an example in (Heffernan 2001), such facts can be named-quantities in a problem and operators linking them (e.g. "distance to shore", "boat speed", the division operation).

Production rules are *if-then* statements composed of actions to be taken when some conditions are met. The conditions refer to elements in working memory (including goal elements), while actions can, notably, alter those elements or cause a message (hint, error feedback) to be displayed. The model-tracing algorithm tries to find rules which can lead to the current student's input. For example, if the student altered an algebraic expression from "7+2*g" to "7+(2*g)", a production rule causing the addition of parentheses to a sub-expression would be returned by the algorithm (given that such a rule was input to the algorithm).

Importantly, "buggy" rules are used to model students' expected errors, and helpful error messages can be included in the "action" part of these rules. The tutor sends the student the error message when the system runs the rule identified by the algorithm. Additionally, hints can be associated with rules and be provided to the student upon request, based on rules that, as computed, could lead to the solution.

When the model-tracing algorithm finds a correct rule (or a chain of such rules) matching the student operation, it provides visual feedback in the user interface, highlighting the step as correct. In the alternative case of the step being matched by a buggy rule, the tutor instead displays an error message. Although the messages are contextual in the sense of referring to the cognitive operations performed by the student (provided the model-tracing model encompasses the specific path), there is no other consideration by the tutor beyond the fact that the student has utilized the buggy rule. For instance, help will be offered whether it has already been given or not, it will not be adapted to the learner profile of the student, and it will not support the learner's cognition beyond the superficial evidence of the faulty knowledge. Some of these limitations can be addressed by adding a completely separate set of tutorial rules embodying pedagogical and tactical knowledge to provide deeper remedial guidance, as explored by Heffernan in *Ms. Lindquist* (Heffernan 2001), which is the system we will examine next.

From a practical standpoint, it should be noted that the production rules must be created manually for the domain at hand (algebra, geometry, etc.) The difficulty of this task led to the creation of authoring environments such as CTAT and automated systems such as Demonstr8, which tries to infer rules from demonstrations by experts (Blessing 1997).

Knowledge tracing:

Starting with version 3.0 of ACT-R, new principles were added and others were modified. Such changes include modularization of the architecture, a set of "sub-symbolic" principles associating weights to rules and to (the equivalent of) working memory elements, and the compilation of new rules on-the-go, by the engine itself. However, it appears these features are *not* used in the TDK. Even though such capabilities might seem to be good candidates for modeling some further aspects of student learning, things did not evolve that way. Instead, knowledge tracing was pursued as a more efficient way (in terms of development effort and computationally) of implementing knowledge about the learner (Koedinger, personal communication, 2009). In knowledge tracing, each rule is assigned a probability of being known by the student. As new problems are worked out by the student, rules that are correctly used (or those that *should have been* used) have their probability adjusted through a bayesian learning formula. The tutor then uses the resulting estimated level of mastery for each rule to select which problem should be presented to the student (Koedinger and Aleven 2007).

Cognitive aspect. The way the word *cognitive* applies to these tutors may not be the obvious one for a newcomer to the field. They are not cognitive in the sense that they are built on a cognitive architecture (see our comment about cognitive *architectures* in the introduction). For reasons of efficiency, they leave aside many cognitive operations (sub-symbolic modeling in declarative memory, chunking, transfer of declarative knowledge to procedural knowledge in rules) in favour of a more efficient *computational modeling platform for cognitive tasks*. "Cognitive" refers here to the fact that these tutors use a cognitive modeling *of the domain* and of the mental operations, correct or faulty, someone is likely to use in solving the problems. The ITSs trace the *learner's cognition*, then apply the action specified in the appropriate part of the matched rule. Information processing is operated by the rules themselves. In stricter terms, the *tutoring* in itself does not rely on humanly cognitive processing.

8.3.1.2 Ms. Lindquist: Tutoring through Model Tracing Extended with Tutorial Rules

Brief description. Ms. Lindquist (Heffernan 2001; Heffernan et al. 2008) offers web-delivered tutoring on writing expressions for algebra "story" problems (symbolization). The behaviors it manifests are inspired by the experience of the real Ms. Lindquist and by Heffernan's own teaching. It offers dialog-based sessions in which the learner's inputs are examined for errors, and the observed difficulty is then broken down into simpler steps. Resolution is promoted by asking questions rather than directly giving hints and advice. The tutoring relies on an explicit cognitive model of the tutoring process, and allows for multi-turn tutorial strategies.

Decision mechanisms. Pursuing the approach taken in the ACT-R cognitive tutors, Ms. Lindquist uses the same model-tracing mechanism for its student model and primary diagnosis tool, and the same bayesian knowledge-tracing. However, it supplements these primary mechanisms by adding more involved tutoring capabilities, with tutorial behaviors arising from selection rules. Following the student's input, the model-tracing algorithm analyzes the content and transfers its conclusions to the tutorial model. The selection rules of the latter decide on the appropriate reaction, which may be to use a tutorial strategy, display a buggy message or provide a hint. Tutorial strategies are contained in plans called Knowledge-Construction Dialogs (KCD) and, in more specific ones called Knowledge-Remediation Dialogs (KRD). Selection rules try out the possible reactions, where relevant, in the following order: KRD, buggy message, KCD, hint. The heuristic behind this ordering is to offer the most contextual response possible (KRD and buggy messages), then use a tutorial strategy (KRD or KCD) before folding back to buggy messages or hints. In multi-step interventions (based on a KRD or a KCD), an *agenda* data structure keeps a memory of the dialog history for questions still awaiting a correct answer from the student, and the remaining steps in the plan (questions that the tutor is planning to ask later on). The steps chosen from the KCD or KRD by the selection rules are added to the agenda. The action at the top of the agenda dictates the next action, usually asking a question.

Cognitive aspect. Ms. Lindquist utilizes a cognitive model for its student modeling, based on ACT-R theory, and a cognitive model of pedagogical interventions for its decisions on tutoring; both operate on a rule-based engine. KRDs and KCDs, replicate questions human tutors ask themselves (their thinking) before posing acts toward the student.

8.3.2 *Tutors That Use an Engineering Approach to Decision Mechanisms*

The next systems adopt an "engineering" approach to problem-solving: what is efficient and useful is what is needed. These systems are thus not organized around any cognitive principle. If a resemblance to cognitive mechanisms is found, it was not intended or sought. But, of course, the mechanisms at work in our minds always influence what solution patterns we find (read "recognize"), at least unconsciously, and the decision mechanisms present in the ITSs can always relate at some level, in some form, to human cognition. So, the distinction we are making here is that the organizing principles of the ITSs we are about to discuss are not voluntarily constrained to reflect the human brain or mind.

8.3.2.1 Andes2: Process Critiquing Based on Model Tracing Plus On-Demand Help

Brief description. Andes (VanLehn et al. 2005) is a tutoring system for Newtonian physics, computer literacy, and critical thinking skills. Its philosophy is to maximize student initiative and freedom during the pedagogical interaction. Intended solely as a homework support, its user interface tries to stay close to the old paper and pencil environment, letting the student process aspects of the problem and enter information in any order. However, it goes beyond paper and pencil to offer multiple levels of help, providing immediate feedback by coloring correct entries green and incorrect ones red, responding to "What's wrong with that?" help requests and supporting "What should I do next?" queries. It can solve algebraically the equations that the student has entered, provided the student has entered enough correct ones. It implements these capabilities via some specific user tools: the Conceptual Helper displays mini lessons on defective knowledge, the Self-Explanation Coach guides the student through a solved physics problem, and the Procedural Helper responds to help requests while the student is solving problems. A more complete description is found in Chapter 21.

Decision mechanisms. Andes' tutoring decisions on when to produce tutoring interventions are straightforward: immediate feedback is given when the equations are entered, and nearly all other help is on-demand. However, several AI mechanisms are used to achieve *process critiquing* (VanLehn et al. 2004). In process critiquing, attention is not devoted to following consecutive steps of a procedural skill. And, while tutoring makes use of model tracing at some point, it isn't primarily a consequence of step matching. Emphasis is put on application of principles, that is, on the more global process, leaving some leeway to the learner. At the core of this tutoring, much of the previous version of Andes help relied chiefly on a bayesian network describing the domain in terms of physics knowledge (expert) rules, problem-specific facts and a solution graph of problem-solving steps. Andes2 has replaced the probabilistic evaluations (Conati et al. 2002) with simpler methods. The flat solution graph with over 600 inference rules has been reorganized around a hundred principles, into a "bubble graph" whose nodes represent physical quantities involved in the problem, and principle application nodes relating to these quantities. Also attached to each principle application node is a *method graph* containing the series of steps that can realize each principle application. To decide how good the learner's inputs and equations are and what help to offer, Andes2 uses multiple AI techniques, primarily revolving around the bubble graph. The reader is referred to Chapter 21 for details.

Cognitive aspect. Andes is a good example of a successful ITS involving purely *artificial AI* techniques with no reference to cognitive processing, either in the student's mind or in the tutor. Nonetheless, the principle of *error handlers* is reminiscent of *elementary processes* in many cognitive theories (see Baars', Crick and Koch's, Dehaene and Naccache's, Dennett's, John V. Jackson's), and the bubble graph is a form of semantic net, which effectively depicts the association of ideas and the thematic organization of knowledge.

8.3.2.2 Rashi: Rule-Based Domain-Independent Tutoring of Student's Exploration

Brief description. Rashi (Woolf et al. 2005) is an inquiry environment designed for domain-independent coaching of very active, hands-on learning in ill-defined domains. It offers an authentic setup in which the learner performs a proper investigation, using tools commonly employed by researchers. Examination room, field and image explorers, interview tools, inquiry notebook, argument editor, various objects such as a field guide and images of artifacts let the learner examine, analyze, find evidence, form hypotheses and reach conclusions, guided throughout the process by the tutor. A *Coach* is coupled to this environment, with rules that aim at global argument formation rather than at reaching specific answers or following process steps. In its various implementations (for forestry, geology, biology and art history), the tutor monitors the learner's unconstrained activities, examines its findings, compares them to expert knowledge and presents prompts and reminders that guide and motivate the student toward extracting sufficient evidence and reaching sound conclusions.

Decision mechanisms. The tutor's coaching is rule-based, utilizing an overlay of student inputs, context and current status on expert knowledge. Coaching currently intervenes only on demand. There is no immediate reaction as in model tracing. The coaching relies on a database containing two types of expert rules: those that serve in monitoring the learner's operations, either for supporting well-formed arguments or the adequate use of available tools; and those which orient the student's search for more material to support his arguments. The coach's reasoning amounts to comparing the learner's input (keyword search) and its context in Rashi to propositions contained in the expert knowledge base (a directed graph of weighted propositions). It looks up the Inquiry Notebook to see whether the student has support for an argument, then inspects the Argument Editor to verify how well these elements are connected to form a satisfactory support. In effect, the Coach inspects the underlying graph representing the learner's discoveries and inferences, counts the correct and incorrect support and refutation links between facts and hypotheses, and computes their weighted value to determine whether they amount to a critical mass of valid support. Then, these observations are compared to the expert rules database. A rule issues an advice or a question when it corresponds to some logical flaw, a lack of supporting data or a missing intermediate step, etc. Rules can also move the learner to another place (to suggest another tool, bring up artifacts, etc.). The generic decision process is subject to the author's (teacher's) settings, such as how many arguments are required to support a specific conclusion, whether such and such rule is desired in the domain, which of the five types of feedback he wants his students to receive, or the order in which they should be offered. Teachers can also influence how the Coach uses the expert knowledge by specifying the priority of content to be presented when needed.

Cognitive aspect. As in Andes, rules and algorithms implement expert and tutoring knowledge. While the system is effective in building knowledge in the learner (with some sort of student knowledge modeling), there is no cognitive processing in the tutor.

8.3.2.3 AutoTutor: Dialog-Based Tutoring

Brief description. AutoTutor (Graesser et al. 2008; Graesser et al. 1999) is a tutoring system concerned with computer literacy, qualitative understanding of Newtonian physics and critical thinking skills. Its aim is to support the learner in his construction of knowledge, helping him express what he knows about the aspect currently focused on. The dialog is central in this process of active learning (see Chapter 9). As an initial contribution, the learner is asked to compose a paragraph about his understanding of a problem and his proposed solution. A complete and correct answer is generally obtained after a series of corrections and additions, with the conversational partners taking turns in a mixed-initiative dialog. Supporting, guiding and encouraging the learner's self-expression on the subject is what AutoTutor is about.

Decision mechanisms. AutoTutor's decision process makes use of a number of cognitive and AI techniques and algorithms. Depending on the current state of knowledge demonstrated by the learner in the course of the conversation, and the latest textual input, AutoTutor provides feedback to the student on what the student types, pumps the student for more information, prompts him to fill in missing words, gives hints, fills in missing information with assertions, identifies and corrects misconceptions and erroneous ideas, answers the student’s questions, or summarizes topics. To accomplish these functions, the consecutive outputs of the architecture's modules are used in a sequential chain of analyses, and then used jointly to decide on the tutor's dialog move (Fig. 8.1). The sequential processing is concerned with understanding and assessing the learner's utterance. For a more specific evaluation, the learner's input is segmented into main clauses by the Language Analyzer, which assigns to each of these a speech act classification. This information is organized into a structured *state object* and passed to the Assessor. The Assessor submits each clause to its Latent Semantic Analysis (LSA) algorithm for an evaluation of its similarity with sentences describing expected answers or misconceptions associated with the problem. It also gauges the likely effects of various dialog moves on the student's learning, and updates the Student Model (i.e., what the student knows about the expectations and misconceptions associated with the problem). All the new information is added to the state object. The Dialog Management module receives this updated, richer state object, including the dialog information from the previous state, notably the particular student speech act category. This rich information constitutes a context that is examined by a set of 15 fuzzy production rules that decide which dialog move category to pursue in AutoTutor's next utterance. They select a path in the Dialog Advancer Network (DAN) (Person et al. 2000) and adapt a plan to produce AutoTutor's next dialog move. The DAN is an augmented finite-state network which describes discourse pathways that may include one or a combination of the following components: Discourse markers (e.g., "Okay" or "Moving on"), AutoTutor dialog moves (e.g., Positive Feedback, Pump, or Assertion), Answers to questions, or Canned expressions (e.g., "That's a good question, but I can't answer that right now"). It is noteworthy that the dialog management relies on many collaborating sub-modules, all adding and passing information through a state object. This bears some similarity to the "blackboard" architecture (Erman et al. 1980).

AutoTutor refers to multiple pedagogical principles in deciding on the next aspect (expectation) to coach on. One of these is the frontier learning or zone of proximal development principle, selecting the next expectation that builds on what the student knows. It is applied here in selecting the expectation with the highest sub-threshold LSA value. For more information about dialog in AutoTutor, see Chapter 9.

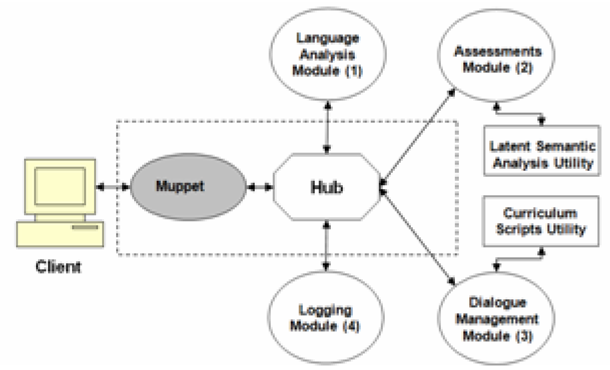

Fig. 8.1 AutoTutor relies on a number of mechanisms in analyzing the student's input and preparing a response. Source: Graesser et al. (2005)

Cognitive aspect. The central element in AutoTutor, its Dialog Management module, relies on inputs from many algorithms and mechanisms. Some of these (the state object that is collaboratively enriched and used by multiple modules; the production rules) do recall cognitive processes; others (those performing natural language interpretation) simply mimic their effects through *artificial AI* mechanisms.

8.3.2.4 Supportive AutoTutor and Shakeup AutoTutor: Decision with Consideration of the Learner's Emotions

Work on extending AutoTutor's reach with respect to the learner has been pursued in the past few years to include the learner's emotional and motivational states (D'Mello et al. 2009). While still under development, two emotion-sensitive versions of AutoTutor represent serious advances in these directions. It is interesting to see how the architecture could be stretched, very simply, to include the new parameters.

Brief description. The newer versions of AutoTutor, called Supportive AutoTutor and the Shakeup AutoTutor, receive not only textual inputs but also cues from complementary channels: posture features from a thin-film pressure pad laid on the chair, and facial feature tracking to monitor facial expressions coming through a camera. Dialog features have also been extended to allow inference of the learner's affect. Fusion of the multiple sensory channels should lead to a more reliable emotion classifier and more adept tutorial responses. The brief feedback AutoTutor provides is only intended to give an appraisal of the learner's latest textual input. "Good job" and "Well done" are examples of positive feedback, while "That is not right" and "You are on the wrong track" are examples of negative feedback. However, this feedback may provoke an emotional response in the learner in non-neutral cases, especially since it is accompanied by an appropriate facial expression from the character. The learner's emotional state is addressed in the tutorial action that follows the feedback, the tutorial move. At this point, the two tutoring variants diverge somewhat. What differentiates the two affect-aware versions is what is considered to be the cause of the emotions. Supportive AutoTutor attributes the source of the emotions to the material or to itself, whereas the Shakeup AutoTutor attributes responsibilities to the learner. This difference affects the wording of the tutorial moves and the conversational style. For instance, we may have "Some of this material can be confusing. Just keep going and I am sure you will get it" from the Supportive one, and "This material has got *you* confused, but I think you have the right idea. Try this…" from the Shakeup one. The way things are expressed will differ as well. In situations where boredom is detected, the Supportive tutor would offer, "Hang in there a bit longer. Things are about to get interesting", where the Shakeup version would instead say, "Geez this stuff sucks. I'd be bored too, but I gotta teach what they tell me".

Decision mechanisms. Attribution theory (Heider 1958; Batson et al. 1995; Weiner 1986), cognitive disequilibrium during learning theory (Piaget 1952; Craig et al. 2004; Festinger 1957) and experts recommendations have been synthesized into new production rules capable of processing emotional cues which reveal some of the emotions believed to exist in relation to learning: boredom, confusion and frustration. The original fuzzy production rules were sensitive to cognitive states of the learner, but not to his emotional states. The newly designed fuzzy production rules map dynamic assessments of the student's cognitive and affective states with tutor actions: feedback delivery (positive, negative, neutral), a host of dialogue moves (hints, pumps, prompts, assertions, and summaries), and facial expressions and speech modulation by AutoTutor’s embodied pedagogical agent (EPA). These are triggered in the following order: (a) feedback for the current answer, (b) an empathetic and motivational statement, (c) the next dialog move, (d) an emotional display on the face of , and (e) emotional modulation of the voice produced by AutoTutor’s text-to-speech engine. Five parameters in the learner model influence the decisions: (a) the current emotion detected, (b) the confidence level of that emotion classification, (c) the previous emotion detected, (d) a global measure of student ability (dynamically updated throughout the session), and (e) the conceptual quality of the student’s immediate response.

Cognitive aspect. Adding emotions to an artificial system brings it a step closer to human information processing. The involved rules have an impact on cognition, and sometimes *are* cognition. Although one could question the way they are implemented, from a naturalistic point of view, the rules are founded on theories of emotions.

8.3.3 Summary of Existing Tutoring Agents

In the systems just described, with the exception of the ACT-R tutors, decision mechanisms are not based on a global theory of cognition – although it is interesting to note that many relate in some measure to high-level human cognition: diagnosing with multiple autonomous error handlers, reaching decisions through a chain of simple decisions, transferring and sharing information by means of a central state object. But these mechanisms are used for their convenience and efficiency, not as part of a global theory of cognition. This is fine, and efficient, as these systems have demonstrated. However, we would like to see how close to human performances a tutoring agent could get if it drew deeper inspiration from its mentor.

ACT-R tutors, for the most part, work within the confines of the ACT-R theory of cognition. The parallel holds for the agent called Steve (Rickel and Johnson 1998; not presented in this chapter), which embodies the Soar theory of cognition, a somewhat similar rule-based approach. Both have shown the feasibility and relevance of the endeavor. Both theories have inspired research, ideas and implementations since the late seventies. However, recent years have seen many proposals for alternative cognitive architectures, and many have been implemented computationally, as Samsonovich illustrates in his overview (Samsonovich 2010). We have singled out one of these, the Global Workspace (GW) theory, because it involves consciousness and offers a complete framework on cognition. We thought its implementation in IDA/LIDA could be extended and form a cognitive tutoring agent. We describe this agent in the coming pages.

8.4 CTS: Designing Cognitive Tutors on Human Bases

The name of the system we present here, CTS (Dubois 2007), stands for "Conscious Tutoring System". Simply put, the agent's processing is based on the conscious/unconscious distinction and on what is called the *access* consciousness, the consciousness phenomenon that renders a piece of information widely accessible to the whole brain. The reader may refer to Block (1995) and (2002) for a clarification of the many levels and types of consciousness. In our implementation, the reference to consciousness is at the functional level. There is no strong claim about CTS being "really" conscious in a human fashion. Not yet... Before embarking on a description of the agent and its architecture, let us see what impact Baars' view on cognition may have on information processing.

8.4.1 Cognition Based on Consciousness, and Some Functional Implications

It cannot be denied that the unconscious/conscious distinction exists. Simple experiences can demonstrate that there are things one can accomplish unconsciously, nearly effortlessly, whereas doing the same operations wilfully, while thinking about the steps, can wear us down pretty fast. Baars presents many such operations (Baars 1997). For instance, try reading a paragraph, then reading it again with the book turned upside down. Do you observe a difference? Of course you do! That shows the difference between doing something consciously and doing it mostly unconsciously. There is a tremendous gain in encapsulating all we can in automated, unconscious processes. However, highly efficient as they are, automated processes are specialized and remain limited in their adaptability. They can take charge only in known cases for which they have been grown. Conscious reflection allows solutions to be created for new situations, improving fitness (Baars 1997; Rolls 1999); it takes time and effort, but it can be done.

When one becomes *conscious* of something, when it "comes to one's mind", *all of one's mind* becomes aware of the information. It becomes available to all of the mind's processes as input for processing, and they can then respond and put forward elements that may be useful in finding a solution. Want it or not, biographical memory will bring back events, especially in fearful situations, and solutions used in situations that are related in some respect; semantic memory will also add "comprehension" of the situation by stringing related concepts to it. Analytic processes may spontaneously propose an overall structure for the event or scenario. Often, all this takes place before one has fully realized what is happening. With all this information at hand, one may decide what is important to address first, choose a multi-step procedure for doing so, and start performing the operations, under the control of the will, until something more important — such as satisfying the urge to sneeze — arises and requires attention.

Which part of the system actually responds is thus determined, and the response is shaped, by the context: current goal and plan, current mood and emotions, complementary information recovered by other systems such as memory or emotional systems, processes currently in the forefront, etc. The situation brought to consciousness is described by coalitions of processors presenting various aspects. Many such coalitions may try to have their information broadcast, but only one can have access to the broadcasting facility at a time, as only one situation can occupy the conscious "space" (one is conscious of only one idea, aspect, situation at a time). Thus, there is "competition" for access to this global workspace. Consciousness appears as a key element in "strong" adaptability.

Baars described this arrangement in his *Global Workspace* (GW) theory of the mind, which he first proposed in 1988. It provides a neuropsychological account and a high-level functional architecture that unifies many of the previous researchers' work on describing and modeling the human mind and consciousness. This theory proposes a general framework describing the essential roles of attention and consciousness in human beings.

Before examining how the agent makes decisions, we will now look at the architecture of CTS, which is rooted in that of IDA, extended to support tutorial decisions.

8.4.2 Conceptual Architecture of CTS

The conceptual architecture of CTS is founded on the ideas put forward by Professor Franklin and his team. Franklin has created an implementation of Bernard Baars' theory in the successive agents Conscious Mattie, IDA, and LIDA. CTS shares many of LIDA's features but is a reimplementation with some differences, some of which are extensions in order to achieve a tutoring agent. The architectures can be considered similar except where noted.

Resources are of two kinds: Codelets and modules (Fig. 8.2). Codelets are internal "micro-agents" that play a specific role in the architecture: perception, metacognitive observation, representing information, monitoring, etc. Those that represent information have an internal structure that allows them to store an item of information along with its classification. Here, we depict Codelets as clouds, since they may work alone or as a team. Modules, shown as rectangles, can be any piece of programming, agent or conventional, that receives information and returns some other element of information to Working Memory through an interface. Links (arrows) describe two kinds of information transfer between these resources: textual information and activation. "Textual information" refers to concepts, facts or sub-information, completed by a category (for instance "camera" + "left"). "Activation" refers to numbers that indicate how strongly individual Codelets or nodes are stimulated. We will come back to activation later on in our tour of the architecture, when we describe the Behavior Network.

All Codelets perform their function in the "unconscious" side of the architecture (the *audience* in the theatre metaphor of the Global Workspace theory), except for the information Codelets, which operate on both sides. Teaming of processors (Codelets) is an essential tenet of the theory: there is both collaboration and competition among the processors. Processors collaborate to describe the

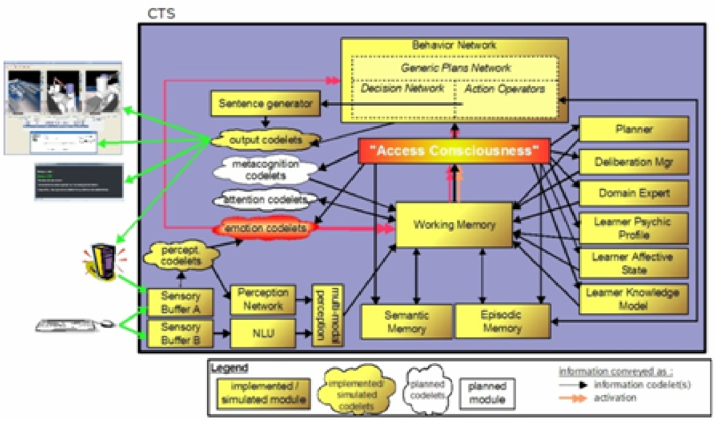

Fig. 8.2 Conceptual architecture for CTS

situation more fully, and then the resulting teams compete for access to broadcasting by consciousness. Central to the architecture are Working Memory and "Access Consciousness", which form the hub for most communications. These are a fundamental element of the architecture, and are placed in the center of the diagram. Working Memory receives input from all sources, and Access Consciousness copies the information selected by Attention to every processing resource.

Processing is organized by a cognitive cycle comprising eight steps (nine for LIDA), going from capturing external stimuli to performing an action. These steps specify when resources are solicited and allowed to contribute information (by sending it to Working Memory). In accordance with Baars' GW theory and experimental results indicating that performing a simple action normally requires about 200 ms, these steps are repeated 5 times per second in an endless cycle. However, since processing the information in a particular subsystem may take more or less time, depending on complexity or other factors, complete treatment of a piece of information may require idle cycling, at least with respect to that specific information (parallel processing of other information in overlapping cycles can, however, continue simultaneously).

Before going into more detail, it is appropriate to interrupt the presentation briefly to present the external environment and application CTS has been adapted for in its current instantiation. This will help in understanding some of the specific goals we have set.

8.4.2.1 CanadarmTutor to Support Astronaut Training

CTS has been developed in a collaborative project with the Canadian Space Agency to coach astronauts in need of training on manipulating the Canadarm2 (Nkambou et al. 2005). Since the robotic arm is a crane built with seven degrees of freedom, it is not easy to predict how a given part will actually move when a joint is activated. What makes matters even more difficult is that the Arm and the

Fig. 8.3 Astronauts have to deal with three camera views and a complementary textual source of information for figuring out Canadarm's situation. NASA

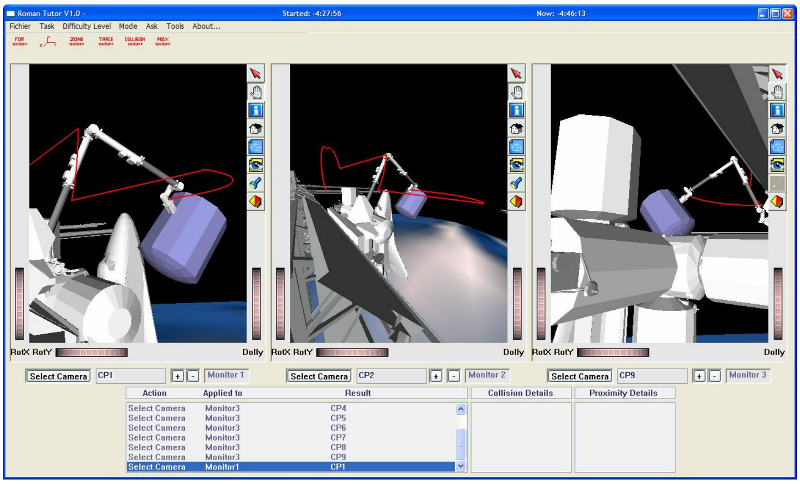

Fig. 8.4 A screenshot of the CanadarmTutor user interface. The red path is the course of action suggested by the path planner.

Space Station can be observed only via three computer monitors that show limited view-provided by cameras located at fixed positions (Fig. 8.3). Astronauts need precisely honed spatial awareness, because they have to operate in a setting very different from terrestrial operations, one in which "up" and "down" have little meaning. However, tutoring resources are scarce. An ITS would provide a welcome support in numerous respects, facilitating learning of concepts and procedures (system of axes, manipulation rules and safe procedures), suggesting appropriate exercises in spatial recognition, and providing the opportunity to practice manipulations with coaching.

Our first implementation of a tutoring system for the International Space Station and its Robotic Manipulator System, Canadarm2, has been designed to give primarily reactive feedback that helps enhance spatial and situational awareness (Roy et al. 2004). CanadarmTutor's tutoring capability initially relied on a path planner that serves as an expert to validate learners' actions. The planner automatically detects student errors in operating the manipulator, produces illustrations of correct and incorrect motions in training, and provides feedback and hints. The path planner we developed acts as a domain expert and can calculate the Arm's moves, consistent with the best available camera views, to avoid obstacles and achieve a given goal. The path planner enables CanadarmTutor to answer several types of student questions, such as "how to....", "what if…", "what's next", "why" and "why not". However, sometimes the solution paths supplied by the path planner prove too complex and difficult for students to follow. The path planner thus does not meet the two basic principles for tutoring agents in procedural tasks: it can't guide the user through an expert solution or recognize the student profile (novice, intermediate, or expert) to offer tailored feedback and assistance. To address these deficits, this capability has thus been extended with our "conscious" cognitive tutoring agent. We will refer to this simulated environment and its user interface (Fig. 8.4) for our coming descriptions and examples. We can now continue with our tour.

There are functional correspondences of modules with the human nervous system and, at a higher level, with functions it accomplishes. On the more biological side, we find Sensory Buffers and Perception, and Semantic and Episodic Memories. At a higher level of correspondence, we find Working Memory, the Learner Models and the Domain Expert. The Behavior Network fits somewhere in between. The inclusion of not-so-biological modules, especially the "peripheral" ones, in roles that may currently be played by external, independent agents, is a concession we are willing to make for two reasons. First, it makes the system easier for people from the ITS world to grasp. Second, it permits any currently operating module or agent to be accepted as an input or extension to CTS. And, if we need a third reason, this is a concession that will eventually be removed when the implementation of the ITS is complete. Then, all the necessary, psychologically and biologically plausible teams of Codelets will replace modules, collectively playing the same roles. Here are the roles Codelets and modules play in reaching decisions.

8.4.2.2 Resources Proper to the Architecture

Senses and Perception. CTS currently possesses two sensing channels for textual inputs, linked to two input buffers, one for the user interface and one for messages from the simulator. After the incoming message has been rearranged in a hierarchical tree by a syntactic parser, the perceptual Codelets (to be explained in the Behavior Network section below) scan the buffer and activate nodes in the Perceptual Network (PN) while simultaneously transferring to them the data they have recognized. These nodes represent the information as concepts the agent can recognize (“Canadarm2 manipulation”, “user answer”, etc.). Nodes already stimulated receive a new partial boost.

Working Memory (WM). In the GW theory, *consciousness is associated with a global workspace in the brain – a fleeting memory capacity whose focal contents are widely distributed ("broadcast") to many unconscious specialized networks* (Baars and Franklin 2003). The architecture depicts WM as a short-term "storage place" (figuratively speaking) where information Codelets from all sources meet, form associations and eventually become part of coalitions that may broadcast. Attention constantly hovers around WM and selects one of the competing coalitions. WM is an information hub between modules, which usually have no direct relation, and for Codelets that need to communicate information (in the form of an information Codelet that they create). Information that lands in WM is copied to Long-Term memories which, like other modules, return the coalition with information complements. The WM in CTS implements important aspects of the Blackboard principles, although modules do not pick what they find of interest to them, but only receive what has been selected by Attention.

Attention. Attention is primarily the mechanism that selects the most prominent (activated) information in WM and supplies it to Access Consciousness for global broadcasting. It corresponds to the *spotlight* used by Baars in his theater metaphor. This selection mechanism is involuntary and uncontrollable. Indeed, one cannot choose what comes to mind, except by making a conscious and difficult effort to *focus* attention, giving preeminence to some specific information. That voluntary focusing of attention is implemented in CTS with attention Codelets that add activation to information Codelets that correspond to their monitoring.

Access Consciousness. Access Consciousness "publishes" the information selected by the Attention mechanism to make it available to all (unconscious) modules. It allows all "actors" to become aware of the situation. During deliberations, this publishing returns to all actors the information contributed by one of them in WM. This mechanism is crucial for the collaboration of the parts in complex, conscious processing, for instance in elaborating a progressive diagnosis.

Transient Episodic Memory (TEM). TEM receives broadcasts and makes a record of them. It can then be probed for recent events by time of occurrence.

Associative Memory (AM). Implements relations between events at the conceptual level. Events can be retrieved via keyword. AM could also contain knowledge of the Domain Model but, for practical reasons, the DM is left as a separate module.

The Behavior Network (BN). Tutorial behavior is stored in a three-tiered Behavior Network. This is based on an original algorithm from Pattie Maes, which she called MASM (Maes' Action Selection Mechanism; Maes 1989), as reimplemented by Franklin and his team. We have reorganized it to include ideas expressed in BEETLE (Zin et al. 2002). It retains Maes' idea of a network with *competence modules*; there are pre- and post-conditions for each node, bidirectional links between them through their preconditions and effects, and both logical conditions and activation levels for deciding on the next behavior to select.

However, Maes had a network of competence nodes performing actions by themselves; Negatu and Franklin (2002) separated decision from action by adding *Codelets* (micro-agents; see further below) related to each node, which allow adapted actions without increasing the number of fixed competence modules. This also made it possible to automatize sequences of frequent actions in a biologically plausible, hebbian manner. Our version of the BN keeps these ideas. It also retains levels of planning and action similar to those in LIDA, but functions somewhat differently, with the help of other mechanisms: a "Planner" and deliberation, which are covered below.

CTS' behavioral knowledge is organized in three groups within the Behavior Network: 1) a Decision Network (DN), 2) a network of Generic Plans (GPN) and 3) a Bag of Actions (BA). The ultimate organization is done via CTS' Planner (4). When the context favors the emergence of a behavior in the DN, a way of implementing the general behavior contextually emerges in the GPN portion of the network, bringing into the Planner the specific steps (actions) that will implement the global behavior; finally, the action is proposed in Working Memory for adaptation and a last-call vote on its execution. The adaptation occurs via a deliberation that summons the various actors in the architecture for their contribution to the information payload.

BN-1) Decision Network: The Decision Network holds nodes representing high-level behaviors and contextually decides on which one to retain.

BN-2) Generic Plan Network: A generic plan contains ways (streams of lower-level actions) of realizing the behaviors. It is comprised of steps that necessarily follow each other, without alternative paths. Generic plans serve two general purposes. Some are high-level plans that orient behavior globally, giving general steps at a coarse level of specification. Others are closer to operations and no further expansion of their steps is required. The latter type either supply a more detailed account of the necessary steps, or present methods for fleshing out a high-level action found in the Decision Network. As an example of a coarse plan needing subspecification, consider the special case of the very generic plan for conducting a learning session, with the steps Beginning, Middle and Ending. Obviously, these steps need more detailed instructions. When the time comes, the Planner submits the step to Working Memory so that it can be elaborated with a more detailed generic plan. As examples of methods, we have the many generic plans describing ways for executing interventions – for instance, after an incorrect operation on the learner's part. Elementary actions are adapted to the context observed at the time the act is performed (specific relevant content, preferences of the current user with respect to optional steps). Deciding on the next thing to do depends partly on the level of activation of a node. The "strongest" node for which all of the *required* preconditions are satisfied is the one that pilots CTS next processing step.

BN-3) Bag of Actions: The Bag of Actions is the repertoire of individual, elementary actions in the form of a collection of Codelets that can be summoned for action. An action may be referred to in many places of the Decision Network, in different plans.

BN-4) Planner: The sub-elements of the BN are ultimately organized through CTS' Planner, which makes it possible to do some planning that is hard to achieve with Maes' algorithm. For instance, in some situations, it may be appropriate to focus temporarily on remediating the learner's faulty understanding, and then resume the presentation or whatever was going on, which in fact amounts to inserting a remediation plan into the presentation plan. Without the Planner, it would be uncertain at best (depending on the BN's concrete implementation) whether the current plan would be pursued when remediation was complete.

Despite the apparent implications of its name in computer terms, the Planner does not decide on anything, it does not *plan*. It simply toys with plans, storing steps and managing the addition, insertion and removal of steps. It implements that portion of Working Memory which holds the list of goals and steps we, as humans, keep alive in our minds.

Codelets. Elementary processing is a fundamental aspect of the architecture, as Baars hypothesized that the mind is a community of simple processes. Franklin used the Copycat concept of *codelet* (Hofstadter and Mitchell 1995) to implement these processes. A Codelet is a light, specialized agent with minimal autonomy. It receives information, discards what it does not recognize, and reacts according to its capabilities. Collectively, a team of Codelets may pursue a higher-level goal, such as interpreting a message in the Sensory Buffer. In our architecture, Codelets are grouped by type of specialty: information, attention, control, or action; or by where they reside or where they look for information: in Perception, Working Memory or the Behavior Network. Here is a description of the varieties:

C-1) Information Codelets: An information Codelet represents or contains information in a specific location, or serves as a transporter of data for communication between modules and Working Memory. Information Codelets may form associations with each other, each one containing a single element within coalitions that provide a richer content. They are short-lived, being created by modules or by some types of Codelets to represent states of fact in Working Memory. When a process becomes automated, they may not be needed anymore, being replaced by a direct communication link between connected actors ("unconscious" communication).

C-2) Action Codelets. Released by the BN's nodes, action Codelets implement the "voluntary" actions of CTS, both on the internal and on the "motor" (outer) side. They act on the environment that is external to CTS, or within CTS. Whereas a Codelet of most other types fulfills its responsibilities in a self-contained fashion and produces an information Codelet, action Codelets act as builders and modifiers in their environment. For instance, some send messages to other computers – say, to request video replay to the simulator; some others display messages in the user interface; etc.

C-3) Attention Codelets. This is a category that encompasses all of the processes that monitor other processes or communications. These Codelets serve in specialized roles: as reinforcers of information appearing in Working Memory (implementing voluntary attention), as processes overseeing the result of an operation in order to declare its success or failure (*expectation Codelets*), or as processes monitoring the global operations of CTS (implementing metacognitive processes).

C-4) Perception Codelets. Codelets in this group are collectively responsible for recognizing elements entering the Sensory Buffers and creating an interpretation by activating the information Codelets that form the perceptual network. They are hybrids, having the perceptual role of attention Codelets (they monitor Sensory Buffers in hope of recognizing something there), and the action role of action Codelets (they act on other Codelets, activating them).

C-5) Emotion Codelets. Collectively, they form CTS' *pseudo-amygdala* (the amygdala being an organ involved in learning emotional associations). Some of them are connected to Perception Codelets on one side, and to emotional motivator Nodes in the BN on the other. They evaluate the low-level features detected by the perception Codelets and stimulate accordingly the emotional motivator Codelets located in the BN (see below). Others are concerned with what comes to Working Memory, adding to relevant information Codelets activation with emotional tag.

C-6) State Codelets. State Codelets are hybrids similar to perception Codelets except that they exist within the Behavior Network, where they represent the current context as observed. They hear broadcasts from Access Consciousness and turn on when they recognize information. As a logical context, they form the required preconditions for behavior nodes to fire. As analog context, they supply activation to the single or multiple nodes they are attached to (which may form part of the DN, the GPN or the BA). Thus, the context determines what operation is most relevant, according to what Attention has selected for broadcasting. Some of the state Codelets are not absolute prerequisites for an action, but rather represent optional conditions: preferences, favorable circumstances, etc. These simply *add* activation to a node (or take away some, in the case of unwanted situations), increasing (lowering) the likeliness that this node will be favored.

C-7) Motivator Codelets. Motivators are the other source for activation supplied to nodes in the BN. Some Motivators implement values and principles dear to the agent; some connect to emotion Codelets and represent them in the BN. Like the optional state Codelets, they are part of a richer context and potentially add to the activation of a node, driving action selection "from the top", with respect to the agent's goals, and from situational analysis (emotions). They are variably stimulated by the perceived events or inferred beliefs they correspond to, and they pour activation into the Behavior nodes that connect to them. Motivator Codelets can be made more or less sensitive to excitators, and sets of parameters organized as profiles may form various personalities for CTS.

8.4.2.3 Resources External to the Architecture

External resources use privileged communication channels that make them virtually part of the architecture. An interface bidirectionally transforms information into information Codelets. These modules and external agents react to information they "hear" from the broadcasts just the same; they also volunteer information when they deem appropriate, eventually priming some “motivators” in the agent (see section C-7).

Domain Expert (DE). Knowledge of the domain and evaluation of the learner's actions has been delegated to an external agent based on MIACE (Mayers 1997) and developed by Fournier-Viger (see (Najjar et al. 2005) for details on how this agent encodes its knowledge). The Domain Expert receives Access Consciousness' broadcasts and then evaluates the learner's operations and answers. It communicates with CTS through an interface which does the bidirectional transformation to and from information Codelets.

Learner Model (LM). The learner model is a distributed one. Its static part, the Learner Profile (LP), contains psychological information, including the learner's learning style. Its dynamic part is twofold: the Learner Affective State (LAS) tracks the learner's mood and emotional state, while the Learner Knowledge Model (LKM) holds facts and the learning history, infers knowledge and trends, and computes statistics.

8.4.3 Global Decision Process in CTS

CTS makes decisions and operates on the basis of three fundamental processes: cognitive cycle, activation transfer and deliberation. They interact and create the global decision mechanism.

Activation. This is the first level of a systemic decision, the most elementary. Much in the architecture processing is activation-based, with decisions emerging both through voting during deliberations and through competition (in Working Memory and in the Behavior Network). Behavior nodes accumulate activation from state and motivator Codelets; information Codelets bring value to a coalition in WM; coalitions having enough activation occupy WM and compete for broadcasting (for becoming the "conscious" information) based on their global activation level. Information Codelets progressively lose their activation and eventually disappear from WM, and cannot anymore be part of an upcoming broadcasting.

Cognitive Cycle. The cognitive cycle is the next level of decision, where all resources get the chance to participate. In CTS/LIDA, it is a detailed version of the standard perceive-process-act cycle. Eight steps (nine in the case of LIDA) specify an order for actors to intervene and for the processing to take place in the Behavior Network (please look at Fig. 8.5): 1) Perception, 2) Percept becomes part of WM, 3) LT memories see the info and return related info, 4) Competition for consciousness of all coalitions in WM, 5) Broadcast of the strongest (coalition of) information, 6) Recruitement of behavioral resources (activation by state and motivator Codelets), 7) Selection (among all activated Behaviors), and 8) Acting (releasing action Codelets).

In one cycle, there is one and only one broadcast of information. But consecutive cycles may overlap, due to the highly distributed processing, at least in the human model. For instance, sensing may take place at the same time as memories are searching for information about a previous broadcast. In any case, a cycle is the elementary unit of conscious processing. However, processing information in a meaningful or useful manner may require more than one cycle, for iterative refinement of an idea or a plan, or for pondering alternatives. Thus, an *action* may not happen at every cycle, in the sense of taking an action on the exterior world, until the *deliberation* has ended.

Deliberation constitutes the most complex decision processing. Except in very special activities (dangerous situations, meditation), context is not a given from the external or the internal environment alone; ideas, points of view, hypotheses and preferences usually supplement the raw information from the outside to form a richer environment for human decisions. Deliberation in CTS is the gathering and analysis of information, including "opinions", through reflection loops made possible by Access Consciousness and its broadcasting of selected information to all actors in the architecture. In CTS, the Behavior Network is not the lone decision-maker that dictates what to do next. All the actors have input, sometimes signaling a situation that may need attention, sometimes opposing a suggested course of action, supplying alternatives, or helping to adapt the proposed intervention. When a proposition is opposed, new suggestions are sought, initiating a new round of deliberative cycles.

The three levels are interlinked, activation passing sustaining the selection of information toward broadcasting, and broadcasting allowing the multiple loops that form a deliberation.

Decisions in CTS may involve another aspect that we have not mentioned until now, a facet that is very humanly: emotions. They are the last component of CTS involved in decision-taking. We now turn our attention to this aspect.

8.4.4 Emotions in the Agent

CTS' architecture allows decisions to be made under the influence of emotions in a biologically plausible manner. Part of its apparatus mimics the involvement of the amygdala in learning and producing spontaneous emotional response. A good description of this segment can be found in (Faghihi et al. 2008). Another part memorizes emotional appraisal along with the factual content of an event, which has repercussions later, when memories are probed and brought back into Working Memory, influencing decision making. We will take a brief tour of these systems below, starting with the amygdala.

Amygdala's emotional involvementThe amygdala is known to play a role in emotional responses. Research in neurobiology suggests that there are two (or more) routes from perception to action (Rolls 1999). The first route is short and direct: information flows from the sensory thalamus directly to the amygdala, and from the amygdala to basal ganglia. Motor reaction is then rapid, even impulsive, as the received information is not interpreted by other brain structures (Squire and Kandel 1999). In the second route, information from the external environment is analyzed by various cortical areas (primary sensory cortex, unimodal associative cortex, polymodal associative cortex). It is then sent to the hippocampus, presumably for memory retrieval and temporary storage. All of this processing serves to give meaning to the external stimulus and link it to other events (through the hippocampus' episodic memory), before it returns to the amygdala for emotional appraisal and response.

These two routes are implemented in CTS and work in parallel. Emotions that become conscious as well as those that remain unconscious (follow the "short" route) excite *emotional nodes* located in the Behavior Network. The short route influences the selection of Behaviors (by priming some of them), even sometimes squarely causing a Behavior to fire before the longer, analytic process comes to a conclusion and selects the "logical" choice.

We'll look at the involvement of emotions in conscious action decisions before contrasting it to the unconscious influence of emotions. You may look at **Error! Reference source not found.** 8.6 to accompany the coming descriptions. It illustrates both the cognitive cycle in CTS and emotional processing. As we will describe, emotions get involved at many points in the cycle.

Emotions influencing *conscious* decisions in CTSIn step 1 of the cognitive cycle, the collective interpretation work made by the perception Codelets results in a percept, which is temporarily stored as the active nodes of the Perceptual Network. The percept becomes part of Working Memory as a single network of information Codelets (step 2), but is looked upon by the Coalition Manager as multiple possible coalitions of Codelets, each describing various aspects of the situation. During the Coalition Manager's processing, emotion Codelets inspect the informational content of each coalition and infuse it with a level of activation proportional to its emotional valuation. They also graft an information Codelet that tags them with the classification of the emotional energy. The activation supplement increases the likelihood of some coalitions to draw Attention to themselves. The emotionally-reinforced coalition may then become the information CTS considers in a forthcoming deliberation (if that coalition is selected and broadcast). In a nutshell, emotions influence the focus of attention. For example, imagine the situation where the learner does not acknowledge repeated prompts from CTS. The tutoring agent has learned in past occasions that this may be a sign of annoyance, so it should "feel" sorry for taking this course of action and tag this situation as negative.

When an emotionally-reinforced coalition is selected (step 4) and broadcast (step 5), its content is received by the Transient Episodic Memory which stores it with its emotional tag and value. When TEM is later probed for concepts or events relating to the current situation, emotional valuation heightens the salience of stored events, influencing which is restored to Working Memory (step 3). Here, too, emotions influence the decision making in CTS by modifying what is memorized and recalled.

The broadcast coalition is also received by the Behavior Network. Some emotional motivator Codelets should react to the emotional aspect of the information and infuse the Behavior nodes they are connected to with some activation, in this case inhibitory, lowering their probability of further being selected (step 6). This is a third involvement of emotions in CTS.

These direct (reaction to WM content) and indirect (memory recall) emotional interventions in WM is how the Amygdala gets involved in CTS' analytical long route.

Emotions causing reflex actions. In the short route (Fig. 8.5, ESR rectangles), emotional involvement occurs sooner, but based on a rough evaluation of the

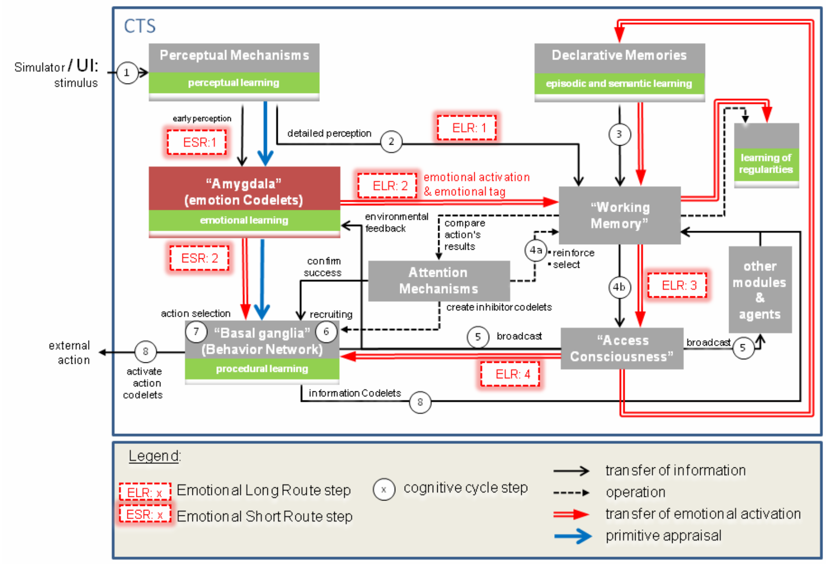

Fig. 8.5 CTS' cognitive cycle states when the various actors of the architecture may do their processing of information. Emotions impact CTS cognition at various points. This influence may happen unconsciously, either to promote the selection of a specific behavior, of to enhance memorization.

situation. The perception Codelets connect in parallel both to the PN and to emotion Codelets. Before their collective interpretation creates a rich and complete interpretation of the stimulus, their individual activation is copied directly to emotional motivator Codelets (themselves connected directly to acts in the BN) the first leg of the short route. This direct stimulation finds its application primarily in dangerous situations. When low-level basic information coming from the perception Codelets is recognized as highly dangerous, there is no time to think, and the emotional Codelets will likely force an action to fire in the Action Bag of the BN. This usually makes CTS act before it has had time to become "conscious" of the situation and consciously plan a course of action. This is the fourth emotional involvement. As a concrete example, say CTS learns that an input from the learner saying "I'm going to quit" after a series of failures shortly precedes his shutting off the session. This contradicts the tutor's goal of seeing the learner conclude an exercise successfully, and is interpreted as saddening. It is very much undesirable. This negative association, learned by an emotional Codelet in the past, will fire an immediate, simple reaction (a short message) allowing some time for CTS to reach a more involved, deliberated decision about what to do next. This implicit reaction corresponds to the process described by Squire and Kandel (1998).

But there is a fifth way emotions may have an impact: they modulate actions intensity. Although not implemented yet in CTS (the tutor currently has means of action limited to text boxes, canned demonstrations and sequences replay), we foresee motor actions being modified in their amplitude and rapidity by the emotional value of the coalition that sparked them.

Emotional reinforcement. Instinctive reactions do not block the longer analytical process; they merely step in first. Eventually, however, a better idea of the situation is worked out (following complete interpretation by Perception) and may come to consciousness, allowing normal action selection to take place. The instantaneous, emotional reflex initiated by the "short route" is then compared by expectation Codelet(s) to the action that has been selected in the BN in the "normal" way. If the two roughly correspond, the expectation Codelet does two things: it grafts an information Codelet with a strong negative valence onto the action proposition in WM (to prevent a repetition of the action); it also grafts a confirmation stating the correspondence, which will serve, when broadcast, as a reinforcer to the emotion Codelet(s) that was or were instrumental in setting off the reflex. In effect, this will cause CTS' "pseudo-amygdala" to reinforce its relevant "rule". However, when the initial reaction diverges from the behavior proposed by the more methodical analysis, some mechanism must be in place to control the incorrect reflex action. Indeed, according to Rolls (2000), the amygdala never unlearns a rule, always reacts to the same stimulus and needs cortical interventions to temper it (Morén 2002). From a neurological point of view, control over actions is the role of cortical areas. We implement the cortical controls here with *inhibition Codelets*, which are generated by the expectation Codelet(s) that spotted the discrepancy. These Codelets attach themselves to a behavior node in the action selection mechanism (BN) and constantly subtract activation from it.

8.4.5 Summary on CTS

It would be nice to have a tutoring agent capable of learning the domain on its own, adapting to various operational settings and offering appropriate tutoring for different types of knowledge, transferring relevant strategies and tutoring tactics to the new instructional constraints. CTS' proposes many of the necessary mechanisms (some of them not described here), and its architecture allows for extensions as well as adaptations to include biological features. The current prototype implements many principles of the GW theory and, while we keep adding and improving aspects of the agent, it already demonstrates that artificial cognition works. The separation unconscious vs. conscious and a functional consciousness offer a promising avenue of advancement.

8.5 Coming Trends: A Glimpse of the Future

In all of the systems presented, including CTS, the cognitive processing is accomplished at a high level of representation, which helps in the design and implementation aspects. However, artificial systems still have very limited capabilities compared to real-world requirements: their ability to interpret visual inputs (recognizing objects, interpreting scenes) is weak, they have little insight about focusing their attention on what counts, their linguistic capabilities remain wanting, they are still awaiting a reasonable foundation of knowledge for real-world "common-sense" reasoning and lack the framework for integrating diversified knowledge (Dutch et al. 2008; Samsonovich et al. 2008). They also fare poorly with regard to autonomy, in its widest ramifications, are pretty clumsy about emotional intelligence (Picard 2007) and remain very limited in their social ability, which may impede their potential for finding solutions and knowledge whose relevance has been tested and validated by other artificial agents. Fidelity to the human model remains remote, biologically speaking, and the explanatory power of the architectures is limited. A match closer to biology may be wished for. But does any of this really matter? Why insist on copying the way humans process information, and why even attempt to reproduce it at the lower levels of processing?

From a negative standpoint, applying non-human-readable artificial neuron networks to accomplish some chain of processing is much harder and not within anyone's reach, at least given the current state of authoring tools. The same may be true even for symbolic-level tools and architectures. On the positive side, having "living" models of biological theories is a good way of testing them, finding their limits and loopholes. In this way, research gains improved theories, and also becomes better able to predict the impact of surgical or psychological interventions. These are some of the reasons prompting the pursuit of biologically-inspired cognitive architectures, known increasingly under the acronym BICA.

We will now present a brief Q&A on the subject, and end by revisiting the question that is the focus of this chapter: "How will this impact an artificial intelligent tutor?"

How does BICA differ from current cognitive architectures? Well, it depends on where an architecture is positioned on the "biological fidelity scale". Research has produced, and is maintaining and developing, systems that use representation and computation means at varying degrees of "natural fidelity". Many existing architectures, such as ACT-R and LIDA/CTS, seek mind-level natural validity, with some functional correspondence to brain structures and mathematical reproduction of neural learning; others, such as CLARION (Sun 2004) and SOVEREIGN (Gnadt and Grossberg 2008), tend to add a lower-level biological *explanatory* level (with biologically-inspired structures and processing that cause and explain the results, with respect to biologically-inspired parameters and mathematical computations that justify the results). In other words, biologically-*inspired* architectures are not constrained to purely and essentially reproducing biological processing; they may incorporate "*artificial* AI" techniques, some being more difficult to justify in biological terms than others. What is really new is the interest for creating complete architectures that rely on low-level fidelity to biological processes ("complete architectures" meaning, here, "systems that cover all of the human nervous system"). Samsonovich proposes an excellent feature-by-feature overview of many cognitive architectures (online at http://members.cox.net/bica2009/cogarch/architectures.htm), showing, among other things, their correspondence with brain structures and their level of modeling. The reader is also referred to the article by Dutch *et al.* (2008) reviewing current symbolic cognitive architectures and emerging paradigms.

What is to be gained with BICA? One thing that has constantly eluded AI research efforts is artificial *general* intelligence (AGI), that is, the ability to adapt and react to unknown situations in unconstrained domains, by applying existing knowledge in new ways or acquiring new knowledge. AI has produced solutions that often outperform humans in specific tasks or on limited problems, but they are either not meant for general applicability or cannot transfer to other domains. What has set humans apart from other species is, for one, our ability to adapt, but more than that, our ability to learn and to set goals for this learning through self-monitoring, self-regulation and valuation of goals. Being able to judge its own lack of knowledge, to determine ways for acquiring that knowledge, to store it in a meaningful and usable manner, and have the motivation to do these things, would allow an artificial system to walk in the footsteps of a human baby and become better at whatever tasks it has been assigned, on its own. Researchers believe that strong ties to the human model, at the biological level, are currently our best bet and that we should pursue "strongly" biology-inspired cognitive architectures (Berg-Cross 2008).

What impact will it have on artificial teachers? One trait that is fundamental in a good teacher is his (her) ability to learn and better adapt to his (her) students. While this sounds simple enough, in reality it calls for a collection of abilities that are well integrated in good human teachers/tutors, both as high-level processes and as low-level mechanisms sustaining them. Reproducing their interplay requires an architecture capable of capturing and making sense of subtle signals, and bringing together various information sources, decision centers, and motivational mechanisms. Motivation fuels a good part of the process of taking action, and this naturally connects to values and emotions. Being interested in the learner's performance may be woven into the fabric of an ITS, but deciding to devote some attention to the learner's emotions and motivational state at some point, rather than solely to his knowledge state, is a more of a human game. And it is touchy. Striking the right balance in different situations, with a variety of learners, requires dynamic adaptation, and dynamic learning too, all done under the supervision of metacognition. Each situation has something to teach about the efficiency of attempts with respect to a specific learner (read: his personality profile). It is only through constant learning (and self-supervision) that an artificial teacher can improve. At some point, human designers can no longer be of much help. The agent has to see for itself, make attempts and gauge the results. Samsonovich and colleagues' Cognitive Constructor relies on these premises (Samsonovich et al. 2008).

We would love to see scenarios where the artificial teaching agent decides to take innovative actions to get out of a situation judged as an impasse. For instance, the agent might conclude that the learner just doesn't get it (results do not improve significantly) in spite of all the effort on both sides, and begin showing demotivation and anger. A clueless agent could keep repeating the same theory and exercises over and over again, or it could simply go on with the next subject after storing the poor grades in its database. An AGI teacher might, instead, detect distress and anger, evaluate the signals as deserving priority of attention, "feel" compassion at the sight of the effort the learner has expended and disappointment at the results, and become motivated ("feel" the urge) to devote more resources to finding a solution. In view of the global context (learner profile, past performances, valid goals), it might then try applying current abilities, such as communicating over the Internet, to a search for pedagogical solutions, or decide to consult other similar pedagogical agents about their proven solutions to similar situations. A reward system ("pleasure") would reinforce the tendency to use the newly created creative path or, conversely, increase the resistance to it. In the latter case (unsuccessful new solution path), the motivation for supporting the learner would remain highly activated and another plan for the creative application of current abilities would therefore be devised. In that scenario, we see the usefulness of emotions in orienting decisions and generating amplitude in the reaction. Besides endowing the agent with more social grace, emotions are a quick way of integrating multiple sources of information – somewhat akin to intuition – , reinforcing decisions and orienting the next action.

As we are coming to understand more and more, emotions are not irrelevant to performance; they are not simply something that's nice to have. They truly are part of intelligence. Plain logic and hard facts usually do not suffice to build lasting relationships among humans. At times, it may become irritating to deal with a senseless or otherwise poker-faced teacher. An ITS meant for long-term interventions (accompanying students year after year) needs to address learners' emotions. Empathy is sometimes sought, and a lack of it may demotivate the learner. Thus, the artificial agent should be able to detect, aptly deal with, and display emotions. It takes time, practice and fine observation to find what works in which situation, and it takes the appropriate memory structures to store emotions in a significant and useful way. Humans possess these, and they work quite well in most situations. Along with our perception and communication abilities, our metacognitive mechanisms and our goal-setting autonomy, they are yet another good reason to take a long, hard look at the human model.

Acknowledgements. We gratefully acknowledge the work and helping hand of many current and past members of the GDAC laboratory and other close collaborators in the realization of CTS and Canadarm Tutor: Patrick Hohmeyer, Pierre Poirier, Usef Faghihi, Mohamed Gaha, Philippe Fournier-Viger, Khaled Belghith.

References

Aleven, V., McLaren, B.M., Sewall, J., Koedinger, K.R.: A New Paradigm for Intelligent Tutoring Systems: Example-Tracing Tutors. International Journal of Artificial Intelligence in Education 19, 105–154 (2009)

Anderson, J.R.: Language, memory, and thought. Erlbaum, Hillsdale (1976)

Anderson, J.R.: Rules of the Mind. Erlbaum, Hillsdale (1993)

Anderson, J.R., Corbett, A., Koedinger, K.R., Pelletier, R.: Cognitive tutors: Lessons learned. J. Learn. Sci. 4(2), 167–207 (1995)

Anderson, J.R., Gluck, K.: What role do cognitive architectures play in intelligent tutoring systems? In: Klahr, D., Carver, S.M. (eds.) Cognition & Instruction: Twenty-five years of progress. Lawrence-Erlbaum Associates, Mahwah (2001)

Baars, B.J.: In the Theater of Consciousness. The Workspace of the Mind. Oxford University Press, New York (1997)
Baars, B.J., Franklin, S.: How conscious experience and working memory interact. Trends in Cognitive Sciences 7, 166–172 (2003)
Batson, C.D., Turk, C.L., Shaw, L.L., Klein, T.R.: Information Function of Empathic Emotion - Learning That We Value the Others Welfare. J. Personal Soc. Psychol. 68, 300–313 (1995)
Berg-Cross, G.: Introduction to Biological Inspiration for Intelligent Systems. In: Madhavan, R., Messina, E.R. (eds.) Performance Metrics for Intelligent Systems (PERMIS) Workshop (2008)
Blessing, S.B.: A Programming by demonstration authoring tool for model-tracing tutors. Int. J. Artif. Intell. Educ. 8, 233–261 (1997); reprinted in Murray, T., Blessing, S., Ainsworth, S. (eds) Authoring tools for advanced technology educational software. Kluwer Academic Publishers, Dordrecht, pp. 93–120 (2003)
Block, N.: On a Confusion About a Function of Consciousness. Behav. Brain Sci. 18, 227–247 (1995)
Block, N.: Some Concepts of Consciousness. In: Chalmers, D. (ed.) Philosophy of Mind: Classical and Contemporary Readings. Oxford University Press, New York (2002)
Carbonell, J.R.: AI in CAI: An Artificial Intelligence Approach to Computer-Aided Instruction. IEEE Trans. Man-Mach. Syst. 11, 190–202 (1970)
Carnegie Learning, Timeline of Cognitive Tutor History (2010), `http://ctat.pact.cs.cmu.edu/index.php?id=timeline` (accessed)
Choksey, S., Heffernan, N.: An Evaluation of the Run-Time Performance of the Model-Tracing Algorithm of Two Different Production Systems: JESS and TDK. In: Tech. Rep. WPI-CS-TR-03-31. Worcester Polytechnic Institute, Worcester, MA (2003)
Cohen, P.A., Kulik, J.A., Kulik, C.C.: Educational outcomes of tutoring: A meta-analysis of findings. Am. Educ. Res. J. 19, 237–248 (1982)
Collins, A., Warnock, E.H., Aiello, N., Miller, M.L.: Reasoning from incomplete knowledge. In: Bobrow, D., Collins, A. (eds.) Representation and Understanding: Studies in Cognitive Science. Academic Press, New York (1975)
Conati, C., Gertner, A., VanLehn, K.: Using Bayesian networks to manage uncertainly in stu-dent modeling. User Modeling & User-Adapted Interaction 12(4), 371–417 (2002)
Craig, S.D., Graesser, A.C., Sullins, J., Gholson, J.: Affect and learning: An exploratory look into the role of affect in leaning. J. Educ. Media 29, 241–250 (2004)
Crick, F., Koch, C.: Towards a neurobiological theory of consciousness. Seminars of the Neurosciences 2, 263–275 (1990)
Crick, F., Koch, C.: A framework for consciousness. Nature neuroscience 6(2) (2003)
Dehaene, S., Naccache, L.: Towards a cognitive neuroscience of consciousness: basic evidence and a workspace framework. Cogn. 79, 1–37 (2001)
Dennett, D.: Are we explaining consciousness yet? Cogn. 79, 221–237 (2001)
Dennett, D., Akins, K.: Multiple drafts model. Scholarpedia 3(4), 4321 (2008) (accessed 21, March 2010)
D'Mello, S.K., Craig, S.D., Fike, K., Graesser, A.C.: Responding to Learners' Cognitive-Affective States with Supportive and Shakeup Dialogues. In: Jacko, J.A. (ed.) Human-Computer Interaction. Ambient, Ubiquitous and Intelligent Interaction. Springer, Heidelberg (2009)
Dubois, D.: Constructing an agent equipped with an artificial consciousness: Application to an intelligent tutoring system. PhD Thesis, Université du Québec à Montréal (2007)

Duch, W., Oentaryo, R.J., Pasquier, M.: Cognitive architectures: where do we go from here? Frontiers in Artificial Intelligence and Applications 171, 122–136 (2008)

Erman, L.D., Hayes-Roth, F., Lesser, V.R., Reddy, D.R.: The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. Computing Surveys 12(2), 213–253 (1980)

Faghihi, U., Poirier, P., Dubois, D., Gaha, M., Nkambou, R.: Implementation of Emotional Learning for Cognitive Tutoring Agents. In: Gelbukh, A., Morales, E.F. (eds.) MICAI 2008. LNCS (LNAI), vol. 5317. Springer, Heidelberg (2008)

Festinger, L.: A theory of cognitive dissonance. Stanford University Press, Stanford (1957)

Fodor, J.A.: The Language of Thought. Thomas Crowell, New York (1975)

Fodor, J.A.: The modularity of the mind. MIT Press/Bradford Books, Cambridge (1983)

Gnadt, W., Grossberg, S.: SOVEREIGN: An autonomous neural system for incrementally learning planned action sequences to navigate towards a rewarded goal: Technical Report CAS/CNS-TR-07-015. Neural Networks 21, 699–758 (2008)

Graesser, A.C., Millis, K., Chipman, P., Cai, Z., Wallace, P., Britt, A., Storey, J., Wiemer, K., Magliano, J., D'Mello, S.: A Demonstration of ITSs That Promote Scientific Inquiry Skills: Critical Thinking Tutor and ARIES. In: Proceedings of ITS 2008 Demo Session (2008)

Graesser, A.C., Olney, A., Haynes, B.C., Chipman, P.: AutoTutor: A cognitive system that simulates a tutor that facilitates learning through mixed-initiative dialogue. In: Forsythe, C., Bernard, M.L., Goldsmith, T.E. (eds.) Cognitive systems: Human cognitive models in systems design. Erlbaum, Mahwah (2005)

Graesser, A.C., Jeon, M., Dufty, D.: Agent technologies designed to facilitate interactive knowledge construction. Discourse Process 45, 298–322 (2008)

Graesser, A.C., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R.: The TRG, Auto Tutor: A simulation of a human tutor. J. Cogn. Syst. Res. 1, 35–51 (1999)

Heffernan, N.T.: Intelligent tutoring systems have forgotten the tutor: adding a cognitive model of human tutors. Dissertation. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA (2001)

Heffernan, N.T., Koedinger, K.R., Razzaq, L.: Expanding the model-tracing architecture: A 3rd generation intelligent tutor for Algebra symbolization. The International Journal of Artificial Intelligence in Education 18(2), 153–178 (2008)

Heider, F.: The Psychology of Interpersonal Relations. John Wiley & Sons, New York (1958)

Hofstadter, D.R., Mitchell, M.: The copycat project: A model of mental fluidity and analogy-making. In: Hofstadter, D. (ed.) The Fluid Analogies Research group. Fluid Concepts and Creative Analogies. Basic Books, New York (1995)

Jackson, J.V.: Idea for a Mind. Siggart Newsletter 181, 23–26 (1987)

Koedinger, K.R., Aleven, V.: Exploring the assistance dilemma in experiments with Cognitive Tutors. Educ. Psychol. Rev. 19(3), 239–264 (2007)

Langley, P., Laird, J.E., Rogers, S.: Cognitive architectures: Research issues and challenges. Cogn. Syst. Res. 10, 141–160 (2009)

Lynch, C., Ashley, K., Aleven, V., Pinkwart, N.: Defining Ill-Defined Domains; A literature survey. In: Aleven, V., Ashley, K., Lynch, C., Pinkwart, N. (eds.) Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains at the 8th International Conference on Intelligent Tutoring Systems (2006)

Maes, P.: How to do the right thing. Connect. Sci. J. 1(3), 291–323 (1989)

Mayers, A.: Miace: Une architecture théorique et computationnelle de la cognition humaine pour étudier l'apprentissage. Thèse. Université de Montréal, Montréal (1997)

Merrill, D.C., Reiser, B.J., Merrill, S.K., Landes, S.: Tutoring: Guided Learning by Doing. Cogn. Instr. 13(3), 315–372 (1995)

Minsky, M.: The society of Mind. Simon & Schuster, New York (1985)

Mitrović, A., Mayo, M., Suraweera, P., Martin, B.: Constraint-Based Tutors: A Success Story. In: Monostori, L., Váncza, J., Ali, M. (eds.) IEA/AIE 2001. LNCS (LNAI), vol. 2070, p. 931. Springer, Heidelberg (2001)

Mitrovic, A., Koedinger, K.R., Martin, B.: A comparative analysis of cognitive tutoring and constraint-based modeling. In: Brusilovsky, P., Corbett, A.T., de Rosis, F. (eds.) UM 2003. LNCS, vol. 2702. Springer, Heidelberg (2003)

Morén, J.: Emotion and learning: A Computational Model of the Amygdala. In: Meyer, Berthoz, Floreano, Roitblat, Wilson (eds.) From Animals to Animats 6: Proceedings of the 6th International Conference on the Simulation of Adaptive Behaviour.. MIT Press, Cambridge (2000)

Najjar, M., Fournier-Viger, P., Mayers, A., Hallé, J.: Tools and Structures for Modelling Do-main/User Knowledge in Virtual Learning. In: Proceedings of the 16th AACE World Confer-ence on Educational Multimedia, Hypermedia & Telecommunications, ED-MEDIA 2005 (2005)

Negatu, A., Franklin, S.: An action selection mechanism for 'conscious' software agents. Cogn. Sci. Q. 2, 363–386 (2002)

Neisser, U.: Cognitive psychology. Prentice-Hall, Englewood Cliffs (1967)

Newell, A.: Production systems: Models of control structures. In: Chase, W.G. (ed.) Visual information processing. Academic Press, New York (1973)

Newell, A.: Unified theories of cognition. Harvard University Press, Cambridge (1990)

Nkambou, R., Belghith, K., Kabanza, F., Khan, M.: Supporting Training on Canadarm Simula-tor using a Flexible Path Planner. In: Proceedings of the 12th International Conference on Ar-tificial Intelligence in Education (AIED). IOS Press, Amsterdam (2005)

Norman, D.A., Gentner, D., Stevens, A.L.: Comments on learning: schemata and memory or-ganization. In: Klahr, D. (ed.) Cognition and Instruction. Lawrence Erlbaum Associates, Hillsdale (1976)

Ohlsson, S.: Constraint-based Student Modeling. In: Student Modeling: the Key to Individualized Knowledge-based Instruction. Springer, Berlin (1994)

O'Shea, T.: Self-improving Teaching Systems: an Application of Artificial Intelligence to Computer-aided instruction. Birkhauser Verlag, Basel (1979)

O'Shea, T., Sleeman, D.H.: A design for an adaptive self-improving teaching system. In: Rose, J. (ed.) Advances in Cybernetics. Gordon and Breach Publishers, London (1973)

Person, N.K., Bautista, L., Kreuz, R.J., Graesser, A.C., TRG: The Dialog Advancer Network: A Conversation Manager for AutoTutor. In: Proceedings of the workshop on modeling human teaching tactics and strategies at the Intelligent Tutoring Systems 2000 conference, University of Quebec, Montreal (2000)

Piaget, J.: The origins of intelligence. International University Press, New York (1952)

Picard, R.W.: Toward Machines with Emotional Intelligence. In: Matthews, G., Zeidner, M., Roberts, R.D. (eds.) The Science of Emotional Intelligence: Knowns and Unknowns. Oxford University Press, Oxford (2007)

Quillian, M.R.: Semantic memory. In: Minsky, M. (ed.) Semantic information Processing. MIT Press, Cambridge (1968)

Rickel, J., Johnson, W.L.: STEVE: A Pedagogical Agent for Virtual Reality, In: Proceedings of the Second International Conference on Autonomous Agents (1998)

Rolls, E.T.: The brain and emotion. Oxford University Press, New York (1999)

Rolls, E.T.: Neurophysiology and functions of the primate amygdala, and the neural basis of emotion. In: Aggleton, J.P. (ed.) The Amygdala: a Functional Analysis. Oxford Univ. Press, Oxford (2000)

Roy, J., Nkambou, R., Kabanza, F., Hartman, L., Jang, T.-S.: Supporting Spatial Awareness in Training on a Telemanipulator in Space. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) ITS 2004. LNCS, vol. 3220, pp. 860–863. Springer, Heidelberg (2004)

Samsonovich, A.V., Kitsantas, A., Dabbagh, N.: Cognitive Constructor: A biologically-inspired self-regulated learning partner. In: Samsonovich, A.V. (ed) Biologically Inspired Cog-nitive Architectures. AAAI Technical Report FS-08-04. AAAI Press, Menlo Park (2008)

Samsonovich, A.V.: Feature-by-Feature Comparison of Cognitive Architectures (2010), `http://members.cox.net/bica2009/cogarch/architectures.htm` (Accessed March 2, 2010)

Self, J.A.: Student models in computer-aided instruction. Int. J. Man-Mach. Stud. 6, 261–276 (1974)

Squire, L.R., Kandel, E.R.: Memory: From Mind to Molecules. WH Freeman & Co., NY (1999)

Stevens, A.L., Collins, A.: The goal structure of a Socratic tutor. In: Proceedings of the National ACM Conference, Seattle, WA (1977)

Sun, R.: The CLARION cognitive architecture: Extending cognitive modeling to social simulation. In: Sun, R. (ed.) Cognition and Multi-Agent Interaction. Cambridge University Press, New York (2004)

Taatgen, N.A., Lebiere, C., Anderson, J.R.: Modeling Paradigms in ACT-R. In: Sun, R. (ed.) Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation. Cambridge University Press, Cambridge (2006)

VanLehn, K., Bhembe, D., Chi, M., Lynch, C., Schulze, K., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., Wintersgill, M.: Implicit versus explicit learning of strategies in a non-procedural cognitive skill. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) ITS 2004. LNCS, vol. 3220, pp. 521–530. Springer, Heidelberg (2004)

VanLehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., Win-tersgill, M.: The Andes Physics Tutoring System: Lessons Learned. Int. J. Artif. Intell. Educ. 15(3) (2005)

Wikipedia. Cognitive architecture, `http://en.wikipedia.org/wiki/Cognitive_architecture` (Accessed March 16, 2010)

Wenger, É.: Artificial intelligence and tutoring systems. In: Computational and Cognitive Approaches to the Communication of Knowledge. Morgan Kaufmann Publishers, San Francisco (1987)

Weiner, B.: An attributional theory of motivation and emotion. Springer, New York (1986)

Woolf, B.P., Murray, T., Marshall, D., Dragon, T., Kohler, K., Mattingly, M., Bruno, M., Murray, D., Sammons, J.: Critical Thinking Environments for Science Education. In: Proceedings of the 12th International Conference on AI and Education (2005)

Woolf, B.P.: Building intelligent interactive tutors. Student-centered strategies for revolutionizing e-learning. Elsevier, New York (2009)

Zinn, C., Morre, J.D., Core, M.G.: A 3-tier Planning Architecture for Managing Tutorial Dialogue. In: Cerri, S.A., Gouardéres, G., Paraguaçu, F. (eds.) ITS 2002. LNCS, vol. 2363, p. 574. Springer, Heidelberg (2002)