UNIVERSITÉ DU QUÉBEC À MONTRÉAL

# RÉALISATION D'UN AGENT DOTÉ D'UNE CONSCIENCE ARTIFICIELLE : APPLICATION À UN SYSTÈME TUTORIEL INTELLIGENT

THÈSE

PRÉSENTÉE COMME EXIGENCE PARTIELLE

DU DOCTORAT EN INFORMATIQUE COGNITIVE

PAR

DANIEL DUBOIS

AOÛT 2007

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

# CONSTRUCTING AN AGENT EQUIPPED WITH AN ARTIFICIAL CONSCIOUSNESS: APPLICATION TO AN INTELLIGENT TUTORING SYSTEM

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

BY

DANIEL DUBOIS

AUGUST 2007

***Je dédie cette recherche à mon épouse, aimante, compréhensive et exemplaire dans son organisation, qui a su accepter mon obsession à mener à terme ce projet de recherche.***

# ACKNOWLEDGMENTS AND SPECIAL THANKS

There are many persons to whom I owe this research's success.

The first person I thank is my wife, who took care of our family's organization and emotional balance. Years after our wedding, I still recognize in her a gift from God. My parents also deserve my gratitude; their love and support took many forms.

My research supervisors have been remarkable in supporting my project. They have encouraged me, knocked down my doubts, answered my never-ending questions, contributed to my papers and oriented my efforts. Roger Nkambou and Pierre Poirier, you are the bests!

I also owe thanks to the Admission Committee for the doctoral degree in Cognitive Computing at the University of Quebec in Montreal; they have been able to see past my borderline academic preparation with my Bachelor in Theology and an MBA, and recognized my eagerness to learn and explore the mind's wonders.

Professor Stan Franklin (University of Memphis) has shown an incredible support to my project, supplying me with IDA's code, assigning some of his team's time to answer questions, and always returning, lightning-fast, thorough emails to my inquiries. Professor Franklin made me feel part of his team.

The colleagues that created the first and the current iteration of our prototype also deserve much credit, both for their patience while I was exploring avenues of solutions with them, sometimes for hours, bringing more than a few grams of good sense into my wild dreams, and for the energy and genius they put in creating a very complex implementation against tremendous odds mostly related to the limited time. Their names are Patrick Hohmeyer, now holder of his Master Thesis, and Mohamed Gaha, the current guru of the code and a brilliant collaborator.

Other members of our lab also brought their contributions, commenting some of my writings or finding time to rush for adapting their own code so that it would

work with or within my new agent. Khaled Belghith, Philippe Fournier-Viger, Usef Faghihi and Vincent Gournay, thank you!

Financial support has much facilitated the two last years of my research. I want to thank people at the Canadian Space Agency (CSA) and the Canadian Government for granting research funds to our lab. Roger Nkambou and Froduald Kabanza (a fellow researcher at the Université de Sherbrooke) have my gratitude for assigning part of these funds to me. I add special thanks to CSA's Leo Hartman, who took time to examine and comment on my early prototypes, offered me a tour of CSA's installations, and to Edward Tabarah, who granted me complete access to a full week of astronauts training on Canadam2.

Finally, I really want to thank my three kids for doing their best at understanding why I showed little availability during all this time. I love them.

# TABLE OF CONTENTS

# LIST OF FIGURES

Figure Page

Figure Page

# LIST OF TABLES

# ABSTRACT

Intelligent tutoring systems (ITS) bear the great potential of supplementing, sometimes even replacing, the human tutors with unbound availability in time as well as in place. Better constructivist tutoring requires taking into account a greater number of contextual information sources. However, tutoring becomes increasingly harder when an ITS designer wishes to take into account many factors. Doing great tutoring calls for many integrated skills, and not only demonstrating an obvious mastery of the subject matter. The more various types of information an agent senses, the more apt it may be. But the variety and volume puts pressure over the processing of all that is sensed, along with all the information already possessed by the tutoring agent. Expert human tutors usually show this capability. However, recreating it in an artificial agent by combining successfully all the information pieces in a computer search algorithm can overwhelm even the most powerful computer; trying to achieve this in a rule-based system will discourage most rules creator.

Humans have evolved all sorts of tricks to tackle complexity. Baars (1988, 1997b), and Sloman and Chrisley (2003) entertain the idea that attention and consciousness are major mechanisms allowing humans to consider various sources of information, even concepts regarding past experiences, create abstract concepts, and not easily get bogged down or overflowed. In my research, I propose that consciousness can be an asset for artificial agents, even if not “complete” or “real”, by human standards. I mean to uncover the possibilities consciousness might bring, and explore whether (and how) it can be implemented. Many models of consciousness exist, and some agents already have been built on some conscious ideas. Of particular interest, Baars has laid down a theory, the *Global Workspace Theory* (Baars, 1988, 1997), which gives a nice account of consciousness phenomena and roles. I propose hereby a tutoring agent architecture based on Franklin’s IDA “conscious” agent architecture, with some modifications and extensions. Scenarios demonstrate the viability of the architecture for real-time interactions when coaching an astronaut during his learning of Canadarm2 manipulation. Whilst being founded on Baars’ theory, this architecture shows many commonalities with ACT-R and BDI theories. The resulting prototype has been validated against expert analysis, work-through analysis, and field observation.

The proposed research offers a new architecture for ITS that bears much potential, and opens up a number of further projects and researches in the fields of ITS and cognitive sciences.



***Keywords****: artificial consciousness, general intelligence, cognitive modeling, cognitive agent, Global Workspace theory, Baars, CTS, tutoring agent, IDA, LIDA.*

# RÉSUMÉ DE LA RECHERCHE

Les systèmes tutoriaux intelligents (STI) portent un remarquable potentiel pour soutenir, parfois remplacer, les tuteurs humains grâce à leur disponibilité sans borne quant au temps et au lieu. Tout comme l'humain, l'agent artificiel doit coordonner de nombreuses habiletés et savoir faire usage d'informations contextuelles excédant les seules connaissances liées au domaine. Toutefois, cela créé une pression importante sur le traitement perceptuel, et augmente les possibilités de combinaisons avec tout ce que l'agent possède déjà. La création d'agents augmente en complexité au fur et à mesure où les concepteurs cherchent à intégrer un plus grand nombre de facteurs dans les processus décisionnels. Toutes les capacités et tous les savoirs doivent opérer d'une manière intégrée. Les tuteurs humains experts y parviennent habituellement, mais la reproduction de ces règles dans un système à base de règles peut décourager la majorité des concepteurs de systèmes à base de règles, ou sinon excéder les capacités computationnelles du plus puissant des ordinateurs.

Les humains se sont dotés, au cours de l'évolution, de toutes sortes d'astuces permettant de gérer la complexité et dépasser les contraintes de l'immédiateté. Baars (1988), et Sloman et Chrisley (2003), soutiennent que l'attention et la conscience sont deux exemples éminents de mécanismes complémentaires permettant aux humains de considérer un grand nombre de sources d'information, incluant des concepts d'expériences passées, tout en demeurant hautement réactifs.

Ma recherche soutient l'hypothèse que la conscience peut enrichir significativement des agents artificiels, même si cette "conscience" n'atteint pas encore la complexité, la complétude, les modes et la réalité de la conscience humaine. La première étape consiste à déterminer les possibilités ouvertes par des mécanismes de conscience artificielle, et d'explorer s'il est techniquement envisageable d'y parvenir, et par quels moyens. Tout particulièrement, la théorie psychologique de Baars, l'*atelier global*, retient mon attention. Elle se fonde sur la modularité de l'esprit et sur les rôles partagés entre mécanismes conscients et inconscients. Je propose ici une architecture d'agent tutoriel fondée sur l'architecture de l'agent "conscient" IDA élaborée et enrichie par le professeur Franklin et son équipe depuis 1996. Le prototype de CTS (*Conscious Tutoring System*) que j'ai développé avec l'aide de plusieurs collaborateurs contient des modifications par rapport à son modèle source, ainsi que les extensions nécessaires au tutorat. Quoique fondé sur la théorie de Baars, on peut y découvrir de multiples parallèles aux théories BDI et ACT-R. CTS a été soumis à deux scénarios inspirés de la réalité. Il y démontre sa capacité à gérer la complexité en temps réel pour assurer l'encadrement d'un astronaute en entraînement sur le télémanipulateur Canadarm2.

***Mots clés :*** *conscience artificielle, intelligence générale, modélisation cognitive, agent cognitif, théorie de l'atelier global, Baars, CTS, agent tutoriel, IDA, LIDA.*

# Chapter 1

# INTRODUCTION

Intelligent tutoring systems (ITS) bear the great potential of supplementing, sometimes even replacing, human tutors with unbound availability in time as well as in place. Apart from occasional maintenance, failures and operating system instability, artificial tutors show a very stable personality, they do not require rest and will never balk at starting a lesson at midnight. In the same vein, under the assistance of an artificial "teacher" running on a computer connected to the Internet, a learner may happily take lessons from home, or get his training by a quiet river.

ITS are already helping learners in various ways. They help learn subject matter, acquire procedural knowledge, improve reasoning abilities through interactive simulations, and train manipulation skills. Some popular systems are Andes (VanLehn, Lynch, *et al*., 2005) in the field of physics, Autotutor (Graesser, 2005) in the field of Newtonian physics and computer literacy, Adele (Shaw, Johnson, *et al.*, 1999) in the field of medical diagnostic skill development, CIRCSIM-Tutor (Evens *et al.*, 2001) in the field of cardiovascular physiology, and SHERLOCK (Lesgold, Lajoie *et al.*, 1992; Sherlock2: Katz, Lesgold, *et al*., 1998) in the field of avionics troubleshooting skills. According to Graesser, Jackson, Mathews *et al.*, (2003), many ITS have been shown to facilitate learning, with learning gains going from 0.3 to 1.0 standard deviations units compared with students learning the same content in a classroom.

Tutoring becomes increasingly harder when an ITS designer wishes to take into account many factors: to the subject matter (level of difficulty, familiarity), considerations about learner's learning style and interaction preferences, appropriate

didactical strategies, pedagogical theories, learner's past history (successes, failures and patterns), learner's actual physical and affective state, and even cultural aspects. Even for human teachers, it remains difficult to determine the right amount of subject matter, the right time to intervene, and the proper way to offer guidance. Intervening too soon does not allow the learner the time to realize he is experiencing a difficulty (or it does not leave him enough time to forge an idea about the nature of the problem); too late, and he might get angry or discouraged. Offering too much of new matter makes him confused of even lost; too far away from what he already knows, demotivation may drive him away. Empathy may be perceived as childish to some, straight talk will be received as rude by others. Many researches are ongoing, trying to find proper ways to model the various aspects of tutoring, and to coordinate them within artificial agents. Some have met some measure of success, as the given examples testify. But they have all been dealing with a subset of all of the parameters that would be of interest, for instance, leaving aside the emotional aspect of tutoring (learner's feelings and emotions, and *tutor's* emotions).

There currently exists quite a few architectures that qualify as *cognitive*[1], that is, that model their internal processing of information upon the human mind's functions. However, to my knowledge, the only cognitive frameworks currently offered to the AIED (**A**rtificial **I**ntelligence in **ED**ucation) community, ones that are fundamentally thought from the ground up for taking many aspects into consideration, are the production rules-based SOAR, suggested by Newell (1990), and ACT-R, from Anderson (Anderson, 1993). SOAR, among other applications, drives STEVE, the animated

[1] The word *cognitive* is understood variably by different disciplines. AI generally sticks to the psychology's understanding of the word as modeling the methods by which human solve problems; it refers to an information processing view of an individual's psychological functions. Architectures may be "cognitive" at varying levels of validity, sometimes making no assertion about the plausibility of their parts, only integrating an eclectic group of AI techniques. Then, "cognitive" may encompass a surprising long list of agent's and architectures, including, along with SOAR and ACT-R, CS/SAS (Norman and Shallice, 1980), TETON (VanLehn and Ball, 1991), PRODIGY (Carbonell *et al.*, 1990), HOMER (Elinas, Hoey and Little, 2005), ICARUS (Langley *et al*., 1991; Langley and Choi, 2006), and others.

pedagogical agent developed at the CARTE (Center for Advanced Research in Technology for Education) of the University of Southern California. ACT-R is a well-known theory of mind in the psychology world, and it came to be applied to artificial tutoring agents. Other approaches to the mind of a tutor might bring new perspectives on how to tackle the challenges of tutoring students; they might bring new tools and new possibilities.

When one aims at taking on all aspects of a situation, and incorporate the "human touch" on top of it for the feedback (that is, grant the artificial agent personality and emotions), applying simplifying assumptions may not offer a viable avenue. The system becomes highly complex, hard to grow and maintain with a centrally-managed rule-based system.

In this research, I propose an original approach to manage the complexity of tutoring: human consciousness mechanisms as the core of a highly decentralized and modular architecture.

## 1.1 PROBLEM STATEMENT: HOW COULD AN ARTIFICIAL AGENT BECOME CONSCIOUS AND WHY WOULD IT WANT TO ANYWAY?

Interacting with humans in general, and with students in specific, requires an awful lot of subtlety if one is to be perceived as a great tutor and a pleasant fellow. That is, doing great tutoring calls for many integrated skills, and not only demonstrating an obvious mastery of the subject matter. What seems to produce results is about being able to track knowledge and misconceptions of the student and adaptively respond to these deficits at a fine-grained level. This happens by scaffolding upon learner's previous knowledge acquisition, help him co-construct his knowledge by answering his questions (Graesser, Person, *et al*. 2005), and teach him to act at a metacognitive level (du Boulay and Luckin, 2001). According to Piaget's *constructivism* (1970), since learning is strongly related to the learner, to what makes him unique, being sensitive to the various dimensions that shape his specificity takes on much importance. It encompasses more than just learner's actual knowledge, it in-

volves taking into account "soft" aspects such as preferences and fears, personality, actual mood and emotions, actual physical state (tired, sick, excited, etc.) and so on. On that basis, the more various types of information an agent senses, the more apt it may be at adapting its behavior and interact specifically. This is what most of us in the ITS field would like to see: artificial tutoring agents reaching (or exceeding!) human-level tutoring.

Then comes the burden of processing all that is sensed, along with all the information already possessed by the tutoring agent and relevant to the situation, adding in all the factors proper to the tutor (personality, professional goals, agenda, etc.). Expert human tutors do it all the time (although not always skillfully…), but it is not as easy as it may seem, generally requiring many years of study and hands-on practice. Combining all the aspects in a computer search algorithm can overwhelm even the most powerful computer, and will discourage any rule creator. Humans have evolved all sorts of tricks to tackle that complexity, cheating as often as possible, selecting parts and aspects, simplifying and "chunking" all they can, and processing the remainder. The visual apparatus contains many remarkable examples of such clever and efficient mechanisms, requiring a detailed account of what is visually perceptible for only about 6 degrees of the field of view. Attention and consciousness are other examples of those tricks. Baars (1988, 1997b), and Sloman and Chrisley (2003) entertain the idea that they are major mechanisms allowing humans to consider various sources of information and not easily get bogged down. More specifically, they are what makes them able to take into account the many aspects that everyday situations involve, and lets them adapt efficiently to unforeseen situations, sometimes in subtle ways. Consciousness does its magic here by first making possible to abstract reality, create concepts that can be manipulated in reasoning. Those allow considering alternatives, especially when first results do not meet expectations. Consciousness and attention evolved to permit intentionality, volition, existence of a self that guides adaptation.

"Adapting" to a learner is complex and cannot all be prepared in advance. It often means creating new plans or modifying the existing ones. A tutor has to make

minor modifications to the general plan he had made about a lesson because the learner does not possess the knowledge he thought he had, or completely change the plan because learner's reactions are indicative of physical fatigue or mental indisposition. The tutor has to be attentive to certain aspects present in his perception and not to others, filtering out what is "noise" (with respect to what he has decided as being his immediate goal, for instance assessing his pupil's disposition for the lesson). He has to set a new goal, be it of starting the lesson, selecting a proper way to do so, modifying the lessons plan, or of rather going after a way to stimulate the student. All these mental activities require the ability to manipulate concepts, concrete as well as abstract ones, such as "attitude", "mental fatigue", "goals", "steps" and "priority". They involve getting access to resources upon which one has little direct control: recalling memories about previous sessions and facts about the type of learner he is tutoring, giving interpretation to fuzzy impressions about the learner's mood or the general situation, relating these to goals of various natures, associating relevant words and facts together and organizing them towards a modified plan and an appropriate reaction, sometimes at an affective level. That sort of adaptation makes use of voluntary actions, of making choices with respect to various criteria. It cannot be reached by applying a single standard pattern of organization, but rather involves the manipulation of abstract notions gathered through experiences and learning from them

Sloman (1999) has hypothesized that it is through environmental pressures that humans have evolved the capacity of taking a distance with respect to the immediate reality, conceptualizing the physical world and becoming able to manipulate a non existent world, exploring and analyzing new configurations and alternatives. Consciousness is the means to those mental manipulations. Its most prominent manifestation is something we do all the time: talking to ourselves, forming sentences that we pronounce "in our heads" to do analyses, translate impressions into words, giving life to those words and images that popped from nowhere when we spoke these words about our impressions, and so on. From there follows the unavoidable, albeit unusual, question in the ITS (AI) field: *what is consciousness*?

## 1.2 FIRST IDEAS ABOUT CONSCIOUSNESS

The deceptively simple question about the nature of consciousness throws us in muddy waters. It may take quite a few more years before we can give the right answer, even though it has come again to be of central interest, with new research tools, only in the recent years. As Baars puts it:

> «*You are conscious, and so am I.* This much we can tell pretty easily, since when we are not conscious, our bodies wilt, our eyes roll up in their orbits, our brain waves become large, slow, and regular, and we cannot read a sentence like this one» (Baars, 1997, p.3).

And that's about all most of us can say about consciousness. But, in fact, even the "You are conscious" part of Baars statement can be doubted: «You say you are conscious? Prove it!»… The debate has been joined very recently by researchers from all fields: psychologists, neuroscientists, physicists, mathematicians, and AI researchers. The growing number of interested qualified researchers has made of a fascinating subject a central concern. With the help of new technological means, we are getting new insights by the month.

I will avoid as much as possible exposing myself to the *black hole's attraction*, as Taylor puts it (Taylor, 2000), of debating consciousness' nature. That is a research field on its own. The width and depth it encompasses illustrates that point pretty well, as demonstrated for instance by David Chalmer's website[2]. It is not a necessary concern for the goal I set for my research. I only mean to identify consciousness' roles, explore possibilities it brings to artificial agents, and determine whether and how it could be implemented. Looking at consciousness from the neuroscientific and psychological points of views, Baars has laid down a theory, the *Global Workspace Theory* (Baars 1988, 1997), that gives a nice account of those roles, and how consciousness serves the purpose of letting humans adapt to their

[2] http://consc.net/biblio/6.html

complex environment. This theory is a foundation of my project, and we will go through its major propositions in Chapter 2.

If the subject has now become an overheated boiler, the train it now hauls took some time to gain its momentum. For instance, the AI and computer hardware communities have not initially been paying much attention to it, at least not at a conscious level (!), busy as they were trying to figure out how they could have a robot perceive and reason about simplified worlds, looking for efficient algorithms that would eventually surpass human performances. On the hardware side, you may be surprised by the idea that computers pretty soon incorporated mechanisms that were inspired by, and reflect, mind's processing, even consciousness' selectivity and seriality. For example, various aspects of the inputs (mouse movements and clicks, keypresses, network packets, and so on) are processed by a collection of specialized *processors* that bring only their conclusions or problems to the "general purpose" central processing unit. This corresponds well to unconscious and automatized processing in humans. Explicit and purposeful consideration of the mind's architecture, and of consciousness, happened only when it became apparent that sheer computer power would not allow a machine to equal human's performances. John McCarthy and Marvin Minsky brought AI's attention to the field. They were precursors with ideas about giving a robot the capacities to do self-observation (McCarthy 1959; Minsky, 1961). Their idea was that robots would need human intelligence if they were to cope well with the task we would like to give them, and that included consciousness in their view. Few will object to recognizing the existence and role of consciousness in human intelligence, but they really have been visionaries in the AI world. Taylor offered a remarkable insight to consciousness with his Relational Mind Model in 1973, but most contributions came later, in the early 80's, for instance with Johnson-Laird’s computational analysis of consciousness (1983), Baars’ Global Workspace Theory of mind and consciousness (1988), and Edelman's Biological Theory of Consciousness (1989). Although McCarthy offered ideas for a *reflexive* computer language, implementations of anything referring to consciousness *for an agent* appeared only in the 90's, with examples in Hexmoor, Lammens and

Shapiro's GLAIR (1993), Cazenave's Introspect (1998), and Franklin's Conscious Mattie (Ramamurthy, Bogner and Franklin, 1998).

If we applied ideas about consciousness to tutoring agents, what are the specific benefits we can expect? What goals would we be pursuing? These are all fundamental questions that I will address when I present the architecture of our "conscious" tutoring agent, called CTS (Conscious Tutoring System), and its instantiation in Canadarm Tutor. I give here an implicit answer by offering a glimpse at CTS' architecture. Our[3] cognitive agent complements Roman Tutor, a non-cognitive tutor integrated in the International Space Station simulator our lab has developed. The original tutor was meant to monitor progress and coach astronauts learning how to manipulate the Canadian robotic Arm, Canadarm2. CTS implements a cognitive architecture based on Baars' Global Workspace (GW) theory, which describes how consciousness allows the various parts of the brain to collaborate when each individual process is not enough to cope with a situation. Franklin and his team have realized a *functional* computer adaptation of that theory into Conscious Mattie, IDA and LIDA. "Functional" means that the *functions* of the brain are reproduced by whatever means is convenient. Biological plausibility is not sought for at that level, although, in the case of these agents, the functional plausibility is maintained to some level. Our CTS agent has its roots in IDA, LIDA's predecessor (LIDA stands for *Learning* IDA). Specialized modules reproducing high-level brain functions (perception, working memory, long-term memory, knowledge about the user and about the domain, action selection mechanisms), are loosely interconnected through mechanisms that implement working memory, attentional mechanism, and "a*ccess consciousness"* (Ned Block's term for one of the many "types" we may identify under the single word of "consciousness"). Without these mechanisms, modules are limited in the collaboration they can conduct to accomplish agent's adaptation; they can

[3] When talking about CTS, I purposely use "we" most of the times, as the agent is the result of a team effort, not just mine.

communicate only within "unconscious", preprogrammed routines. Another fundamental idea about these agents is that they pervasively use Baars' idea of the mind's elementary and autonomous processes as a foundation of much of the processing; thereof, consciousness is required for sophisticated adaptation. These are specialized processes (or process***ors***, representing neuronal groups, implemented as *codelets* in Franklin's agents and in our own CTS) that can accomplish a simple task very fast, but are devoid of the capability to adapt. Just as unconscious processes accomplish very fast processing and require little of mind's energy resources, codelets are very efficient, compared to "heavy", "conscious", iterative collaborative processes. They allow a fast processing of standard information and familiar patterns, allowing an agent to react fast in many common situations. They make possible for an agent to do more than one thing at a time, do parallel processing, eventually on the same information. More than that, they allow a tutoring agent to consider a situation from multiple points of view. Just as do all the modules of the agent, they work independently of each other, but are however listening to the Access Consciousness' "publications" (or "broadcastings"), reacting to what they recognize, lending a helping hand when they can. They bring their information, the result of their manipulations, into Working Memory, where all information codelets either cooperate as coalitions or compete to come to Consciousness. "Coming to consciousness" is the result of being selected by the Attention mechanism to be broadcast throughout the agent. An example of that would be the processing of the stimuli we call "a written sentence", each portion of it being processed at a physical level by a multitude of simple processes specialized in recognizing lines, circles, and so on, to make out letters, with other processes taking their resulting output, letters, and organizing them into words (with the help of the perceptual memory), then words into semantic structures (with the collaboration of the semantic memory). These automatic unconscious operations allow a tutor to interact verbally in real time. Making all that processing through voluntary (conscious) operations would take minutes instead of fractions of seconds, and a lot of mental energy. Consciousness is needed to tackle new information, or unexpected situations, for which no automatic routine exists, but such examination takes time and is heavy on resources. It came to exist through evolutionary pres-

sures for stronger adaptation means. A tutor needs both: fast unconscious (but limited) reactive capabilities, and powerful (but slow) conscious analysis.

How does CTS' cognitive architecture compare? There are various aspects under which CTS may be studied. In Chapter 6, we will have a look at CTS' architecture with respect to a popular agent architecture, BDI, and to a theory of mind computationally implemented: ACT-R.

## 1.3 OBJECTIVES OF THE RESEARCH

The proposed research aims chiefly at extending Franklin's agent IDA to create an artificial tutoring agent endowed with many mechanisms proper to human consciousness. At the same time, I set the constraint of respecting Baars' Global Workspace theory as much as possible for the core of the architecture, and finding inspiration from cognitive sciences for aspects that get segregated inside "peripheral modules". Tutoring offers to this research a field of application, with real life situations demonstrating how consciousness may allow better interactions, flexible adaptation to the learner.

A secondary objective is about offering a new architecture for intelligent tutoring systems, one that will be considered because of its richness, its extensibility, and its potential for reproducing humanly behaviors.

A ternary objective, almost a side effect arising from the necessities of this research, proposes to build some tools that will be the foundation for a complete framework of development for future cognitive agents. A Behavior Network editor is a major step in this direction.

## 1.4 RESEARCH METHODOLOGY

The following steps give a summary of my research:

- Clarification of the concept of consciousness and related concepts
- Hunting for conscious models, architectures and agents, in order to find examples of how consciousness could be brought into agents
- Selection of a starting point (a theory, a model, an existing architectures, etc.)
- Adaptation of the architecture to the domain
- Iterative implementation of the architecture, looping as often as needed to bring improvements when cognitive aspects are better understood, and to try and solve theoretical and implementation difficulties
- Elaboration of life-like scenarios to test the agent
- Evaluation of the results

I found early on that complexity is the beast to tame towards fruitful tutoring. Complexity also exists in every aspect of the research, with many competing concepts, points of views, and propositions for solutions. So, I found that sticking to a global theory that seems pretty well supported was a prudent line of conduct for such an ambitious project ("ambitious" in the sense that it encompasses a great number of fields, each with its own richness and peculiarities, and that tries to bring them to work together). Getting a clear view of what is generally understood as consciousness seemed a necessary first step. That in itself is a major undertaking, and a good example of each field being a complex world in its own right. So, I will propose an integrated overview of the ideas surrounding consciousness (awareness, intelligence, metacognition and reflection), but will steer clear of the sophisticated philosophical discussions. Then, I review the literature to find whether consciousness has already been modeled, maybe implemented or even demonstrated in actual agents, and if so, how, with what benefits, under what limitations. That brings me to selecting the most

promising approach towards my goal of effectively constructing a conscious tutoring agent. I have elected IDA as the best foundation for my project and established contacts with Professor Franklin, who allowed our lab to use as much of their code as was relevant. From this starting point, building an architecture entails examining the code and see how we can add missing features for a tutoring system. We used the code as a source of inspiration and rebuilt the agent from scratch. As our project has a specific application with tutoring astronauts, I was immensely grateful that the Canadian Space Agency would permit me to go observing the astronaut's training. It allowed me to see the activities first hand, noting the human tutor's techniques, attitudes, behaviors and reactions. Then, I was better positioned to adapt the architecture and construct the features I wished to add. Evaluating how well my colleagues and I have implemented the theory and how much the resulting agent performs as expected are the final steps of this research

The structure of the document is the following:

The next chapter, Chapter 2, presents the concept of consciousness and offers Baars' point of view, that is, his Global Workspace theory. Having an understanding of consciousness and of Baars' theory equips us with some perspective before examining consciousness-related works in the next chapter.

Chapter 3 reviews the literature on the field of conscious agents and consciousness models. I present a sampling of what is available: a computer science approach with McCarthy's reflexive language; functional approaches to consciousness with Hexmoor, Lammens and Shapiro's GLAIR, and Cazenave's Introspect; and biologically-motivated approaches with Cotterill's Cyberchild and Grossberg's ART.

Chapter 4 presents CTS, our "conscious" tutoring agent's architecture. I highlight some interesting features of the global architecture, then say a few words on the implementation and on the Behavior Network editor.

Chapter 5 describes the tutoring context and activities expected when training astronauts to the manipulation of Canadarm2. I explain services offered by the tutoring agent, and give insights on its internal operations through two example scenarios.

Chapter 6 discusses how CTS compares to a popular agent architecture, BDI, and to a just as popular cognitive architecture, ACT-R.

Chapter 7 presents the validation methods adopted and evaluates the prototype.

Finally, Chapter 8 discusses the next steps for a continuation of this research.

# Chapter 2

# CONSCIOUSNESS AND THE GLOBAL WORKSPACE THEORY

Before restricting our attention to one hypothesis about consciousness, I would like in this chapter to lay some ideas about fundamental notions I am going to use for building CTS. First, I offer an overview of the concept of consciousness, what this mysterious word covers, and whether a machine can ever be said to possess it. Then, for a deeper understanding, I refine the concept some more by distinguishing consciousness from very close ones: awareness, reflection, metacognition and intelligence. This will give us a much better perspective when we examine Baars' theory of mind and consciousness in the third and last section of the chapter.

## 2.1 WHAT THE CONCEPT OF CONSCIOUSNESS REFERS TO

Much can be said about consciousness, and little can be said. These words may summarize the traditional debate about consciousness. Trying to seize consciousness, trying to understand what is asserted about it, one may easily be forced to dig deeper and deeper in subtle discriminations, and may find himself attempting to grasp the ever wider horizon of the immense variety of diverging opinions. I certainly got confused at some point and had to backtrack quite a few times and even hire a guide. For the intent and purposes of my actual research, I will offer here a rather tangential approach on the subject, that is, I will do a touch-and-go on the debate, trying to avoid running deep into a region full of quicksand. My interest in this thesis is not to debate the existence of consciousness, to discern its nature, or even to establish the "truthfulness" (or reality) of the consciousness I create in a software

agent. I take consciousness existence as a given, then I only intend to show its usefulness for an artificial agent, and present a way to recreate its mechanisms.

Recent works in cognitive sciences keep adding new insights, offering a richer, more detailed view of consciousness and of how it may function. But more is not always better. Much of the debate may be due to inconsistencies in the language and misunderstandings. There are so many ways to look at the subject, beginning with the popular understanding of "being conscious" («She came back to consciousness a while after hearing the great news.»). There is the scientific approach, advocated by Chalmers as beyond our current scientific means (we have no measuring counter for consciousness, it does not register on our instruments). There is the *new mysterians* approach of McGinn that proposes that consciousness simply is out of the reach of our minds, so there is no way, scientific or otherwise humanly possible, to investigate it. The eliminativist standpoint claims that consciousness as an autonomous entity does not exist, it is just a by-product of biological processes; so, Chalmer's *hard problem of consciousness* (explaining the experiences we live and feel; Chalmers, 1995) is an illusion that philosophers love to lean on (Dennett's position; Dennett, 1991). Other scientific personalities do not care about consciousness nature and focus on trying to find its *neural correlates*, that is, to identify the neurobiological processes that support it; for instance, two scientific teams, Crick with Koch, and Dehaene with Naccache are about to uncover them, with pretty convincing evidence. The theological point of view suggests that consciousness is what gives Man his superiority over Nature; some see it as what allows him to talk with God, joining the metaphysical stance (the universe is based on a non-physical independent reality: consciousness; it is akin to soul), and the esoteric interest (the goal of meditating is to reach Pure Consciousness and rejoin our common essence).

Each stream shows a variety of hypotheses, and some authors propose hybridization or an eclectic assembly of proposals. With reasons, in 1995 Ned Block claimed that "consciousness" is a mongrel concept, and that we won't be able to hold an appropriate discussion on its nature if we do not recognize that the word encompasses many phenomena and mechanisms. Minsky asserted the same idea in 1998

when he said that “consciousness” is a suitcase-word, like intuition, learning, memory, a word that all of us use to encapsulate our jumbled ideas about our minds. Block (1995) attempted to make the debate more focused by declaring four "types" of consciousness: the *access* consciousness (the phenomenon that temporarily connects an unconscious resource to other unconscious resources in our brain so that they can interact), the *phenomenal* consciousness (that holds the properties of the experience, the ineffable qualities of the phenomenon), the *monitoring* consciousness (the processes that monitor our senses and our internal states and make them known), and the *self*-consciousness (our knowing of being an individual with his own, separate existence).

Chalmers (1995) offered a similar account, and also isolated the *easy* problems of consciousness from what he coined the *hard problem of consciousness*. The easy problems of consciousness are those that are directly susceptible to the standard methods of cognitive science, whereby a phenomenon is explained in terms of computational or neural mechanisms. The hard problems are those that resist those methods.

The easy problems of consciousness include the following phenomena:

- the ability to discriminate, categorize, and react to environmental stimuli;
- the integration of information by a cognitive system;
- the reportability of mental states;
- the ability of a system to access its own internal states;
- the focus of attention;
- the deliberate control of behavior;
- the difference between wakefulness and sleep.

As Chalmers describes it, «All of [these phenomena] are straightforwardly vulnerable to explanation in terms of computational or neural mechanisms». Although one should not exaggerate the "easiness" of theses questions, one must recognize that computer science has long been using these concepts (an idea sustained in

Bechtel, 1995), albeit maybe sometimes at a sub-conscious level (!). Johnson-Laird, for one, has made a very voluntary, very conscious effort at discovering consciousness' roles, with the explicit goal of later implementing them in computers (see for instance Johnson-Laird, 1988). The simple fact of making an explicit use of a piece of information and being able to report on it, is made possible by its becoming conscious; this is how it becomes available to other processes, among them language analysis and generation. Such accounts from a psychologist (Johnson-Laird) and a philosopher (see below description from Chalmers) could easily be thought of coming from an AI proponent.

> Sometimes a system is said to be conscious of some information when it has the ability to react on the basis of that information, or, more strongly, when it attends to that information, or when it can integrate that information and exploit it in the sophisticated control of behavior (Chalmers, 1995).

One can recognize access consciousness and monitoring consciousness in the way the central processor (CPU) and collections of sub-processors collaborate. The CPU acknowledges inputs from keyboard and mouse and processes them along with requests from everywhere in the system, but does so one at a time[4], serially, in the same fashion that attention selects one information at a time. The CPU sends requests to sub-processors, which contain compiled processes, very efficient at dealing with information within known boundaries; they have commonalities with our unconscious processes. Many requests and tasks are processed in parallel by theses sub-processors (unconscious processes). They either return the result of their "silent" (*unconscious*) work, or raise problems they encounter and cannot settle, to the attention of the CPU (for a "conscious, slow but adaptive, reparation process).

---

[4] This parallel is getting more and more imperfect as CPUs became capable of processing more than one operation in a single cycle by taking in-board co-processors, even another CPU (for instance, Intel's *Core2 Duo* processors).

Exposing the kinship of these ideas with computers is drawing us on a slippery slope towards a difficult debate about machine consciousness and zombies. But the ascription of "consciousness" onto my agent is in jeopardy, even turning illegal if not settled here! I must confront objections now.

Searle has taken a strong position against the possibility of *real* consciousness in a machine. He posits that simulating a process *is not* that process. Creating a simulation of digestion does not make the software digest; simulating comprehension does not produce comprehension. A famous image Searle offered to illustrate his opposition to the fact that computer do, or even just *can* think, is the *Chinese room experiment* (Searle, 1980). It tells of a person sitting in a room one could call a *processing chamber*. Only two openings on opposite sides allow documents to flow in and out. The person, who knows nothing of the Chinese language, has to read Chinese symbols that come in, consult a lexicon and a set of rules written in his native language, and write on another paper the appropriate symbols. If the rules are properly written, then, to an external observer, this closed, opaque room, a *black-box,* manifests an understanding of the sentences given to it since it is able to process them and respond as appropriately as a native Chinese speaker. However, in this example, no understanding is necessary, as it suffices for the internal process to connect words together through rules. The simulation of comprehension fools an external observer into thinking there is real understanding. Similarly, simulating consciousness does not produce real consciousness. Searle is positive that consciousness is entirely caused by neurobiological processes and is realized in brain structures. Now, since consciousness is *caused* by biological processes, combining artificial processes that simulate them will never produce consciousness in a machine that does not have the same biological substrate.

One could reply that, whether artificial or biological, information processing has a causal effect; of either nature, it causes appropriate results by a chain of effects, and they both have real impacts. That is the position that Harnad (2003) defends, saying that if an engineered being is able to fool us about its true nature (that is, its being artificial) during its whole life, it deserves to be considered as possessing con-

sciousness, not just existing as a philosopher's zombie. In agreement with Harnad and Minsky, Kurzweil (1999) posits that there exists no test or criterion absolutely trustworthy that can establish the presence of consciousness in an entity (human or otherwise). Neuroscientific imagery only shows correlations of brain activity with verbal reports from the human subject; it does not *prove* consciousness *per se*. There are only the behaviors, the introspection reporting by the subjects, and *contrastive phenomenology*[5] that offer tangible facts which can be analyzed with scientific means (Baars, 1997b).

I choose to take AI's stance in thinking that, real consciousness or not, a simulation can produce in an artificial being effects similar to those in a human. On that basis, computational mechanisms can give an agent most of the same advantages that we can see in natural beings, especially human. But let's be careful here. I emphasize that I use the words "effects similar to", and not "phenomenon of the same nature as". Similarly, the fact that I will be using "consciousness" and "conscious" without the quotation marks throughout the document when talking about my agent should never be interpreted as an affirmation of "true" consciousness. Dropping the quotation marks is only for easier reading. I am in no way stating that I posit CTS' consciousness mechanisms as producing "real" or "true" consciousness. In fact, I doubt that the recreation of a process at a mere functional level can produce the same phenomenon in its essence. I nevertheless believe the mechanisms can account for many of the "easy" problems Chalmers talks about, and that this can help us think about the phenomenon and go further.

To understand what features of consciousness I believe our functional-level mechanisms are *not* reproducing, here is a description of one of the four types Block

---

[5] According to Baars (1997b, p.12), phenomenology is the study of consciousness based on subjective reports; in scientific practice, we always supplement subjective reports with objectively verifiable methods. *Contrastive phenomenology* compares results of operations where people can report accurately, to ones that can be inferred and studied indirectly. Examples are normal *versus* subliminal perception, attended *versus* nonattended speech, explicit *versus* implicit memory, *etc*.

suggested: phenomenal consciousness. Phenomenal consciousness refers to the feeling we experience about a state, the qualitative aspect of that experience, one that we cannot easily communicate to other people because it comes from a personal, internal reality. We have no way to really compare it to other people's. Examples are the felt quality of red (the *redness* of that physical stimulus), the experience of dark and light, the feeling created by the sound of a big bell, the bodily sensation of pain, the internal reactions we call emotions, the experience of a stream of conscious thoughts – some of the examples a gave come from Chalmers. They can all be referred to as "what it is like to be in that state" (Nagel, 1974). What does underlie these felt experiences? Is it a matter of mechanism, structure, complexity of organization, or of substrate? We still do not know, or there is no strong consensus on this point. I, as a conservative researcher (and a prudent doctoral candidate) would not posit CTS as having this kind (or level) of experience; I would offer the idea (not *hypothesis*) that this will not happen at least until CTS implementation reaches a richness capable of sustaining general intelligence and true *grounding* (so that its *experiences* stop being tied to single-word descriptions. But, again, I do not want to get involved in the debate here, and lean on that aspect for the realization of the initial prototype of CTS[6]. As a first step, I adopt the "engineering" stance of using what seems like promising means for attaining the goal of a well-performing, adaptive agent. Access consciousness seems sufficient to this end[7].

---

[6] So, I will not address questions such as "*What role does phenomenal consciousness play in adaptation?", "How does it influence reasoning?", "Is the phenomenal consciousness dependant of the "more functional" access consciousness?", "What is their relation?" "Can they be separated?"* These are all fascinating issues, and I foresee that they will have to be taken under consideration at some point in the future evolution of CTS. Indeed, some hard-to-describe-in-words states, the phenomenal content of an experience, certainly can play a role as motivator to take action. Pain certainly can. It can even become part of reasoning when its content comes to be abstracted into propositional knowledge. Phenomenal consciousness must eventually be considered in CTS, having, at least potentially, a causal role.

[7] Block (2002) tries to clarify further the differences between phenomenal consciousness and access consciousness. One of the elements he suggests is that only *representational* content (as opposed to phenomenal) can play a role in reasoning. Whereas the status

As we will see in the next chapter, there are many ways to recreate consciousness. But first, and before I take you on a tour that will help better understand what consciousness might be, I feel it necessary to clarify a few concepts that are often used interchangeably with consciousness: awareness, reflection, metacognition and intelligence. Then, I will go on describing in some detail one specific model of consciousness, one that will become the foundation of our agent: Baars' Global Workspace theory.

## 2.2 A FEW WORDS ON RELATED CONCEPTS: AWARENESS, REFLECTION, METACOGNITION AND INTELLIGENCE

Talking about consciousness without having a clear understanding of its distinction with close concepts makes it difficult to stay on track. It even poses problems to philosophers. Here is a quick overview of these near cousins (sometimes twins) of consciousness.

### 2.2.1 Awareness

Awareness is the term closest to consciousness. In fact, Chalmers (1995) recommended that we use "awareness" to refer to the "easy" phenomena of consciousness, and that we reserve "consciousness" to phenomena that refer to the experience (the aspect quite well described by Nagel's (1974) famous circumlocution *What it is like to be a bat*). It is not to be confused with "sentience", the *ability* to have

---

of phenomenal consciousness content is less certain, access consciousness content is essentially representational. He adds that "what makes a state A-conscious is what a representation of its content does in a system". Therefore, I infer that access consciousness, a functional notion, offers a natural platform for causality and may be minimally sufficient alone in this role for an artificial agent.

sensations[8], a concept very close to phenomenal consciousness, to which it is a precondition. Awareness depends on sentience to exist. It is also sometimes confused with *sapience*, which adds a level of knowledge to the stimulus (from the perceptual processing). We usually try to restrict "awareness" to refer to what sentience directly permits, that is, to have a sensed stimulus create a reaction in our internal system. But I would gladly see Chalmers proposition be widely adopted, as I constantly find myself struggling with the difficulty of keeping "consciousness" and "awareness" in their designated realms!

### 2.2.2 Reflection and metacognition

These two concepts are intimately related. They may be used interchangeably, depending on what one puts under "reflection". Flavel (1979) describes metacognition as the cognitive faculty that allows the subject to think about how he thinks. As its name indicates, metacognition is a cognitive level on top of *another cognitive level*, observing it, taking action to regulate it (Brown, 1987). Note that it is not to be confused with *monitoring consciousness*, which is a cognitive process that observes the senses (a *non-cognitive* faculty).

---

[8] As is still the case with most words surrounding consciousness, "sentience" nature and description are debated and may be understood as the mere ability to sense. However, it can be nearly confused with phenomenal consciousness. According to David Cole (found in David Chalmer's compilation at http://consc.net/online1.html#perception),

> sentience, having a sensation or a feeling, or "qualia", is a phenomenon which goes beyond mere sensing, for it involves an internal state in which information (typically) about the environment is treated by the system so that it comes to have a subjective character. We know what this is like from our own case. Each normal person has had sensations of cold, bright light, sound, and pain. It is from such occurrences that we understand the reference of "having a sensation". Once we distinguish sensing from sentience, we may note that sensing is neither a necessary nor a sufficient condition for sentience.

Reflection (or *"self-reflection"*, a term more clearly differentiated from *deliberation*) may refer only to the voluntary activity or process that turns the subject's attention towards itself, as if he was two persons at the same time, one making observations about the other one. A reflexively conscious state is one that is phenomenally presented in a thought about that state (Block, 2003). When this reflection, this "discussion", turns into an analysis, it becomes metacognition, especially if it primes mechanisms that will work at regulating further actions and thinking. But then, there may be reflections of the person about his metacognitive abilities to improve them (Gama, 2000)… So, we see that these two concepts are not the same, but may sometimes do the same thing.

The deflection of the thinking process towards oneself (reflection) is not consciousness in itself but uses it. If the report gets accompanied by thought to the effect that one is in that state, then we talk about *metacognition*, according to Block (2002). One may then decide to enter a deliberation for further analyzing the facts, finding corrective measures and applying them (the *control* aspect of metacognition). This level of interaction requires consciousness.

### 2.2.3 Intelligence

As for most concepts, the exact description of intelligence is debated and imprecise. Here are two that I like for their simplicity and globality:

- Yam (1998): An exact definition of intelligence is probably impossible, but the data at hand suggest at least one: an ability to handle complexity and solve problems in some useful context.
- Peter Voss (2004): an entity's ability to achieve goals. Greater intelligence allows coping with more complex and novel situations. On three axes (complexity, adaptability and flexibility), intelligence exists on a continuum.

Voss puts consciousness as the highest level of intelligence. This corresponds to the iceberg hypothesis in which consciousness is the controlled part of the infor-

mation processing. In the opinion of Edelman (1989, 1992), consciousness emerges from intelligent processing (essentially taking place as re-entrant signaling between neural maps, confronting self to non-self, or memories to perception; see note 16 differentiating reentrant signaling and CTS/IDA's looping through its cognitive cycle). Block is just a little clearer about their separation, saying that consciousness allows intelligence to contemplate and regulate its effects.

So, we may conclude that they are separate but strongly connected realities. Consciousness makes possible the highest form of intelligence, and reciprocally, intelligence is the substrate from which consciousness emerges.

Now, after separating apples from oranges, and oranges from mandarins and tangerines, we are better equipped to dive into Baars' theory about consciousness and appreciate how well it encompasses consciousness and its related phenomena.

## 2.3 BAARS' GLOBAL WORKSPACE THEORY

### 2.3.1 Background

There are many hypotheses about what consciousness is, and there are many others that propose how it may work. I have discussed the former ones in the two previous sections; the latter ones are of concern in this section. I will only mention some that correspond to the basic ideas of Baars' theory.

Baars' theory is a global one that has taken many separate ideas and organized them in a coherent whole. His proposal is gathering a growing consensus and is receiving new confirmations every year from neuroscientific empirical research (see especially Baars, 2002). Interestingly, the ideas it contains are descriptions that bear themselves quite well to computer implementations. This, and the globality of the

theory, may explain why it was chosen by Professor Franklin as the basis for his agents (Conscious Mattie, IDA and LIDA).

In a 2001 paper, Engel and Singer gave an overview of the synchrony hypothesis, exposing that many researchers came to similar ideas. For instance, Crick and Koch (1990) proposed that only appropriately bound neuronal activity can trigger short-term memory and, thus, become available for access to phenomenal consciousness. Damasio (1990) presented a similar idea, stating that conscious recall of sensory contents requires the binding of distributed information stored in spatially separate cortical areas; the binding happens through synchronization of the firing rates of local and distant neurons, which eventually makes the content globally available. Edelman (1989; 1992) and Tononi and Edelman (1998) also suggested a similar binding process by *reentrant loops* between systems performing perceptual categorization and brain structures related to working memory and action planning. They also explain self-consciousness by the distance this process maintains between feeds from the perception and feeds from memories. Grossberg, in ART, has offered his explanation of conscious states as resulting from a *resonance* (or match) between top-down priming and bottom-up processing of incoming information, which also allows learning of information into coherent internal representations (Grossberg, 1999). These ideas about synchrony try to explain how various aspects, analyzed by separate brain structures, can come together under a common "concept" or a unified sensation. Various neurons from different cell assemblies fire their action potentials in temporal synchrony, putting together the various bits of information about an object or event to form a coalition making up the perception (or the complete idea, when the coalition is formed by internally generated information).

This binding of sources of information is also present in Baar's Global Workspace theory, albeit in a higher-level view of the process. We will now examine his description.

### 2.3.2 A theater metaphor for the Global Workspace

The Global Workspace theory can be summarized in a theater metaphor as follows (which I adapted a little from Baars (1997b, p.41)). The mind can be modeled after a theater, where we find a stage, a large audience (and I mean LARGE!) of specialized actors, and a backstage setting. The play has no script and relies on the talent of actors found in the audience to intervene when they feel they can contribute to the story they are watching. Actors are members of specialized theatrical companies. They may come to the stage alone, but generally have a complex message that needs the presence of more than one actor to present it (often coming from different companies). On stage, there is always only a small number of actors, with only a few of them having the spotlight shining on them. Those in the spotlight are somehow related and synergistically support each other; their global excitement demonstrates that they have the most important message to tell to the audience at the current point in the play. Backstage, there is a small number of staff that hear what is said on stage, prepare material that the actors request, and change the backdrops that set the meaning of what is spoken to the audience. There is also the director, never to be seen but often having a major influence on the next part of the play.

Figure 1 depicts mind's functions (appearing in bold in my description) corresponding to the entities of this metaphor. **Working memory** is like a theater stage (Baars, 1997b, p.41). It is the "structure" which contains the information we intend to use. For instance, it holds a telephone number we are rehearsing (to use it in a little while). It is also the *place* that sees our inner speech and visual imagery. The audience members are **the processes** that respond to the content of consciousness; they are neuronal networks that perform unconscious functions, widely distributed throughout the brain (massively-parallel processing is distributed over millions of specialized neural groupings; Baars, 1997b). Some are automatic routines, such as the brain mechanisms that guide muscles activation for a gesture, or jaw and tongue muscles that are needed for speaking. Others involve **declarative memories,** which are semantic networks that hold our abstract knowledge of the world (**semantic** memory, for facts and beliefs, and autobiographical **memory,** the subjective memo-

ries of our life), and **implicit memories,** that maintain attitudes, skills, and social *savoir-faire*.

Audience members may come on stage, making new content in working memory available to the next consciousness "oration". The spotlight represents the

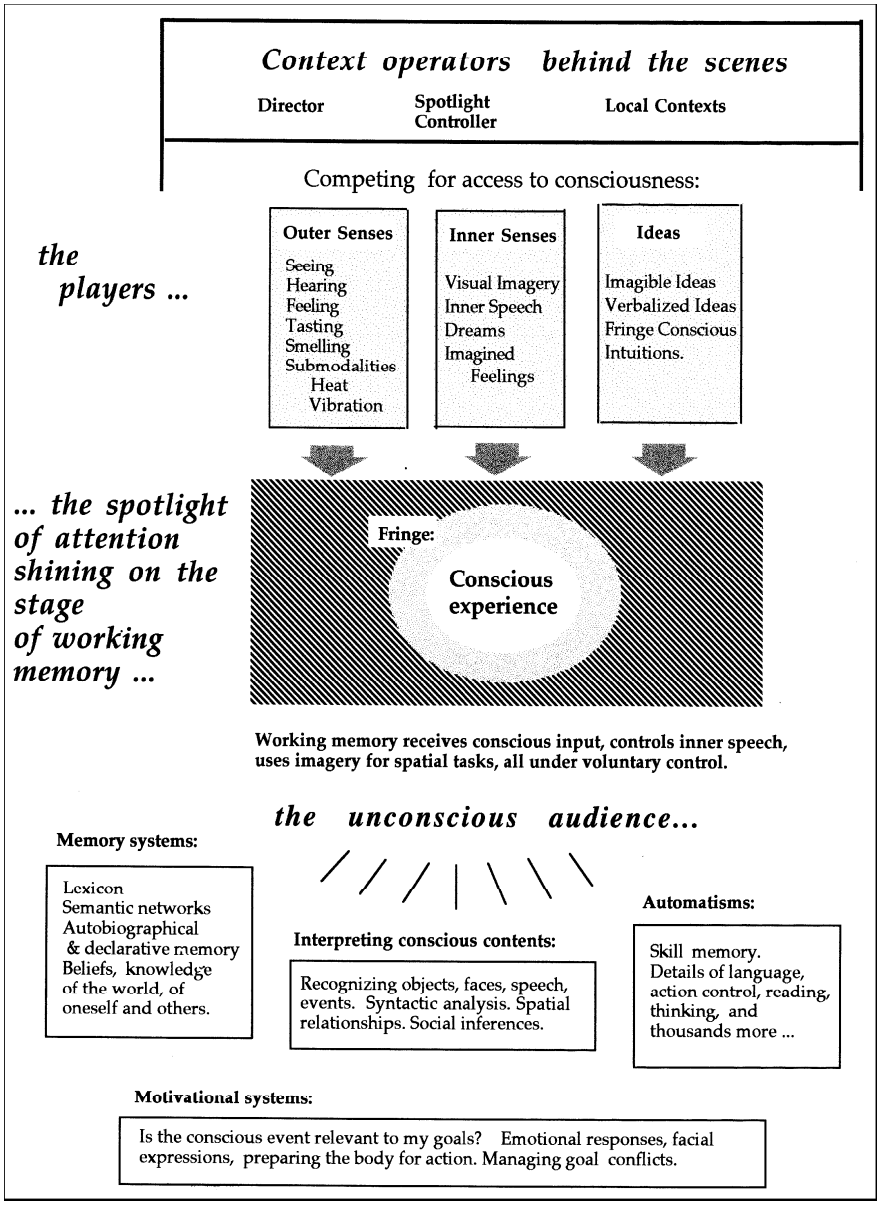



**Figure 1 Baars' interpretation of the theater metaphor.** *Source: Baars, 1997b, p.42.*

**attention mechanism** that makes an information (a coalition of information fragments or aspects) become the one presented to the audience (to all of the unconscious processes). What is in working memory but not under the spotlight of attention creates the feeling of knowing (Baars, 1997b, p.44). William James (1889) called it the *fringe consciousness*. It is what creates the sense of familiarity, or the inclination of thinking something to be true, without being able to pinpoint the conscious event that is the source of such impression.

The Director represents the Self. Baars relates it to the frontal cortex exerting a voluntary control over parts of working memory to request what will come into consciousness next, sometimes redirecting the current stream when something more urgent happens.

The back-drop of the stage represents the **contexts of interpretation** that have been primed by anterior conscious contents. They are **semantic networks** that supply possible referents; neural networks implementing **implicit memories** that encode frames of knowledge; automated processes; attitudes that feed expectations forward. They all tend to have their information connect to the current conscious content and so orient the final meaning extracted from what is declaimed on stage.

### 2.3.3 Some specific ideas proposed in the theory

**A collection of distributed specialized networks**

The brain can be viewed as a collection of distributed specialized networks, most of which do not directly support conscious experiences (Baars, Ramsoy and Laureys, 2003)

**Consciousness limited capacity**

As Baars states (Baars, 1997b, p.43, 54, 56), psychologists believe that consciousness is capable of containing only one chunk of information (simple or complex idea that makes sense on its own) at a time. Or, to put it in a way we are more familiar with, we are capable of sustaining only one idea at a time (or, in Baars' terms,

of only one coherent event, or unified experience, in each moment), although successive ideas may form a chain of ideas that flows very fast, giving the impression of entertaining many ideas at the same time. We may jump from idea to idea pretty fast, maintaining alive a few items in our working memory (seven, plus or minus two) – but not simultaneously as conscious content. That limited capacity, which forces a serial processing, is in shocking contrast with the massively parallel unconscious processing. Would it not be much more efficient to sometimes be able to voluntarily process many things at a time? Of course it would, and we are capable of this! But at a very limited level, with just a few automated processes at a time (Baars, 1997b, p.96). And, in fact, these processes are not *controlled* through consciousness, but simply monitored (with only inconsistencies being brought into working memory; Baars, 1997b, pp. 134-136, 116). Consciousness limited capacity *is* in fact for the efficiency of the system, implementing trade-offs between energy consumption and benefits in our ability to plan, to control ourselves, and to think (Baars, 1997b, p. 57). The one thing that comes into consciousness is what appears to be the most important information at that point in time, all things being considered (we will come back to these last few words in the next point). That way, the consciously mediated process, heavy on resources, is restricted to acting where it counts most. Consciousness' selectivity produces a reduction in complexity (Baars, 1997b, p.55).

Another point about this limited capacity is related to the next one, as it has to do with collaboration and competition among information sources. Because only one coherent idea may come to consciousness at a time (inputs incompatible to the current content are excluded; Baars, 1997b, p.43, 54), grouping various aspects under higher concepts allows processing more information at a time, optimizing our consciousness space usage. So, systems that collaborate and supply sub-ideas that form a coherent concept produce more enticing information for the attention.

**Collaboration and competition**

"All things being considered", as mentioned in the previous point, refers to the "global" conclusion brought to working memory, drawn from the parallel processing made by multiple mind structures operating in the darkness of the theater. At the

same time as the potentially conscious brain activities may collaborate to create a rich and strong description of the situation, various other coalitions that are forming about something else can compete for access to the limited-capacity neuronal global workspace capacity (generally called "consciousness" in this section).

**Convergence and divergence**

What comes into working memory may be the result of the collaboration of many structures, reinforcing one another and ultimately having the result of their collaboration come into the conscious bright spot. This reflects a process of *convergence* that consciousness forces. In Baars opinion, this is perhaps the single most important feature of consciousness (Baars, 1997b, p.162), and is well represented in the theater metaphor: it combines *convergent input* with *divergent output*. Whatever comes to mind reflects a compromise between competition and cooperation, fusing whatever is compatible and excluding for the moment anything that is not (Baars, 1997b, p.52). Then, what occupies consciousness is pushed out, diverging toward the vast audience of unconscious processes.

**Recruiting of unconscious resources**

The three previous features (limited capacity, collaboration, convergence/divergence) add up to say that consciousness is the gateway to the unconscious mind. This idea opens up to another consciousness' feature: recruiting unconscious resources. William James' *ideomotor theory* corresponds well to this idea, showing how a conscious goal can recruit and activate automatisms to carry out a voluntary act. Conscious goal images serve to organize and trigger automatically controlled actions, when not opposed by an inhibitory idea. James proposes the amusing but powerful example of the debate that precedes getting out of bed in a cold morning. Opposite wishes may meet in our conscious mind: the desire to rise and fill our normal obligations, and the desire to remain in the warm bed. At some point, we may resolve to get up, putting in our mind the goal of getting out of bed, and then we just do it, thanks to involuntarily coordinated responses from our muscles through an (usually) involuntary script. However, the stronger illustration of James *ideomotor* theory lies in the occasions where, during debating whether to get

out of bed, our mind starts to wander on a new stream about our daily obligations and routines, creating stronger resolve about getting up, and, most of all, bringing a salutary lapse of consciousness about the cold that awaits us outside of the bed. We suddenly realize, after the fact, that we just got up. The original conscious goal of getting up ceased being inhibited, or counter-balanced, and played its role of "calling condition".

**The Director and the Self**

The Director backstage involves a set of deep layers of expectations and intentions about the world. The "self" of everyday life can be seen as a context that maintains long-term stability in our experiences and actions (Baars, 1997, p.142). William James sees two aspects to the self: the self as agent, and the self as observer. The "agent" part of the self is constituted by the processes that maintain a goal hierarchy that distinguishes long-term goals, such as survival, from momentary goals like reading to the end of this sentence (Baars, 1997, p.143). They are intentions with various temporal spans.

William James' "self as an observer" may be understood as a collection of "pattern recognizers" (Baars, 1997, p. 144), a notion that Block sees as part of monitoring consciousness. These processes constantly compare the current experience to immediate memory, routine personal facts, personal "marker" memories, and future plans or fantasized images. In addition, we have expectations about our abilities; we expect to perform some action in some way, and bodily sensations that do not match are signaled. Similarly, the unexpected absence of the ability would create great surprise; in the same way, the loss of expected memories may impact one's sense of self, as would sudden blindness (Baars, 1997, p.153).

Discrepancies are noted and reported in working memory. If published, they trigger various systems, among which autobiographical memory, that will recall how beneficial or painful that experience has been in the past; it may also trigger attitude processes, which will send stimulation to other systems. These pattern recognizers may declare expectations (things we came to like, fear, or hope for).

### 2.3.4 The functions of consciousness

Baars' theory includes the explicit enumeration of consciousness' roles[9]. Nine points organize the many phenomena associated with consciousness.

1. Creating access to unconscious resources

   The most prominent function of consciousness is to increase accessibility between otherwise separate sources of processing and of information. Everything is connected to most everything else via the bright spot onstage. Most other functions use this one. Some nervous systems (or functions) are reputed as being unreachable by design. Even there, Baars describes an experiment that may prove this wrong: learning to control a number of physiological functions thanks to immediate conscious biofeedback (Baars, 1997b, pp. 58-59).

2. Prioritizing

   Some things are more important than others, such as imminent dangers, the prospect of a very pleasurable activity, or the sound of one's own name spoken in a buzzing crowd. Unconscious processes monitor our senses and may bring a stimulus that requires breaking through to consciousness. But, whether coming from a voluntary thought or popping up from the unconscious, simultaneously occurring ideas may be willingly compared and prioritized for an orderly utilization or simply to choose the most appropriate one. It must be pointed out that this prioritizing may happen unconsciously in working memory, with only the most important information finally coming to consciousness. This is what happens in experts, who come to progressively automate processes; it also happens in intuitively-inclined persons.

---

[9] The exact list and the order of the functions vary a little between the books and the papers; I offer here an arrangement that tries to reflect best Baars' writings.

3. Using unconscious error-detection and correcting defective perceptions

   If we hear a sentence that contains a lexical or semantic error, the problem pops-up to our mind without any voluntary analysis. Expectations about the phrase structure and coming words have not been satisfied. Unconscious processes always monitor our senses in many ways and at many levels. When theses processes cannot themselves find the right correction (for instance, automatically replacing the faulty word by the strongly expected one), they need other processes to take over. Bringing the problem to consciousness presents the situation to all the unconscious processes, some of which, in this case, will propose fixes that allow the sentence to regain meaning, and that satisfy the context.

   In the same way, perception is about giving meaning to stimuli. If the perceptual process cannot interpret a stimulus, this fact brings the executive processes to devote more attention to that process. Consequently, if what is delivered after perceptual processing is in discrepancy with past recordings, with our semantic knowledge, or with expectations, that fact will be submitted for becoming conscious so that other processes may suggest fixes.

4. Problem-solving and plans editing

   Consciousness allows the presentation of ideas, situations, and problems to the unconscious audience so that they analyze them and suggest a solution. Consciousness makes it possible to use the tremendous power of the millions of specialized neural groups, otherwise unreachable by any act of will. They may then supply the most appropriate plan or the proper information to face a situation. Episodic memory may, for instance, bring back the information about where I parked my car. Consciousness may serve as kind of a blackboard to elaborate a completely new plan or procedure if none came up, or modify a plan that analysis revealed inappropriate. Prearranged or generic plans rarely fit the situation at hand.

5. Adapting mental structures for learning

   Learning new material, as researchers like Piaget have explained, is more than plain memorizing into long-term memory. To become useful knowledge, it must be integrated into mental structures we already possess. The more the new information differs from our existing knowledge, the longer it takes to modify the existing structures, and the more it requires the involvement of consciousness to keep the information alive while the knowledge structures are being modified by unconscious processes.

6. Reflection, self-monitoring and executive control

   Through inner speech and imagery, we can reflect upon, , trouble-shoot and modify our own functioning. The self is quite involved in these operations. It supplies the baseline to compare to the actual experience. It often influences decisions at an unconscious level, but it may manifest itself at the *feelings* level (when one does not try to suppress them). Self systems located in the prefrontal cortex probably exercise their control by means of influencing conscious ‘publicity’, never entering consciousness directly.

7. Creating the context for understanding

   The context is the combination of many unconscious networks that shape conscious contents: goals (conscious or not) in their many levels and types, the self, those representing the situation.

   Reactions of the system are, in part, the result of past and current goals, which are presently conscious or came to consciousness some time before. These goals have primed mental structures, including semantic networks, creating a "context of understanding" which favors those structures (they should respond first to the content of consciousness).

   Other aspects of the context (self, expectations, state of the perceptual networks, emotions) also react to, influence somehow, even constrain what appears in consciousness. They orient what will ultimately be the global meaning of the perception or, more generally, the conscious experience.

For instance, contextual parietal maps of the visual field, which do not support conscious features, modulate visual feature cells that directly contribute to conscious aspects of seen objects (Baars *et al*., 2003).

8. Optimizing the trade-off between organization and flexibility

   Automatic responses are highly adaptive in predictable situations. However, in the face of novelty and uncertainty, the capacity of consciousness to recruit and reconfigure specialized knowledge sources becomes vital. This being said, given no time and great urgency, only prepared actions are serviceable (Baars, 1997b, p.160), as there is insufficient time to make a long analysis, organize a thoroughly worked-out plan, or even simply adapt a script. Two phenomena may force this compromise: either an automatic reaction has already been put in motion when inhibitory information comes to consciousness, or, since in such situations all the conscious space is already filled with uncontrollable, task-irrelevant thoughts, conscious volition is struck-out and will not be serviceable until one calms down.

9. Recruiting and controlling actions (James' ideomotor theory)

   As illustrated with the difficult morning decision about getting out of bed, putting that goal in my mind is sufficient to have an uncontrolled script (automatic routine) fire-up, if no counter-acting idea shows up. Conscious goals serve to mobilize automatic routines and body muscles in order to carry out voluntary actions.

   Similarly, entertaining a thought about a life-threatening situation is sufficient to mobilize autonomic arousal and prepare rapid muscular responses.

As you can see, Baars' theory contains rather high-level ideas and descriptions. But they are well organized and offer an interesting framework for a computer transposition. You will see an example of such a work in two chapters. Before coming to this, I offer in the next chapter kind of a baseline, with an overview of existing solutions for consciousness models and "conscious" agents.

# Chapter 3

# CONSCIOUSNESS ARCHITECTURES AND "CONSCIOUS" AGENTS

## 3.1 PREAMBLE: WHY FAVOUR AGENT ARCHITECTURE?

Building tutoring systems as an agent (or as a multi-agents system) is the main stream of the recent years in the ITS community. But before turning to agent concepts, computer-assisted learning systems (CALS) were designed within the conventional paradigm of subsystems that perceive, process and react. *Agents* also do that, but they go further, as I will briefly describe in the following lines that essentially reproduce Franklin and Graesser's comprehension (Franklin and Graesser, 1997).

I must point out, at the onset, that the word *agent* has an unclear definition; there is not consensus on what an agent incorporates, as exposed by Franklin and Graesser (1997). I will adopt the proposition of these authors to understand the concept as *a system situated within and a part of an environment, that senses that environment and acts on it, over time, in pursuit of its own agenda*. That definition in itself poses a problem as it contains an implicit reference to *autonomy*, a difficult concept to pin down precisely. Jennings *et al.* wants to convey the simple idea that the system should be able to decide and act without the direct intervention of humans (or other agents), and should have control over its own actions and internal state. Autonomous behavior is not a new idea. It has been implemented in numerous applications: we find these capabilities in process control systems, which must monitor a real-world environment and perform actions to modify it as conditions change (typically in real-time); we also find them in software *daemons*, which monitor a software environment and perform actions to modify the environment as conditions

change. However, these systems cannot be called *intelligent* agents. When we add "intelligence" in the picture, we get the finer definition of an intelligent agent as a computer system that is capable of *flexible* autonomous action in order to meet its design objectives. By "flexible", Jennings, Sycara and Wooldridge (1998) mean that the system must be:

- responsive: an agent should perceive its environment (which may be the physical world, a user, a collection of agents, the Internet, etc.) and respond in a timely fashion to changes that occur in it,
- proactive: an agent should not simply act in response to its environment; it should be able to exhibit opportunistic, goal-directed behavior and take the initiative where appropriate,
- social: an agent should be able to interact, when it deems appropriate, with other artificial agents and humans in order to complete its own problem solving and to help others with their activities.

It is the presence of the four components in a single software entity (autonomy, plus the three sub-components of "intelligence": responsiveness, proactivity and sociability) that makes for the originality and power of the agent paradigm. Hereafter, when I use the term ‘agent’, it should be understood that I am using it as an abbreviation for the rich definition of ‘artificial intelligent agent’.

Just a little thinking makes it obvious that a tutor (human or artificial) has to be able to perceive his environment (including the learner) and possess the autonomy that allows him to react *or act* in the most appropriate way, at the right time. That is, he has to be able to seize the context, recognize trends, foresee consequences, plan and adapt on these bases and act to try producing the most appropriate result in the context. He might need to interact with other agents to reach that goal. This description goes beyond the capability of a conventional system and justifies the point of adopting an agent paradigm.

The appropriateness of the agent paradigm being clarified, I now present different implementations of agents that attempt at capturing, or at least use, some of the features characteristic of human consciousness. I also cover some implementations that are not agents by themselves but offer the tools or framework that can support one. Taking a stroll along this overview, even if limited, will give a better perspective on CTS, which I will describe in the next chapter.

## 3.2 VARIOUS APPROACHES TO CONSCIOUSNESS IN THE AI FIELD

AI has integrated human consciousness in its realizations long ago. Bechtel (1995) recalls this fact quite elegantly, stating that many aspects proper to consciousness seemed critical to any successful information processing model. For instance, an interactive program (and more recently, *agents*) shows selective attention, either by design (with limited sensors), or by prioritization. Some of the captured data is considered, but much is left ignored, as the mass of irrelevant stimuli in the real world would overwhelm the processes. Another parallel between technical artefacts and human consciousness holds in computers central control systems, typically summed up in the acronym CPU (Central Processing Unit). When CPUs get involved in the processing, they mimic the non parallelism of consciousness, churning one item at a time[10] from what is fed in its stack by the multiple autonomous co-processors and sub-systems working in parallel. A third example can be given in the subsystems sending to the CPU only a fraction of their conclusions, making all they can on their own, in the “unconscious” of the computer, bypassing the central processor as often as they can and having direct communications to other sub-systems. If needed, some of their work and some of the internal states of the computer can be made available, “bringing them to consciousness” so that some process can report

[10] See note 4 about CPU's seriality.

on them to computer designers, or so that direct actions can be taken accordingly by safeguard processes. Hardware people might balk at looking at computers architectures as emulating consciousness and unconsciousness. Nature often inspires us without our realizing it, and it is sometimes difficult to admit that our great ideas are simply an intuitive transposition of what already exists in nature…

Authors such as Johnson-Laird (1988) saw no shame in having an explicit, inquisitive look into consciousness, trying to understand its functions to implement them into computer algorithms. Paillard (1999) explains that Johnson-Laird was positive about the fact that those “thinking” machines, computers, can generate functions analogous to *becoming conscious*. However, he remained sceptical about their usefulness for computers' “mental” operations and their performance.

This kind of scepticism seems to have somewhat eroded over the time. Researchers keep asking questions about the usefulness of consciousness for robots (or agents in general), but not anymore as a doubt, but a lighthouse’s beam to follow, an obvious goal to reach. Recent researches, often stimulated by discoveries in neurosciences, aim at integrating consciousness in various artefacts: models of the human mind, models of consciousness, computer implementations of the models. We see scientific communications proliferate on the subject. Conferences are created not only in the field of philosophy, but also in events assembling AI leaders. In 2001, a three-days multidisciplinary workshop headed by Christof Koch (one of the authors of the biological 40Hz synchrony model), Chalmers, Goodman, Holland and Schwartz, had for theme «Can a machine be conscious?». At the end of the workshop, Koch inquired to the twenty researchers on how many would now give a positive answer; all but one raised their hand. The theme had gone from an interesting subject to a clear and stimulating prospect. In 2003, another similar workshop had the objective of identifying the aspects in the diverse consciousness models which

could be implemented in computers or robots and explain the experimental data (Sloman and Chrisley, 2003)[11].

Indeed, many researches aim at creating either a *functional* implementation of consciousness or an "authentic" artificial consciousness (biologically plausible). I will present a few of them that cover a spectrum of possibilities. Franklin (2003b) mentions some examples of such serious projects that I will not cover here: one headed by Igor Aleksander, MAGNUS, uses neural modelling; another one inspired by neural modelling is the proposal of Lee McCauley that builds consciousness into a neural schema system; Owen Holland and Rodney Goodman follow a bottom up approach, adding capabilities to a robotic system until it shows signs of consciousness. Many more exist, inspired by different horizons and field of interest, some with similarities, most with a specificity that would be worth mentioning.

I classify the systems that I will present under the following classes:

**Functional implementations** want to reproduce the *roles* held by consciousness. Two subtypes exist.

- *Purely functional implementations*. Here, all is sought for are the alleged benefits coming with the consciousness mechanisms (for one, the mode of operation it enriches mind with). Whatever way is used to render them is fine. You will see here the reflexive computer language of McCarthy, and two "conscious" agents: GLAIR and Introspect.
- *Psychologically plausible functional implementations*. In these cases, the authors try to respect some plausibility, for instance by founding their work on a psychological theory of the mind and consciousness (the Conscious Mattie/IDA/LIDA family rooted in Baars' Global Work-

[11] Other examples: in 2003, ASSC 7 Symposium in Memphis; the 2003 ESF exploratory workshop "Models of Consciousness", in Birmingham; the 2004 "NoE 'Exystence' in Turino; in 2004, the parallel session at the ASSC 8 in Antwerp.

space theory). ACT-R also fits in this category, although consciousness was not at the root of the project. I will present ACT-R in a separate section (6.2), in a comparison to CTS. IDA/LIDA description permeates this whole document as those agents found CTS, so no section will be devoted to them; differences are pointed out in italics text when a CTS feature is presented.

**Biologically plausible implementations** want not only the results of consciousness, but a closer relation with the low level of the "biological tissues" that are thought of as supporting consciousness. CyberChild appears in this category. I will also briefly present a neuron network that attempts to explain with some level of biological plausibility how the mind learns and how one could derive consciousness from it: ART.

### 3.2.1 Functional approaches

#### 3.2.1.1 A Computer Science approach to consciousness: McCarthy's reflexive language

Among the firsts to propose the possible benefits of tracking and inserting human consciousness features in robots is John McCarthy (McCarthy 2002/1995; 1959). He proposed mechanisms and a logical language making possible to reproduce some of human consciousness functions, including metacognition and introspection, which he posits as equivalent to self-consciousness. The robot's beliefs are directly accessible in the computer's working memory, forming its awareness. Some permanent processes running in parallel can generate sentences about the beliefs. These comments on the beliefs create the robot's consciousness. Other sentences come into "consciousness" as the result of introspective actions the robot decided to make, and create its self-consciousness. McCarthy conjectures that ro-

bots will need meta-sentences and better abilities to comprehend so that they understand how they do things and can improve.

Summing up, McCarthy proposal uses words associated to consciousness («consciousness», «unconscious», «introspection», «awareness», «contexts», «free will») and proposes clever mechanisms for them. However, he admits not being interested in “real” consciousness and makes no attempt in this direction (for instance, the “consciousness” the talks about is a specific “place”, a subset of memory). By no means does he feel bound by any human limitation ("many features of human consciousness will be wanted, some will not" – in his opinion, not everything in human consciousness is useful for intelligent behavior). His robots’ unconscious mind can be inspected at will. This is a pragmatic, engineering view, with a priority on getting results. It obtains some benefits from a distant observation of consciousness, but makes no attempt at explaining anything. In my opinion, it does not reap the true benefits offered by human consciousness.

#### 3.2.1.2 Hexmoor, Lammens and Shapiro's GLAIR (1993)

GLAIR (*Grounded Layered Architecture with Integrated Reasoning*) uses an architecture with three layers relating to the conscious/unconscious arrangement of the mind (see Figure 2). The two first layers process “unconsciously” what is sensed, deciding on the right action to take with their automated capabilities. The third, top layer is said to be “conscious” and is concerned with the tasks requiring deliberation for the adaptation to new situations. Albeit “on top”, this layer does not take on any coordination role.

The creators of GLAIR define an agent’s consciousness as the awareness it has of its environment. It takes three forms: (1) internal states or representations causally connected to the environment through perception and action, (2) explicit reasoning capabilities about the environment, and (3) its ability to communicate with an external agent about the environment (“reportability”).

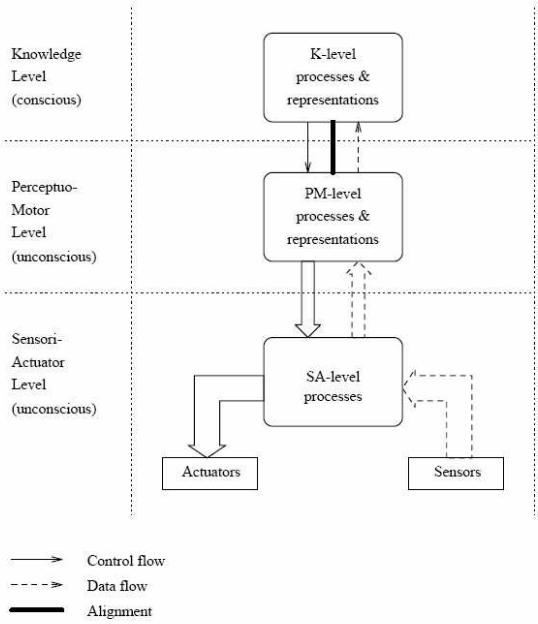



**Figure 2 GLAIR's architecture.**

The three layers operate in parallel but collaborate: the conscious reasoning guides the unconscious (automatic, reflexes) behaviors while these, constantly processing the inputs and preparing the outputs (the motor actions), can alarm the conscious level about important events. In case of such events, the conscious level may take control of the agent. So, action selection and monitoring is not confined to a specific level. Moreover, explicit rules elaborated by the conscious layer are transferred to the lower levels in an implicit form, where it is learned as a state transition. The next time the same conditions appear, this transition will automatically be selected without any recourse to deliberation.

The architecture has even more interesting functions. It possesses reflexive and metacognitive mechanisms that evaluate actions value based on results. They serve in the agent's improvement. First, they identify frequent sequences of actions. If the routine can be associated with an improvement of the situation in the environment, the agent believes that a valuable routine emerged from reflex actions, and it augments its confidence in the sequence. When this confidence reaches a threshold,

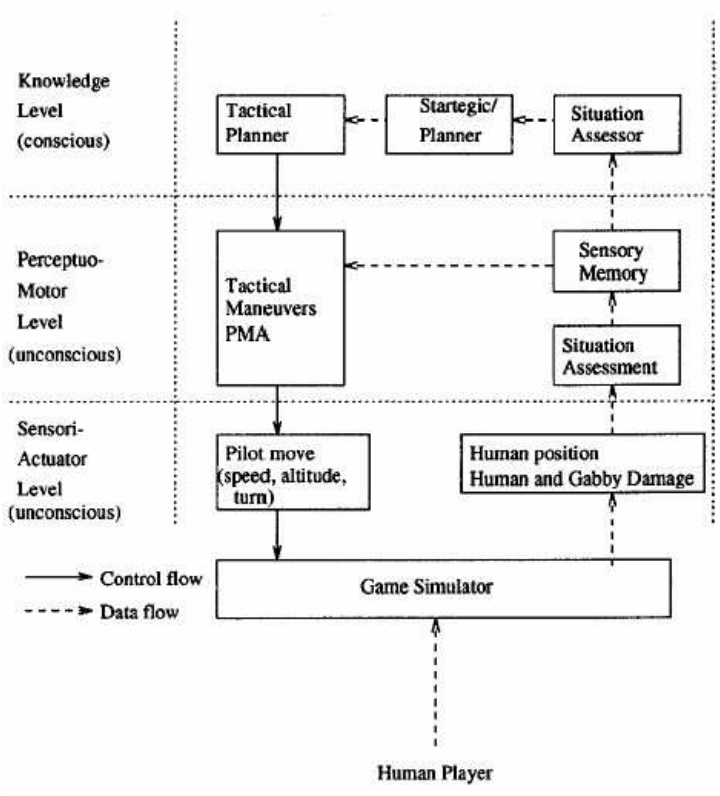



**Figure 3 GLAIR air battler's (Gabby) architecture.**

a *concept* is created in the top level, which deals with explicit knowledge; this concept will be available for ulterior reasoning. So, from its interactions with the environment, the agent creates its own concepts. Self-observation and reasoning allow the agent to improve its efficiency in choosing a behavior, and its abilities to act, all towards achieving its goals.

GLAIR has been tested in air-combat simulations (named *Gabby* in this videogame, for “GLAIR air battler”). In non-learning mode, it lost nearly 70% of the battles. When it has been allowed to learn, the agent rapidly became more reactive (reacting more rapidly) and eventually won 50% of the combats. This is a convincing demonstration of the value of this architecture’s self-observation and auto-modification.

So, while not referring to any global theory, the concepts GLAIR uses show their usefulness. It presents an interesting combination of “unconscious”/automatic and “conscious”/deliberative mechanisms, balancing immediate efficiency and adaptability.



**Figure 4 Comparison of developmental algorithm followed by the team of expert-programmers, and the one followed by Introspect.**

3.2.1.3 Cazenave's Introspect (1998)

Cazenave has produced an agent capable of observing the results of its real-time actions (as resulting from its current know-how), of evaluating how well it had predicted the results, and of finding the failings in its plans to correct them and improve its performance. He demonstrated the value of his proposal by applying it to a Go player. *Go* is a very popular Chinese game of life, and the most complex two-players game. Learning it takes years for humans, and transferring expert knowl-

edge into a program to a proficiency level borders on impossibility. An agent self-observation and auto-improvement is the solution offered by Cazenave.

In spite of the simplicity of its rules, playing the game of Go is a very complex task. It is impossible to make a brute force search of all the moves in the game, and the best Go playing systems all rely on a knowledge intensive approach. Traditionally, expert players team with programmers to extract and encode knowledge, in a conventional knowledge design approach (for expert systems). Due to the high specificity of the situations, learning time is enormous and learned rules tend to become unconscious in the experts. One would be tempted to log the moves made by two players during a great number of games and throw a machine learning algorithm at it. This is somewhat what Cazenave suggests, but instead of observing from scratch every time, he proposes a system that builds rules on the go (no pun intended), and then uses “conscious introspection” to identify new rules, find errors in existing ones, and accommodate this new knowledge.

After observing the state of the Go board, Introspects makes all inferences it can with its knowledge of the game, and records these deductions. Then, it chooses and applies a move, and deduces all it can from the resulting configuration. It compares the prediction with the actual result. If something unexpected of interest is discovered, something it was not capable of anticipating, it tries to find, by backward chaining, the source rule that needs to be modified or that should have been involved before deciding on the move.

The algorithm also tries to generalize the rules, replacing constants by variables; it “forgets” those that are now part of the generalization. It completes its memory/time optimization with some meta-analysis that kills *harmful* rules. In Introspect, "harmful rules" are those which have a high probability of failing, either on the count of too many conditions to match, or too many actions to take afterward. The more conditions are to be fulfilled, the more the rule becomes likely to add match time without being applied; action lists with more than five actions to fulfil are rules likely to fail (according to Cazenave experience; Cazenave, 1998, p.3). Finally, a compilation of the rules transforms them to an “unconscious”, implicit form.

Introspect is an interesting example of AI finding inspiration in cognitive researches to construct an agent. It parallels some features of consciousness (using ideas from Minsky and Sloman) to obtain a superior performance. It mimics *short-term* memory utilization, reflexivity (introspection), deliberation, metacognition and implicit learning. The passage from explicit knowledge to implicit is only a matter of compiling the knowledge, which makes short work of the humanly process! Another negative small point is that the resulting agent does not possess human's reflexive and metacognitive capabilities on-line; improvement of its abilities comes only with an off-line process; Introspect is a Go tournament player, and has to live with time constraints. However, adorning it with on-line adaptive capabilities could easily be done. In any case, just like humans, it operates on the principle of trial-and-error, practice-and-improve to perfect its abilities. Although it does not try to explain anything, it is nice to see the application of ideas about consciousness in real, efficient applications.

### 3.2.2 Biologically-motivated approaches

I call "biologically-motivated approaches" those that try to mimic nature. Some very far-fetched researches attempt to create human tissue through biomedical engineering, but we are very far from anything that will lead to a brain. The closest things to human neural circuitry still exists only in computer simulations (for instance, de Garis' project in Starlab to build *artilects*, "artificial intellects" upon 100 million artificial brain cells, in a 2001 description of the project[12]). More "traditional" approaches are the ones from Grossberg and from Cotterill. I describe them hereafter.

---

[12] http://www.kurzweilai.net/meme/frame.html?main=/articles/art0131.html

### 3.2.2.1 Cotterill's Cyberchild

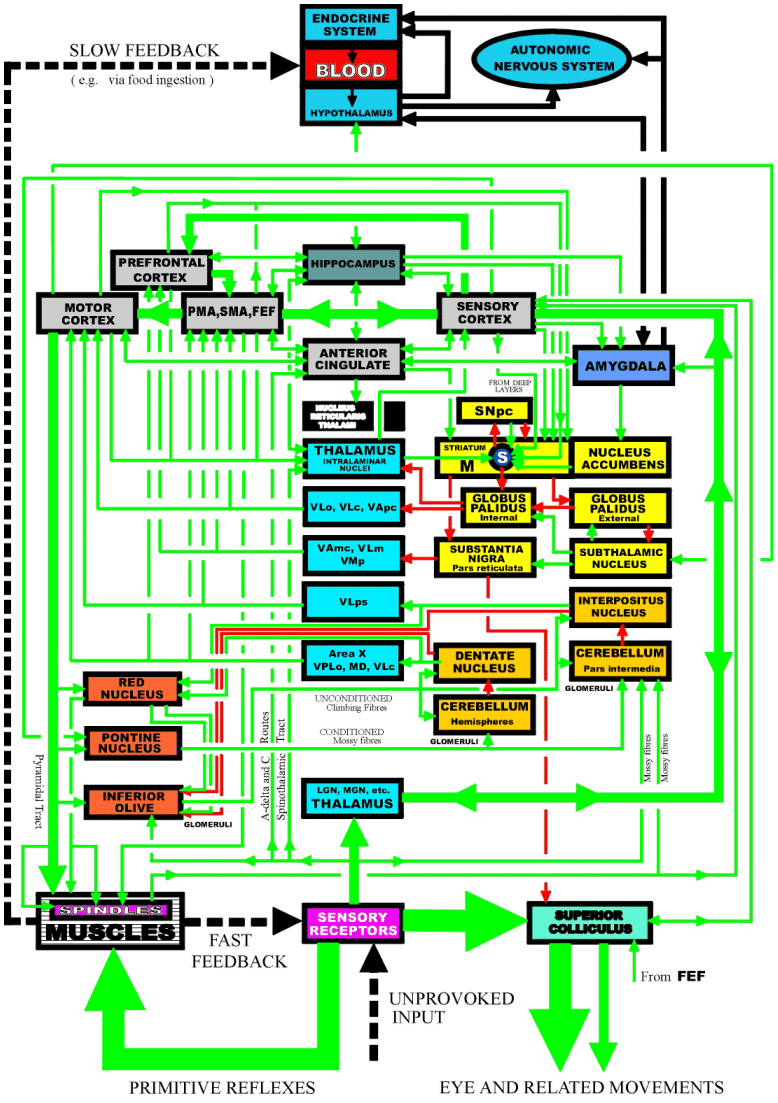



**Figure 5 Cyberchild's architecture.** Cotterill wants his project to show not only a brain, but also biological functions and motivated emotions. Consciousness will eventually emerge from it.

Igor Aleksander describes Cyberchild as «An accurate biochemical model of a young baby.»[13] Although purely a computer simulation, Cyberchild is meant to be faithful to its model, a human child that has everything to learn. With metabolic functions (bladders, blood stream with nutrients, a stomach that digests), it has needs, and experiences emotions. The *child* has to learn to behave correctly so as to receive what it needs from the experimenter. Its brain is quite detailed. Rodney Cotterill explains:

> The underlying model is based on the known circuitry of the mammalian nervous system, the neuronal groups of which are approximated as binary composite units. The simulated nervous system includes just two senses — hearing and touch — and it drives a set of muscles that serve vocalisation, feeding and bladder control. These functions were chosen because of their relevance to the earliest stages of human life, and the simulation has been given the name CyberChild. The system's pain receptors respond to a sufficiently low milk level in the stomach, if there is simultaneously a low level of blood sugar, and also to a full bladder and an unchanged diaper. It is believed that it may be possible to infer the presence of consciousness in the simulation through observations of CyberChild's behaviour, and from the monitoring of its ability to ontogenetically acquire novel reflexes.[14]

Cotterill thinks that sophisticated neural apparel is a prerequisite to consciousness. It must allow, among others, for the attention, re-entrant neuronal loops, and brain's plasticity. Everything is set up so that the *child* can do an authentic exploration of his universe, can learn and make inferences, and eventually let us see his consciousness emerge. In 2002, Cotterill did not think he saw any consciousness evidences in CyberChild. But he was not "cyberchilled" so soon…

Even though Cotterill demonstrates a very honorable candor when he does not see traces of consciousness in his CyberChild, all the apparatus seems in place for it, if complexity or grounding are conditions to consciousness, although not in real flesh and blood. In fact, it possesses the mechanisms that correspond to other agents and

---

[13] http://www.cs.stir.ac.uk/~lss/BICS2004/Tutorials/AleksanderTutorial.pdf

[14] http://www.imprint.co.uk/jcs_10_4-5.html#cotterill

other models deemed conscious. Moreover, the consciousness that will eventually emerge is in good position to be quite believable since it is totally *grounded* to the agent's environment (that is, linked to the environment's stimuli, which are then *perceived* by processes that are in accordance to what we know of human cognition.

### 3.2.2.2 Carpenter and Grossberg's ART (1976; 1987)

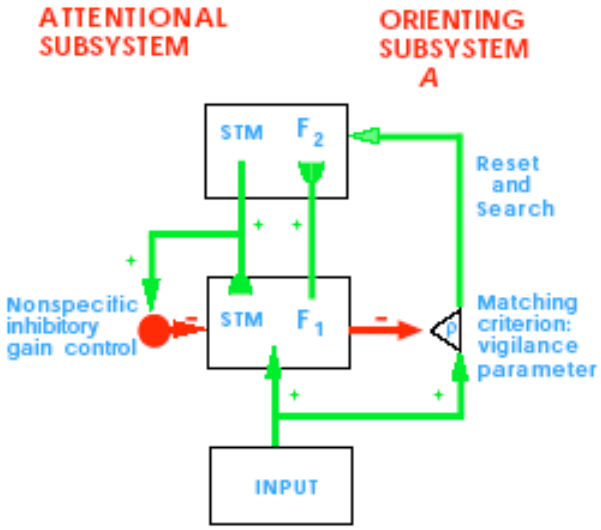



**Figure 6 Fundamental principles of ART.**

Well known for its applications to categorization and learning, ART (*Adaptive Resonance Theory*) could easily go unnoticed when talking about consciousness. Indeed, the basic ART system is usually classified an unsupervised learning model. Yet, in a 1999 article, Grossberg affirms that ever since its inception, a central hypothesis in ART poses conscious states as *resonant* states (states that lead to the recall of existing memories). Resonant states are what ART is about, thus consciousness concerns were present at the very beginning of ART. He adds that those processes that allow our brain to learn over a lifetime while maintaining its stability (remaining organized, not becoming chaotic) create conscious experiences. Conse-

quently, only those resonant states should be learned. The mechanism that forms the resonant loops takes time to stabilize, which corresponds to the delay observed between a stimulus and the report a subject is able to do about it (after becoming conscious of it).

The general mechanism that makes us learn while preserving the existing knowledge is based on expectations that center our attention on stimuli having value. The focus of our attention becomes confirmed when a resonant state emerges from a feed-back loop. This can only happen when the bottom-up signal (coming from the input) corresponds to the top-down signal (the expectation). The latter is prepared and oriented by the priming mechanism. It stimulates, ahead of time, cells (conceptual nodes) that should react to the sensory information, amplifying some characteristics and inhibiting cells of which no activity is expected. This process filters out "noise" that would otherwise rapidly destabilize past acquisitions. After stability is obtained, the resonant state locks up the activity pattern at a much higher activation level and makes it last much longer then what would be observed from individual activations. Only these highly activated patterns emerge and remain observable long enough to be learned.

Top-down signals represent expectation learned by the brain about what the inputs should be, based on past experiences. Philosophers often call them *intentionality*. Since past experiences incur intentionality, Grossberg asserts that ART offers the basis for self-consciousness. Carpenter and Grossberg (2003, p.10) cite Pollen as backing their hypotheses and the correspondence of their model with consciousness:

> Pollen (1999) resolves various past and current views of cortical function by placing them in a framework he calls adaptive resonance theories. This unifying perspective postulates resonant feedback loops as the substrate of phenomenal experience. (…) As Pollen (pp. 15-16) suggests: "it may be the consensus of neuronal activity across ascending and descending pathways linking multiple cortical areas that in anatomical sequence subserves phenomenal visual experience and object recognition and that may underlie the normal unity of conscious experience."

ART was at first a theory and a functional recreation of mind attempting to explain categorization and lifelong learning. It keeps growing towards a robust framework with links to experimental data. Consciousness in this framework is becoming less of a peripheral interest, and more of a central concern, as a recent (2005) paper by Grossberg demonstrates: «*Attention, like consciousness, is often described in a disembodied way. The present article summarizes neural models and supportive data about how attention is linked to processes of learning, expectation, competition, and consciousness*». Grossberg deserves credits for offering a viable explanation of how consciousness could emerge and why. He also provides some roles for consciousness.

Many other models of the mind and of consciousness would have deserved being included in this overview: Taylor's models (the relational model of the mind, the ACTION network, etc.), Sun's CLARION, McCauley's neural schemas network, Aleksander's Magnus, Minsky's ideas about the mind, and many others. My first aim for this section was to show some agents that incorporated some form of consciousness; I extended the review to incorporate some famous models of the mind, and an essential historical figure (McCarthy). However limited, this review of some AI's architectural use of consciousness is sufficient to supply us with a much more enlightened look at my own proposal for a conscious agent.

# Chapter 4

# CTS, OUR "CONSCIOUS" TUTORING AGENT

The architecture that I propose for a conscious agent is the foundation for a tutoring agent I called CTS (*Conscious Tutoring System*). CTS is a son of IDA, the agent developed by professor Franklin (University of Memphis) and his team. CTS shares IDA's fundamental mechanisms for consciousness, and some other structures such as a Behavior Network, a Perception Network and long-term memories. However there are differences in the implementation of some mechanisms; I will present them along the way, while touring CTS.

Before starting, I'd like to recall the advisory caution given in Chapter 2. Although I do not put quotation marks around the word "consciousness" when talking about CTS, I do not mean to support the interpretation that CTS consciousness is "real", or on a par with human consciousness.

I also wish to make orthographical and naming clarifications. First, an orthographical convention. Since many of CTS' modules refer by name to the biological function they implement (for instance, access consciousness, working memory, autobiographical memory), there may be confusion as to which side a sentence refers to. I will be indicating CTS' modules with initial capitals (ex.: Working Memory, Behavior Network, Learner Model, *etc*.), whereas I will leave brain's biological "functions" in small caps (working memory, access consciousness, perception, *etc*.). "Codelets", which names do not duplicate biological counterparts, will be left in small caps. For instance, I will explain about CTS that the coalition selected in Working Memory by the Attention mechanism is then broadcast by the Access Consciousness.

As second clarification, I wish to explain that I will be using "broadcasting" and "publishing" as synonyms in the descriptions. I use both to give some variety to descriptions that use them quite intensively and might get a little boring at times!

## 4.1 CTS' ARCHITECTURE

CTS presents a *functional* (it implements brain and mind *functions*), distributed architecture with both high-level entities (modules) and low-level entities (codelets, to be described later on). The coupling between modules is weak, with message exchanges happening mostly (in fact, *exclusively*, for the time being) through the intermediary of Working Memory (WM) and Access Consciousness. It covers every major aspect of cognition, with many functional correlations to the physiology of the brain (see Baars and Franklin, 2003; see also (Franklin, 2003a) for a comparison of IDA with Crick and Koch's framework for consciousness).

Two general considerations have to be mentioned before starting the tour. The architecture that underlies CTS is concerned with consciousness and all the benefits this faculty can bring. To try to reap all the advantages, one has to respect the principles enunciated in an all-encompassing theory, in this case, Baars', and reproduce every aspect of consequence. In this line of reasoning, it would be nice to create all *peripheral* modules in a faithful manner, but is not required. What is really necessary is that they allow the consciousness mechanisms (Working Memory, Attention echanism, Access Consciousness) to work in the fundamental way they have been designed to follow, using *codelets* to communicate with Working Memory and Access Consciousness. So, aside from a communication layer that reads and translates information into information codelet structures, designers of a module are free to use whatever mean they find useful to produce their "unconscious" analyses. This opens the door to an easy integration of any existing module. As an illustration, the Domain Expert and the Transient Episodic Memory show an eclectic collection of techniques that collaborate perfectly to the performance of the global agent.

The second point I need to make is about the width, that is, the number of fields our architecture encompasses. Each of its modules would deserve pages of description to give a thorough account, and each will require its own research program to reach a satisfactory implementation. I will not try to cover every base. This thesis is about the Global Workspace theory, and its possibilities when applied to a tutoring agent. My descriptions will stay within the ideas that this theory offers, allowing the reader to understand the theory, its implementation, and its possibilities.

Along the way, I will indicate major discrepancies with respect to IDA, sometimes to LIDA, with *sentences written in italics*. LIDA (Learning IDA) is the newest member of Franklin's agents family; IDA has been the starting point of CTS and has more direct resemblances.

Now, let's start the tour. To be able to describe many functions of the architecture, I need first to describe special low-level entities: codelets. Talking about them will often bring references to CTS' architecture, so I include its diagram here, but will be specifically referring to it only starting with section 4.1.4.

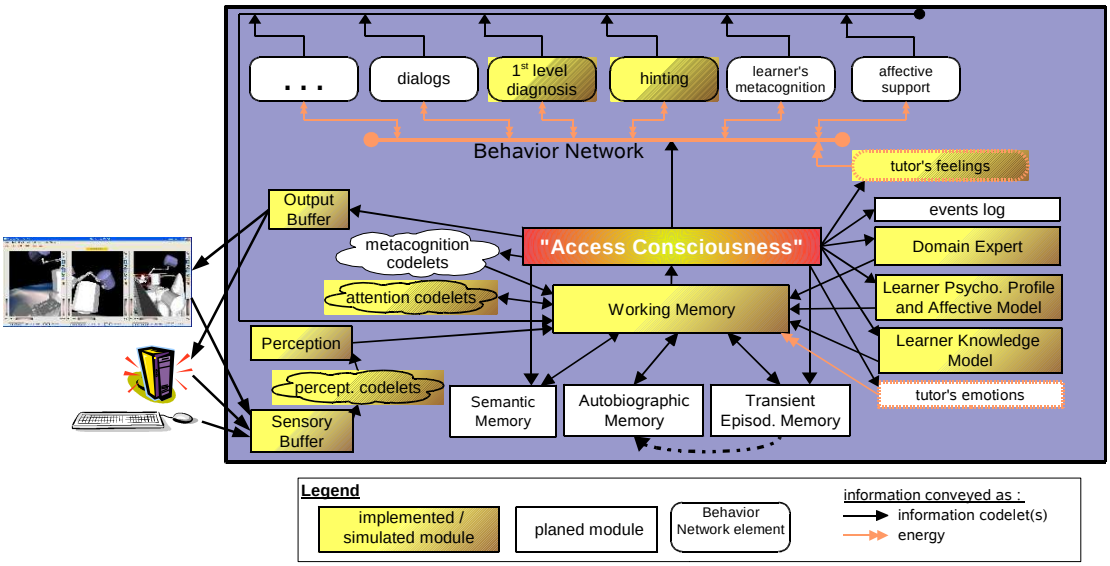



**Figure 7 Conceptual architecture of CTS.** Colored (grayed) boxes indicate which functions are implemented in the prototype. Orange doubled arrows show messages flowing in the form of energy feeds.

### 4.1.1 Codelets

Codelets, although individualistic in their nature, show up on the conceptual architecture only in boxes indicative of "full-status" modules. Some of them effectively render as a group the services of a "virtual" module (attention codelets, metacognition codelets). Others are "hidden" within their module, since they work as part of a higher-level structure (Perception codelets, in the Perception module, and a variety of codelets in the Behavior Net). Still others (information codelets) masquerade as communication arrows in the diagram since they play the traditional role of information vectors.

The name “codelet” has been kept from IDA, which borrowed it to the Copycat architecture (Hofstadter and Mitchell, 1995). It designates simple unintelligent agents that simulate neuronal groups. As their counterparts, they are specialized in their roles, with limited abilities and range, but very efficient. Various types have been prepared that reflect the types of activities (or functions) various neuronal groups may: perception, reasoning (information and attention codelets, the latter including expectation and metacognition codelets), and behavior codelets.

We classify codelets as agents, after Franklin, in the sense of Minsky's Society of Mind (Minsky, 1985). They possess many of agents’ characteristics: they are autonomous, perceive, process, and act. They also do an elementary form of learning in the strength of the associations they create with each other, this mechanism coming from Pandemonium theory (Jackson, 1987). CTS is an agent containing a multi-agent architecture.

Codelets life spans reflect that in the human mind. We, human, have processes always active (or, at least, never very far away) that have to do with basic survival. We exhibit others that exist for an extended period of time (for instance, when playing hockey, the very needed single-minded processes that monitor senses to detect and recognize the arrival of an adversary); they exist at the same time as those related to the survival instincts. We also constantly start some very specific and short-lived ones, such as those that monitor the events after I screamed at the

left-wing player to receive a pass. An even shorter process might be one attending to the sound my car's motor does after turning the ignition key. I may be thinking of something else while I do it, but if the sound is *strange* (differs from what usually happens), I will instantly turn my attention to it.

**Table 4- 1 CTS' codelets taxonomy.**

| Type | Sub-type | Group name | Role |
|---|---|---|---|
| Perception | Perception | Perception codelets | Give an interpretation to what the agent senses from its environment |
| Reasoning | Arbiter | Arbiter codelets | Control the deliberation process |
| | Attention | Attention codelets | • Monitor WM for patterns<br>• Bias information selection |
| | | Metacognition codelets | • Monitor CTS' internal processes<br>• Help regulate and correct processes |
| | | Expectation codelets | • Check that expected results do happen, then either :<br>o strengthen links in the Behavior Net,<br>o put information codelets describing the difficulty encountered. |
| | Information | Information codelet | Represent and transfer information |
| | Encoding | Encoding codelets | Find new information in WM, encode it and feed LTM; decode associations returned by LTM and deposit them in WM. |
| | Emotion | Emotion codelets | Represent affective valuation of information in coalitions of codelets |
| Behavior | Motor | Motor codelets | Act on the environment |
| | Generator | Generator codelets | Create reasoning codelets |

#### 4.1.1.1 Perception codelets

Although they show commonalities with attention codelets, they do not watch Working Memory. They have effects in the processing of sensed information (or data). Their eyes stay riveted on the Sensory Buffer(s[15]), each one looking for one or a few specific patterns of letters. If a perceptual codelet finds what it is after, it transfers the information found to the Perception Network node it is attached to, and it resets the activation of the node to its nominal value (or somewhat less, when habituation kicks in).

#### 4.1.1.2 Attention codelets: Arbiter, attention, expectation, metacognition codelets

A very interesting type of codelet is the attention one. The name designates a category of codelets that include a sub-category with the same name (attention codelets). They are either innate, starting their activity with the start-up of the system, or released by a Behavior node to attend to some matter. In all their varieties, they are pattern recognizers that watch WM.

**Arbiter codelets** watch WM and detect when an information calls for deliberation. In such cases, it will successively play various roles: counting cycles since the last enrichment of the coalition, selecting the most probable cause to attach to a coalition, declaring the end of a deliberation and marking a coalition as apt to enter the competition for the selection by the Attention mechanism.

**Attention codelets** are not to be confused with the mechanism named *Attention* that selects the most activated coalition in Working Memory. Some of them look for a specific word or pattern of letters; some are interested in the appearance of a

[15] We presently have only one Sensory Buffer that holds textual information coming from the environment (which is presently limited to the ISS simulator and the user interface), but the architecture can accommodate multiple Sensory Buffers.

type of information; others try to spot the presence of some instantaneous codelet pattern (the co-occurrence of some codelets in WM) or the build-up of a temporal pattern, even over quite distant but related events. *Whereas IDA uses attention codelets as the exclusive means to form coalitions that can enter consciousness*, CTS takes the position that coalitions form without the necessary intervention of attention codelets. Coalitions form on the simple basis of compatibility between information codelets (Baars *convergence process* that fuses whatever is compatible (Baars, 1997, p.52)) and on acquaintance (innate or learned). In CTS (as in IDA), attention codelets look for information in WM (before it becomes conscious); what is proper to CTS is they may serve as a means to create a voluntary bias toward certain information (Baars, 1997, p. 100) *in temporary situations*. They may be involved with early perceptual stages, recreating the phenomenon of that Feldman, Barrett *et al*. describe as "influencing how sensory information is selected, taken in, and processed." (Feldman, Barrett, Tugade and Engle, 2004).

**Expectation codelets** are of the short-lived kind. They are sent by a Behavior to ascertain whether the intended effect(s) did happen after the action has been executed. They keep a vigilant eye directly on WM for the appearance of perception or other information codelets with content that confirm the expected effect. If so, they see no need to bother anyone about the normality of things (Baars, 1997, p.116), and the expectation codelet will only silently send a *reinforcement energy* to the Behavior node that created the effect, confirming its effectiveness and bringing its base-level activation higher. If the expected effect does not show up, the expectation codelet does not wait forever in hope. After a predetermined number of cycles, it puts into WM an information codelet advising of the problem.

**Metacognition codelets** (none designed yet), just like attention codelets, may be looking for patterns, but these are about the processing of the information, about the repetition of unsatisfactory interventions, about trends. They may also be trying to identify patterns of patterns.

When either of these attention codelets is aroused, it spins-out an information codelet, or a coalition of them, that contains words indicative of the situation detected and deposits it in Working Memory.

#### 4.1.1.3 Information codelets

Information codelets are of the short-lived kind, serving only for holding an idea during its transit to and from Working Memory and represent it there until it is published or naturally dies away. The role of information codelets, although very simple, is crucial: they transport information. They are those codelets that progressively form associations leading to new concepts, and they enable the deliberation whereby an idea gets iteratively enriched or inhibited.

#### 4.1.1.4 Encoding codelets, emotion codelets and motor codelets.

There are three kinds of codelets that are not yet designed but for which a role has been conceived. **Encoding codelets** bear similarities with perceptual codelets: they recognize information and feed the mid- or long-term memory they are related to. They prepare the information found in WM to be supplied to Long-Term and Transient Episodic memories. **Emotion codelets** have yet to be elaborated in our architecture but have an important role for an agent that wants to be perceived as *really* intelligent by its human user (Picard, 2000). In fact, IDA has had an emotional mechanism for some time, which is now being redesigned. I would agree with Franklin and Ramamurthy' propositions that they intervene and influence in many places and ways in the cognitive cycle (see Franklin and Ramamurthy, 2006 for more details). **Motor codelets**, members of behavior codelets, will one day serve the purpose of activating bodily parts of the agent, if it ever gets a body.

#### 4.1.1.5 Behavior codelets

In their latent state, they are attached to a Behavior node in the Behavior Net, along with other types of codelets (information codelets, expectation codelets, and some fleeting attention codelets that remain dormant under Behavior nodes). Behavior codelets are those that know how to, and do take action on demand. For instance, one codelet may know how to contact a database to receive information about a recent space mission, or how to contact a jokes service to get some material to present to the astronaut after a long session. Their actions could aim at the internal structure of the agent, to bring modifications to a Behavior node or insert a new one in the BN (after being notified of that need by a metacognition codelet).

### 4.1.2 Coalitions of codelets

Information codelets almost always form coalitions with other compatible codelets. They have an activation value indicative of the importance of the information they bear. When grouped in a coalition, they form a global activation value that will decide whether Attention will descend on them. The coming sections 4.1.3, 4.1.4 and 4.1.5 describe how the activation value of a coalition is obtained.

### 4.1.3 Energy and activation value

I present the concepts of energy and activation separately from other entities, even though they do not exist on their own in CTS. They are found at many places in CTS, within many entities, and play a crucial role in planning, in the organization of the information and in the processing accomplished by the agent. Understanding them is essential for a good comprehension of the way CTS works.

Within CTS, the concept of energy appears in the energy flows, in the activation levels, and in the links' strength. *Energy* represents the signal "strength" (fre-

quency of pulse) that neurons generate from the stimulations they receive by their dendrites and push along their axon to communicate information to other neurons. The activation level (or just “activation”) of neurons comes in part form the accumulation of the energy received recently, and from the stable base-level activation level they acquire with experience. Just as in real neurons and in neural groups as a whole, internal energy (activation level) of various entities of CTS increases with the stimulation coming from internal sources, from codelets representing the environment, or from the passage of time. Nodes in the Perception Network and attention codelets show this phenomenon. Every entity in the BN also do. For instance, when stimulated or when the time has come, a BN's Feeling pushes energy in the Behavior Network to the Goal nodes that connect to it. When satisfied, the Feeling of the need decays rapidly, and it stops feeding the network with energy. The flow of energy within the Behavior Network accomplishes an important part of the planning. The energy that flows from the Feeling nodes indicates the wishes of the agent and which Goal may be relevant to satisfy it; it sustains a *top-down* (goal-driven / proactive) planning. Conversely, the energy that comes from the activated States maintains CTS reactivity to the outside world by sustaining *bottom-up* (reactive) planning.

Energy serves as a common language between multiple entities of various functions and Behavior nodes. The accumulation of energy in Behavior nodes indicates how much a Behavior is appropriate to the global context. However, precise causality is lost, just as happens with intuition, where someone knows what he should do without being able to tell exactly why. "Intuition" is also found in Feeling or Desire nodes, which are stimulated by a variety of stimuli (events); one cannot say precisely what caused the Feeling to become strong.

The Perception Network also makes use of energy levels. When messages received in the Sensory Buffer contain an appropriate chunk of information, a perceptual node activates to its "natural" activation value, or climbs to its "habituated" (diminished) value.

The activation level indicates the importance of the information that an information codelet or a State carries. It may come from the *natural* value of the information

(see Appendix C), from the *contextual* value (as obtained in the BN or in the PN), or the historical value that a link represents. In Working Memory, the activation level of related codelets add-up in some way and support the competition for consciousness. Just as is the case for a single codelet, it is on the basis of its activation level that the importance of the information borne by a coalition is measured. I will say more about the computation of coalition's activation level in the coming section 4.1.5. States in the BN compute their activation from the value of the information they find in the conscious broadcasts (see section 4.1.7 for more on States). So, broadcasts by the Access Consciousness (to be explained in a few moments) also realize an energy transfer.

Everywhere activation exists in CTS, activation decay follows. It implements the general idea that information looses importance with its aging, a hypothesis generally verified in a dynamic environment. It also reflects the fact that our mind (and our whole nervous system, in fact) is a dynamic system that needs let go of some information to avoid becoming clogged by pieces of information that are no longer relevant to the context. Without it, all our senses would eventually become stimulated and remain excited long after the event happened, even years later! In the Perception Network, a stimulated node starts loosing its activation right after being excited by a stimulus. If stimulated again, it reacquires a portion of this activation, not all of it, and even less in the following stimulations. This corresponds to the *habituation* of the senses, a progressive desensitization of the perceptual mechanisms that tends to orient attention towards what changes in the environment. Decaying is also present in the links' strength acquired by experience. Without the forgetting process, the system would keep planning on the basis of relevance that was true long ago but was never seen again.

We use curves similar as those used in LIDA for learning and forgetting. Learning (acquiring base-value activation) rate follows a sigmoid curve, starting a bit slowly in the first few experiences, then growing rapidly in the next occurrences of the event, to eventually saturate and reach a quasi permanent status. Forgetting rate uses the inverse sigmoid curve, for a slow forgetting when the base-activation level

is close to an irreversible value, being faster if the value has not reached the quasi-stable level, and returning very slowly to a final erasure a the lower values of the curve. As decaying is a continuous process, if the experiences are not repeated soon enough, the base-level activation returns to zero (and the link created is eventually forgotten, erased). However, we have chosen to apply a softer slant to the forgetting curve, since we think that it generally takes much longer to forget than to learn (we recall meeting someone days later, even if the event lasted just for a few seconds). If the events happen frequently enough, the base-level grows. Faghihi (2007) gives a more detailed account of these processes.

### 4.1.4 Sensory Buffer (SB) and Perception Network (PN)

The Sensory Buffer serves as an inward interface to any external actor. In its present instantiation, CTS only has the International Space Station simulator and RomanTutor's user interface as an external environment. These sources alone supply a relatively rich information: every dynamic aspect of the "environment" appear in the messages received from the Simulator: Canadarm2 configuration (rotation angle of every joint), position of the payload, camera selected on each of the three monitors along with its dynamic attributes (zoom, pitch and yaw angles), etc. If the event was not manipulation related, other types of information are supplied, such as exer-

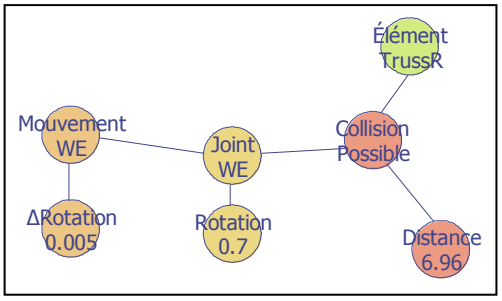



**Figure 8 Portion of the "active" Perception Network.** In this state, the network describes a rotation that brought joint WE to a distance of 6.96 from the Space Station element TrussR, creating a situation of possible collisions. *Source: Hohmeyer (2006).*

cise type and specifications – these aspects remain to be implemented. The nodes (information codelets produced by these nodes, in fact) give semantic meaning to the data through the hierarchical organization of the network (resulting in "concepts" the agent can recognize: “Canadarm2 manipulation”, “user answer”, etc.). They also grant importance on a semantic basis. It must be pointed out that "Sensory Buffer" is a somewhat abusive terminology in the current setup since what the simulator currently sends is not images or sounds but a train of words, which already have semantic meaning as such. But for CTS, these words constitute *data* since they still need to be given a meaning for CTS to understand and use.

*Our Perception Network does not incorporate IDA's slipnet capabilities*. However, the Perception module is an example of our architecture's capability to integrate "alien" mechanisms. Our Perception creates the flexible, weakly coupled bridge between the environment and our internal Perception Network. Patrick Hohmeyer, as part of his Master thesis (Hohmeyer, 2006), designed this mechanism that takes any message coming from the environment and rewrites it so that it can be processed by CTS' internal perceptual codelets. That translation makes use of a formal grammar describing the environment's elements and their semantic relations. A syntactic analyzer that incorporates feature-detecting processes examines the incoming message and reorganizes it into a hierarchical tree. Then, the perceptual codelets can inspect the tree, looking for information they recognize. When a codelet does recognize something, it grabs the information and passes it to the PN information codelet to which it corresponds. For instance, there is a perceptual codelet for each Canadarm2 joint, one for each camera, one for each monitor; there are others corresponding to higher-level concepts such as a request for help made by the user. When a specific joint's perception codelet finds the proper descriptor, it isolates the information about the joint's angle and starts its transfer process. It compares it to the information previously held by the information codelet representing the joint; if identical, it does only a partial infusion of energy.

That particular aspect, the *energy transfer*, is directly related to a fundamental hypothesis in CTS' architecture: information has a value. It corresponds to Baars

affirmation that some things are more important than others, and that high-priority stimuli, like the sound of one's own name, are even detected unconsciously (Baars, 1997, p.158). Perception is the first place (the first step in the cognitive cycle) where information value is implemented. This will later allow CTS to put its attention on what is most important, *on which information has more value in the current context*, and put its costly conscious resources at work on what deserves it most. This, as we will see again later, is part of the prioritizing function of consciousness (the phenomenon called *monitoring consciousness*). Although there is still much work to be done in this area, we have made some preliminary hypothesis about the relative importance of information (see Appendix C) and elaborated a short set of heuristics. The heuristics that apply to stimuli valuation are:

1. The information type dictates a first part of the information value. For instance, an *environmental consequence* (collisions risk or effective collision) has more importance than a *joint rotation*. Canadarm2 coming into close proximity of the Station can be generalized as a "proximity" situation. If situations of the type "proximity" happen often, the tutor may feel the need to intervene to correct this problem. So, the type of the situation is sufficient to draw the tutor's attention and has a value on its own.
2. A piece of information that changed (a variable that changed its value) is more important than one that remains unchanged. For instance, there is usually less danger with something standing still than with a moving object. This principle is related to *sensory habituation*. This rule holds true unless it refers to a situation of repetition (insistence) watched for by an attention codelet.
3. An improving situation is of less immediate consequence than a deteriorating one. For instance, a coalition of information indicating a joint getting closer to a Space Station module is of greater immediate importance than one indicating a joint moving away from the Station.

Codelets that have some activation (just received, or remaining from previous stimulations; see an example in Figure 8) form the perception (sub-)network that joins the *Scene* (in the theater metaphor), that is, appears in Working Memory.

At this point, the Coalition Manager will identify the various possible coalitions from the percept and compute each one's value, allowing Attention to find the most activated coalition (the most important one in WM). I will give more detail about coalitions and the valuation process in the next section (4.1.5).

### 4.1.5 Working Memory (WM) and the creation of coalitions

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). That "fleeting memory capacity" is more commonly seen as a working memory that is central to many processes. As recalled by Franklin (Franklin, 2006) from Baddeley and Hitch (1974), *working memory* is not a biological structure on its own, but a cognitive psychology term referring to a theoretical framework specifying and describing multiple structures and processes used for temporarily storing and manipulating information. However, in accordance with one of GW theory's assumptions asserting that a global workspace can also serve to integrate many competing and cooperating input networks (Baars and Franklin, 2003), it is interpreted in our architecture as a single, unconstrained "place" where all codelets meet when needing to be "published". It corresponds to the stage in the theater metaphor of Baars' theory. It is where associations are created and where these associations get stronger between codelets that spend time together. Working Memory is where coalitions are sent by all modules, where they combine, get enriched or opposed. This is where all attention codelets (the group in that specific name) look for information. The Attention mechanism, corresponding to the theater's spotlight, constantly monitors it, selecting at every cognitive cycle the next winner to come to consciousness. One could call it the *Central* Working Memory, as there also seems to exist in the brain many local

working memories for the use of each specialized neuronal group (Baars, 1997, p.41).

There are two differences of our WM with IDA's. *First, IDA's Working Memory is analogous to the preconscious buffers of human working memory* (D'Mello et al., 2006) *and limited to them. Second, IDA's WM is constrained by the fixed structure of the preconscious buffers that implement it* (Franklin, 2003b, p.5) – *LIDA has seen this constraint relieved with its workspace that keeps the still constrained preconscious buffers but otherwise now allows the building of unconstrained structures over multiple cycles.* In comparison, CTS uses as WM a structure that is not constrained by fixed registers or depth limitations, that is highly dynamic and that is separate from preconscious buffers (read sub-section 4.1.11 for some more explanations). "Highly dynamic" is used here in the sense that it allows the formation of links between codelets and coalitions that temporarily inhabit it, whereas IDA's preconscious buffers simply serve as receptacles for information.

When our team will put a declarative memory in place, I plan to add *encoding codelets* that will create the bi-directional communication between the preconscious buffers feeding the long-term declarative memories (called the *Focus* by Kanerva) and Working Memory. The *Focus' cue vector* will be built with information either found in WM by *encoding codelets* or placed there directly from the Perception Network.

Perception is not the only place where information valuation takes place. Things that slip into consciousness are not all issued by the perceptual process; they are as much a matter of internally-generated material: concepts recalled from memories, remembered episodes, preferences, emotional state, personality, and lessons learned (experience, currently exclusively stored in CTS links' strength). They may also result from reflection, where CTS analyses, links ideas together, compares alternatives and makes decisions. Even conceptual (*vs.* concrete) ideas have value to humans, and thus are able to compete with coalitions coming from Perception.

CTS is endowed with explicit deliberation capability thanks to its consciousness mechanism. As a result, just as codelets create coalitions, coalitions may form

bigger coalitions of coalitions, which I call *complex* coalitions. A complex coalition progressively forms during deliberations, adding new chunks of information (codelets or coalition of them) in subsequent steps until the deliberation is considered complete by the Arbiter codelet. Figure 9 shows an example of complex coalition comprising three coalitions.

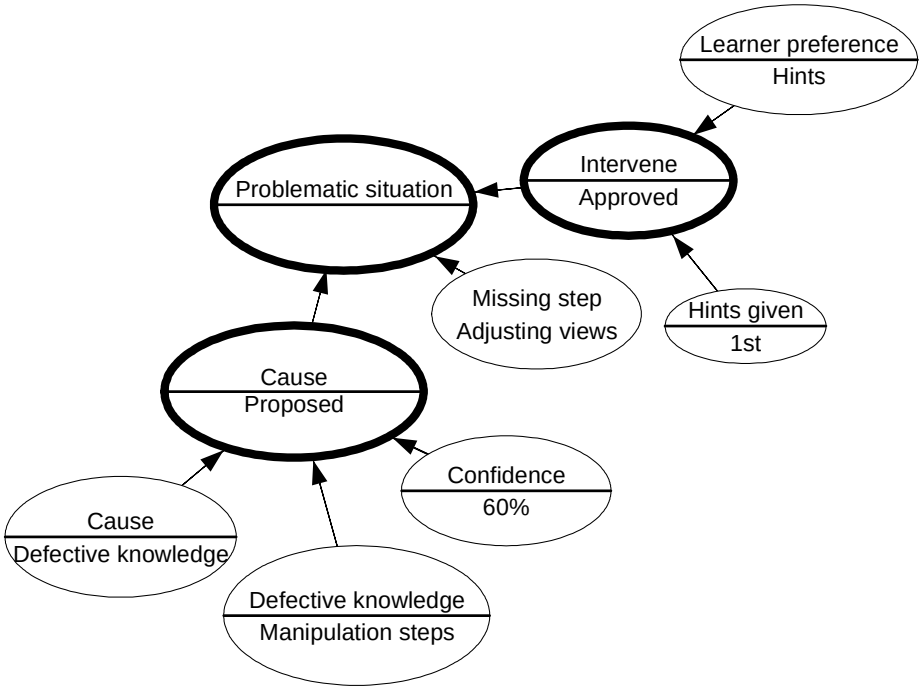



**Figure 9 Example of a complex coalition** as it has grown to during an ongoing deliberation. Nodes with a bold contour are central nodes of a coalition that appeared in WM as a direct response to a conscious publication, at some part of the deliberation.

The complex refers to a problematic situation noticed by the Domain Expert module: a procedural step that has been omitted by the astronaut. Before impelling any new motion to Canadarm2, he was supposed to create the appropriate views on the three monitors of the workstation (see Figure 1), selecting the right camera on each and adjusting the orientation and zoom. The combination of the three views has to be the most informative possible. If the astronaut starts moving Canadarm2

before adjusting the views, this is a procedural mistake, and the Domain Expert will spot it. In subsequent cognitive cycles, relevant resources (most likely modules, here) will then supply probable causes for this problem.

The main principles guiding the valuation process in WM, in addition to those mentioned about Perception, are the following:

1. In addition to the information type, specific aspects and specific values may also bear some importance. For instance, a *weak* understanding of a concept may be more important to take into account than an *average* understanding, which has less probability of incurring grave consequences. If the aspect is a *belief* of CTS, than it is also subject to a *confidence* level, which modulates the importance of the information. Another example of the value of specifics can be given with the cause of a problematic situation. Some causes of a problem are more serious than others and should be prioritized.
2. If an information coalition gets richer through deliberation, CTS should *usually* bear more consideration to it than to an idea just arrived in WM (although the specific evaluation depends on the intrinsic value of the information just arrived). The more the global resulting coalition comprises central nodes, indicative of more deliberation steps, the more the coalition has value.

Of course, it would be ideal if the intrinsic value of the information evolved following CTS experiences and observations. It will be part of future works.

*The valuation process differs from IDA's, where the coalition takes on the average value of all the codelets that form it.*

Here is the algorithm deriving from these principles that computes the coalitions values.

1. If a coalition has two or more levels, consider only the first two for the valuation process.
2. Compute the average value of information codelets in the second level.

3. Add this average value to the value of the central (main) node (that indicates the type of the info)

For complex coalitions:

4. Add the values of the central nodes to that of the "first" central node.

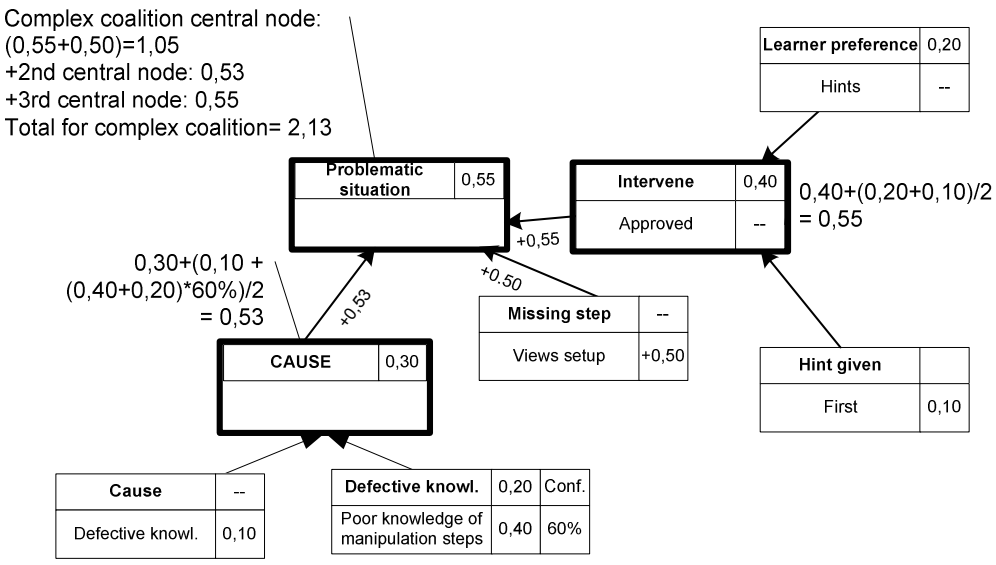



**Figure 10 Example of a complex coalition grown by a deliberation process.** Boxes with a bold contour indicate central nodes that represent their coalition in terms of activation level. The first central node, the one that caused others to come to consciousness from various sources takes on the supplemental role of "central" central node, that is, the central node for the complex coalition, the one that is considered in the competition for consciousness.

What is implicit in this algorithm is that adding or suppressing a link (a codelet) ultimately changes the global value of the whole complex through the variation of the value of the central node to which it connects (or was connected). More than one coalition may be formed from the same percept, each having its own value that depends on the arrangement of the codelets. Figure 10 shows how the complex coali-

tion's value is computed from the value of each of its member coalitions. The first step of the algorithm is a consequence of this practical consideration. We must limit coalitions evaluation to the first two levels of a coalition: its central node and the nodes directly connected to it. This is of particular importance for information coming from Perception, as our Perception Network is a cyclical graph (it contains loops) and at some point, it becomes impossible to isolate the various possible coalitions in the same network to compute the value of each. Indeed, in this cyclical graph, coalitions are interdependent. By limiting the number of levels considered, we render the computation possible while preserving the richness of the information: the coalition that is published contains all of the attached information, not just the first two levels.

Another mechanism affects coalitions' values: activation decays. Just as is happening in the PN nodes, in preconscious buffers, in LTM, in BN nodes and links, the activation of an entity decays with the passage of time (D'Mello *et al*., 2006; Franklin, 2005b). As mentioned in the section on Energy and activation value (4.1.3), we have adopted the same inverse log curve as D'Mello for this phenomenon.

A coalition formed in a previous cycle may compete in the next coming cycles, but not forever, as it will eventually decay away.

### 4.1.6 Access Consciousness

At the center (graphically and conceptually) of the architecture, we find the *access "consciousness"* which "publishes" the information selected by the Attention mechanism to make it available to all (unconscious) modules (by a "broadcast"). It implements the still debated mechanism that effectively binds various regions of the brain together and propagates their information content, allowing all other systems to become aware of the situation. This mechanism is crucial for the collaboration of the parts, for instance in reaching a diagnosis.

Selecting information that will be broadcast establishes an important difference with the Blackboard model. A blackboard broadcasts all the information it contains to

every resource, whereas the LIDA/CTS architecture selects which information is the most important one in the context, and only that one is broadcast. This helps steer the global behavior of the agent, and it avoids that many modules work on information of lesser importance, or globally insignificant.

The apparatus for producing “consciousness” consists of a Coalition Manager, a Spotlight Controller, and a Broadcast Manager. *LIDA adds the attention codelets as part of the necessary elements for information to come to consciousness* (Ramamurthy *et al*., 2006). In CTS, attention codelets are not required for a coalition to be chosen and reach consciousness. Their purpose is to watch for the appearance of some situation, sometimes to insure fast reaction in situations akin to an alert, or to offer the means for attentional bias (adding activation to some coalition).

The Coalition Manager is responsible for forming and tracking coalitions of codelets. Such coalitions are initiated on the basis of the mutual associations between the member codelets; in CTS, these associations come either from their neighboring in a percept, or from spontaneous association in WM (see sub-section 4.3.5.1 on this last subject). During any given cognitive cycle, one of these coalitions finds its way to “consciousness,” chosen by the Spotlight Controller (the "Attention"), which picks the coalition with the highest global (*average, in LIDA*) activation among its member codelets (see section 4.1.5 above about the computation of coalitions activation). Global Workspace theory calls for the contents of “consciousness” to be broadcast to every codelet in the system. The Broadcast Manager accomplishes this.

### 4.1.7 The Behavior Network (BN)

This structure is at the same time a planning mechanism, a decision structure, and a long-term procedural memory. The planning and the decisions it makes are taken "unconsciously" but rely on consciousness' broadcasts to keep abreast of the evolution of situations. This is what is called "consciously mediated action selection". The actions it provokes eventually become conscious, either when they justify a publication, either indirectly in their perceived effects (information received from the out-

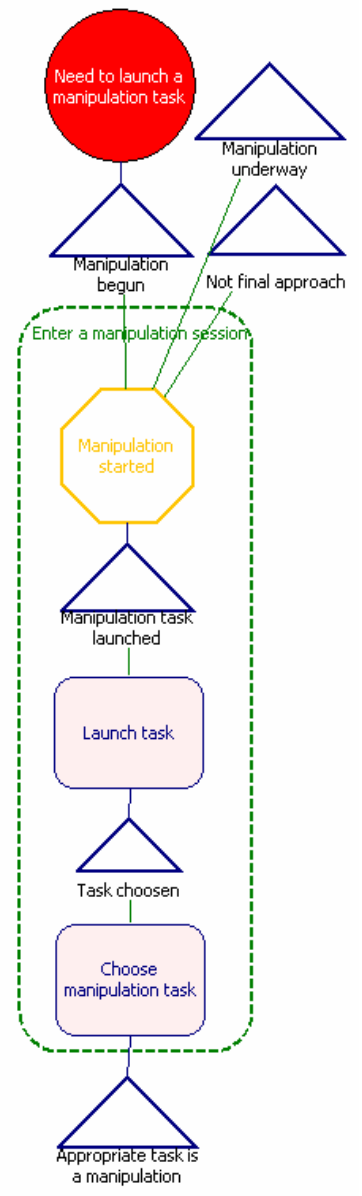



**Figure 11 Example of a simple structure in the BN**. Filled red circles are "Feelings" or "Desires", octagons are sub-goals, triangles indicate States, and rounded rectangles contain Behaviors. A more complex stream could include branching and multiple preconditions and effects.

side) or by a message from metacognition codelets about them. Its learning is both unconscious and conscious (*conscious learning already exists in LIDA*, but is part of future works on CTS).

Based on an idea from Maes (1989) and modified by Negatu and Franklin (2002), the Behavior Network holds the repertoire of the agent's know-how in the form of ***streams*** of Behavior nodes. *LIDA has separated the procedural memory from the decision mechanism, adding the Scheme Net to hold the dormant templates of schemes.* Streams are organized under *motivators* and sub-goal nodes (see Figure 11). Examples of motivators (or, in terms of CTS structures, ***Feelings* and *Desires***) are *Need to launch a manipulation task*, *Need to intervene*, *Need to remediate*, and *Need to give affective support*. The network architecture offers a way to decide on which Behavior should activate according to "Feelings'" and "Desires'" activation (see next point), to States (shown as triangles in the BN Editor), to links between nodes, and to various thresholds. In its logical aspect, the BN does "HTN decomposition and plan-

ning" (HTN stands for "Hierarchical Task Network"), as described by Russell and Norvig 2003). Each **Behavior node** has necessary preconditions and indicates effects it should have on the environment, creating a natural link with those nodes that have these effects as preconditions. Preconditions and effects exist as ***State nodes*** connecting Behavior nodes together through **links**. For every top-down link, there is a reciprocal bottom-up one. There also are inhibitory links that project from a Behavior toward other Behaviors that would undo its preconditions. When a node receives some stimulation (in the form of **energy**), it pushes some of it towards inhibiting these contradictory Behaviors. More generally speaking, at every cycle, some energy is replicated by all active energy sources to their neighbors. A Behavior node accumulates the energy that comes "atop" from the agent's "Feelings" and "Desires" (*feelings*, *or drives, in IDA*), from *States* and other nodes until it is elected for action. The direction of the energy flow is *forward* (from the Behavior node towards a Goal node) only when that node *fires*, in which case it pushes its energy to its temporal successor(s). Otherwise, the flow follows the *downward* (top-down) links, splitting at branches, eventually reaching a "starting" node, a node at the temporal beginning of a sequence. One main idea that this arrangement supports is that shorter solutions are better; indeed, shorter plans should usually be tried first because there is a lower probability of making a mistake (with less steps and less understanding required) and because a good approximation is often sufficient. This heuristic is particularly true in survival situations. Automated responses ("reflexes") are of the same nature, on this account, and will normally be used even before any non-reflex plan, unless they are consciously blocked. They come to exist in the BN when experience reinforces links and Behaviors in a stream so much that they transform the stream's (or the sub-stream's) reaction, making it react much sooner and unfold faster. We will come back later to these subjects, in the section presenting the implicit learning mechanisms in CTS (section 4.3.5.3).

Franklin and his team modified Maes model so that each Behavior is realized by a collection of codelets (Behaviors nodes do not act by themselves; they only serve in the planning of the next behavior). *Behavior codelets in IDA are always active, listening to broadcasts; when they find themselves relevant, they have their*

*stream template get instantiated to become part of the Behavior Net (the actually instantiated streams), and they send activation to their respective node.* In CTS as in IDA, when a Behavior node gets elected, its codelets are released with the energy level of their node and start doing what they are meant to do, some putting information in WM, some requesting information, some starting to watch what will happen in the next few cycles to confirm that the expected effect(s) happens, *etc*.

*States* in CTS have multiple roles. As preconditions, they control the Behavior nodes selection, since all preconditions must be present before the Behavior may fire. They are meant to represent neuronal groups that have been stimulated by previous interactions (by previous information received), and that keep that information alive for a certain time (their activation decays with time). In that respect, they play much of the same roles as IDA's behavior codelets in their listening mode. They recognize words in broadcasts, store them in variables, transfer their content to the codelets of the Behaviors they are connected to, along with environmental activation. *Differences lie in that our States do not connect to a procedural memory; they do not request for the instantiation of the relevant stream: there is no instantiation of the "active BN" in CTS*. *Whereas IDA separates the (active) Behavior Network from the scheme memory*, *CTS keeps its whole BN "on-line"* on the idea that in our brain, neurons are all active at the same time (Baars, 1997, p.55), even if they do not participate in the current intervention. This idea is particularly of relevance in the planning phase. Franklin's idea of instantiation stands its grounds on the basis that all our action plans are not active all the time, only those that are of relevance in the precise context and get recruited.

At each cognitive cycle, a Behavior is selected for action. If, however, none has reached the necessary energy threshold, the value of the threshold is lowered by the BN Manager for the next cycle. The higher the threshold value is set, the longer the BN will be planning before a node can fire, exploring longer paths (plans) and allowing for more interactions between nodes to influence each other. This makes for a more "prudent" and "analytical" agent.

### 4.1.8 "Feelings" and "Desires"

*Feelings* and *Desires* in CTS' Behavior Network are special high-level Goal nodes playing the role of behavior motivators. Much can be said about motivators, feelings, drives and the likes, and the concepts are still quite debated. «Motivation, drive, goal and emotion are used to refer to and mean a number of different things. There is no universal definition of these terms across (or even within) the fields of philosophy, psychology, cognitive science and artificial intelligence» (Davis, 2002). In an undocumented writing (on the website of University of Geneva), Gagné cites Good and Brophy (1990) for their understanding of motivation and attitude: «Motivation is whatever initiates, sustains or causes a direction or intensity toward a particular behavior. In contrast, an attitude, as discussed above, is a predisposition to choose one behavior over another». In a perfect illustration of the fact that there exists various comprehensions of these concepts, Davis proposes to define motivations rather as dispositions to assess situations in certain ways. In his view, they can include goals and desires as well as attitudes. Goals would be quantitative or qualitative, the latter mostly being used in agents as involving relations, predicates, states and behaviors. Attitudes are predispositions to respond or act. Ideas from Good and Brophy, and from Davis are found in CTS' motivators as specific sensitivity of Feelings and Desires to events, and preeminence of some over others.

Franklin and Ramamurthy (2006) also have reviewed the subjects of motivations, values and emotions. Feelings in human include hunger, thirst, pain, being hot or cold, the urge to urinate, tiredness, depression, *etc*. "One feels feelings in the body." Feelings refer to the basic needs of a person; they result from his fundamental biological processes. Referring to Johnston (1999), the authors separate emotions from feelings. Fear, anger, joy, sadness, shame, embarrassment, resentment, guilt, etc. are higher-level feelings, that is, with cognitive content. One "feels good" after meeting one's objectives; a tutor might feel shame if it cannot bring a student to complete an exercise. But one is simply hungry, tired, or cold (intransitive words). Emotions are relational; they come from an interaction with the environment. So, according to Franklin and McCauley, feelings are the motivators that refer to homeo-

static drives (the motivators that refer exclusively to the subject and help regulate his internal, basic, physiological needs), and emotions, motivators that result from external events. «Feelings, including emotions, are nature's means of implementing motivations for actions in humans and other animals. They have evolved so as to adapt us to regularities in our environments». They are the two mechanisms that implement general preferences, often called "values". IDA implements *drives* as *feelings nodes* in a way similar to CTS, at the "top" of its BN; *but LIDA has removed feelings as "top-level" sources of activation in its Behavior Net (now called Scheme net) and changed the way "feelings" and "emotions" intervene – they bring their direct influence elsewhere in the architecture and still modulate action selection. Feelings and emotions act at the perceptual level as semantic nodes of the Slipnet to help determine the content of the percept. Feelings and emotions are also found in episodic memories, as part of episodes content; when recalled into workspace, they influence information structures by bringing activation into them and help determine the selection of conscious content. When broadcast, these coalitions transfer not only information, but also the activation content borne by the emotional codelets, to instantiating schemes.*

In CTS, I adopt Franklin's conceptual views about feelings and emotions, to which I add a clear role for the idea of *attitude* (predisposition to react in a certain way). CTS' *Feelings* sense broadcasts by the way of their word-specific sensors (I was tempted to write their "dendrites"), and the sensitivity of each sensor can be adjusted (by the network designer). As a result, each Feeling can react more or less to the various aspects of an event, depending on the personality profile selected for the agent. However, since I have not yet designed emotional mechanism, both feelings and emotions are incorporated in CTS as the generic mechanism of Feelings; thus, CTS' "Feelings" play a role only in the Behavior Network. This being said, I add another notion in the BN, *Desires*, as high-level motivators of a psychological nature but that play the same role on action selection. Desires will never have an impact over learning. So, in CTS, *Feelings* (including Franklin's notion of "emotion") and *Desires* behave the same way; the distinction is currently only for the benefit of the agent's designer (although, in time, the separation may allow separate processing

methods). "Feelings" are meant primarily to react to the external environment but might include preoccupations for internal needs the agent might have (for instance, the need to terminate a session so that it can organize its data, or the need to make deep analyses, as indicated by serious users problems to which the agent presently finds no cause). "Desires" cover mostly the agent professional goals (as a teacher).

Upon announcement of situations (consciousness broadcasts), a Feeling (or Desire) node elevates its activation level according to the importance of that information (or a portion of it), according to the sensitivity it has for it, and according to how many of its sensors have been stimulated. The Feeling starts feeding energy to BN Goal nodes attached to it, or increases its previous output. But its activation decreases progressively, following the inverse logarithmic curve.

A last word about CTS' Feelings and Desires: they have a correspondence to high-level, global desires in BDI parlance. In effect, they accomplish high-level planning. The backward flow of energy from the Feelings and Desires nodes is a form of implicit planning influenced by the goals.

### 4.1.9 The Learner Model

The learner model is composed of three separate mechanisms that run in parallel. The Learner Profile Model (LPM) contains stable psychological indications about the learner, including learner’s learning style and preferences; the Learner Affective Status Model (LASM) tracks learner’s mood and affective state. The Learner Knowledge Model (LKM) holds facts, infers knowledge and trends, and computes statistics. All three (sub-)modules receive the consciousness publications to stay informed about the learner. They send information to WM when they receive a request or whenever they deem appropriate. The LPM has a light inference engine and reacts when it recognizes a situation needing a first-level (superficial) diagnosis, that is, a probable immediate cause for the problematic situation. The LKM will also do this from its standpoint, but it will also volunteer information when its inferences show a significant problem with learner's knowledge, or when a trend about learner's

performances should be signaled, eventually priming some "*Feeling"* in the agent. At the time of this writing, all three modules of the Learner Model (LPM, LASM and LKM) are temporarily faked with attention codelets and will very likely be replaced with Bayesian networks.

### 4.1.10 The Domain Expert (DE)

The Domain Expert is a good example of an external module (or an agent) getting integrated into our conscious agent framework.

In a single three-layered entity, Fournier-Viger's dynamic model contains the capacities to reason about a specific domain, a semantic memory that is used as CTS' own semantic memory (see Fournier-Viger *et al.*, 2006, for a complete description), and a procedural memory about the domain. Together, the semantic and procedural structures encode CTS' expertise about the domain of application. For example, the semantic part of the memory indicates what camera can see what structures of the International Space Station (ISS); the procedural part describes the required steps before moving Canadarm2. CTS could use those procedures to accomplish operations in the simulator and demonstrate how to proceed.

The Domain Expert's first (bottom) layer contains descriptive knowledge, that is, the physical description of the (simulated) world (the elements and the relations between them) and the concepts that refer to that world, organized in an ontology. This allows logical reasoning. Applied to our case, it describes the ISS' structures (the modules, their role(s), their physical characteristics and their relations, such as «is above», «is below», «is to the left of», etc.), the elements of Canadarm2. It also contains abstract concepts such as distance, coordinate systems and collision risk. The formalism used is description logics (Baader, 2003).

The second layer describes the relevant correct and erroneous knowledge that students may manipulate while utilizing the learning environment. Knowledge is encoded from a cognitive perspective as semantic and procedural memory with struc-

tures having their roots in ACT-R (Anderson, 1993, 2004) and Miace (Mayers *et al*., 2001) theories. Layer 2 links the semantic memory description to the layer 1 ontology. Whereas, layer 1 describes concepts as part of an ontology, layer 2 describes concepts using attributes that describes their cognitive use (for instance the intentions that a student can have with a concept). Procedural knowledge describes the means to manipulate semantic knowledge to realize intentions. The second layer also adds domain specific didactic knowledge.

The third layer defines learning objectives and organizes the two other layers under these, packaging them into reusable learning units.

The Domain Expert provides information in three situations: a) if it “hears” (or “sees”…) a request for information it can supply; b) if it recognizes a situation that needs a first-level diagnosis; c) if it believes the user errs about a procedure (missing step, reversed steps, wrong procedure). It is informed of what the user is doing by the consciousness' publications, and tries to recognize the plan and steps followed by the user (doing *model tracing* against the procedures stored in its domain model). Then, as is the case with the Learner Model, when it believes it identified important information to communicate, it sends a coalition of codelets to WM describing the fact.

In the present instance of our prototype, we have incorporated in the Domain Expert four "levels" of hinting: 1) a general clue, 2) a more specific clue, 3) the fact observed, and 4) the suggested course of action. They are made available when an intervention is needed to correct a problematic situation. The hints may be given in this order, or can be supplied by the Domain Expert specifically, upon explicit requests from a tutoring Behavior. There could be more complex algorithms involved here, with more material available (for instance, many hints of the same intervention level per situation, each adopting a different point of view), more involved BN streams, a variety of tones and styles, and so on, but we kept it simple in our prototype, with a simple sequence of levels bearing only one hint per level.

### 4.1.11 Long-term memories

There is a variety of long-term memories in CTS' conceptual architecture, corresponding to what is generally believed about human memories (Baars and Franklin, 2003; see Figure 12): the Transient Episodic Memory, the Autobiographical Memory, and the Semantic Memory. «The **Transient Episodic Memory** (TEM) corresponds to humans' content-addressable, associative, transient episodic memory with a decay rate measured in hours.» (Conway 2001, cited by Baars and Franklin, 2003). IDA implements the hypothesis that a conscious event is stored in transient episodic memory by a broadcast from the global workspace. A corollary to this hypothesis says that conscious contents can only be encoded (consolidated) in long-term declarative memory via transient episodic memory (Franklin *et al.*, 2005). The **Autobiographical Memory** (AM) integrates events that have not decayed away in TEM when the consolidation takes place. This transfer process remains to be specified in our architecture. AM is currently only part of CTS' conceptual architecture. The **Semantic Memory** (SM) is held in the Domain Expert.

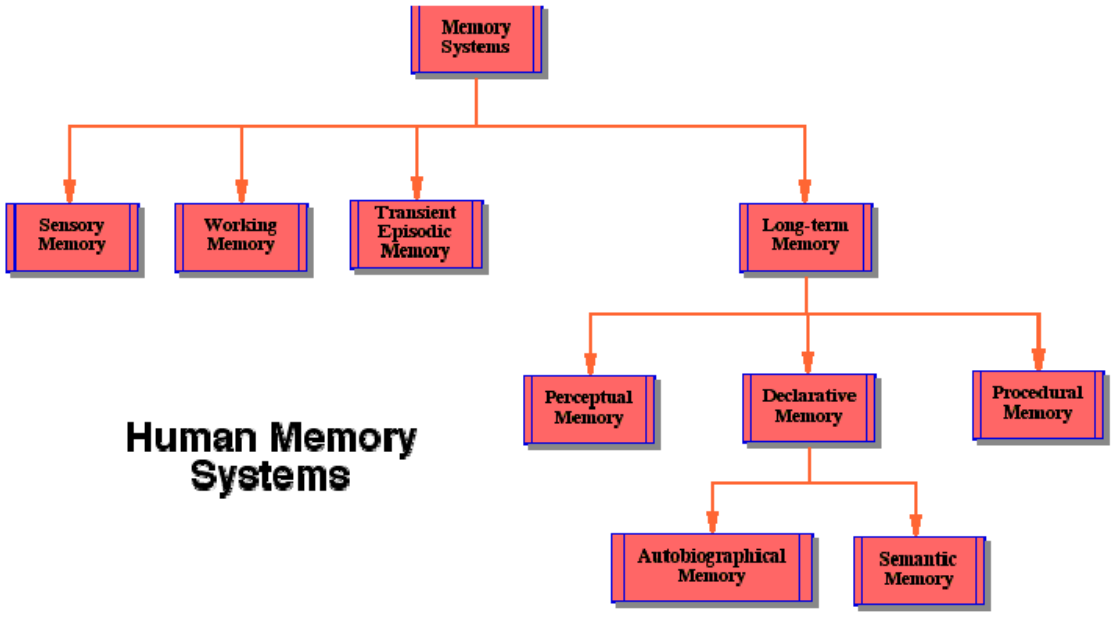



**Figure 12 The human memory.** *Source: (Franklin* et al*., 2005)*

Just like other modules in the architecture, memory mechanisms receive the broadcasts from Access Consciousness. This is what they store (integrate with their anterior knowledge, in other words, what they learn). Learning only from the broadcasts is congruent with the hypothesis that we learn explicit concepts only from what comes into consciousness (Baars, 1997, p.60). It allows storing only structured knowledge, not bits and pieces of unorganized data, as is the case with the *cues* that trigger a *read* operation. The Consciousness content, structured and meaningful, differs from what is put in the unconscious buffers by the encoding codelets. The content of Working Memory is constantly looked up by a great number of encoding codelets that hover it and do exactly the same kind of job perceptual codelets do. When they find information of the type they are concerned with, they put it into their corresponding *slots* in the *cue vectors* that will be submitted to the various long-term declarative memories. The cues assembled in the input vectors by these codelets may show a pretty nice jumble. If it makes no sense, the "read operation" will not converge, and declarative memories will not be able to return anything from this cue. Aside from completed words (what was submitted may have been partial words), these explicit memories return associated information. The *vectors* are intermediary constructs between WM and long-term declarative memories ("declarative", as opposed to "implicit").

I need to make a digression here to present the general idea of a special algorithm we are currently reimplementing at our lab for our TEM: Kanerva's (1988, 1993) *Sparse Distributed Memory*. Franklin and his team have been experimenting with this algorithm ever since 1995 (Franklin, 1995) to implement a transient episodic memory and an autobiographical memory. They improved the original specifications to obtain better results when retrieving the information associated with less complete cues (Ramamurthy, D'Mello and Franklin, 2004; D'Mello, Ramamurthy, and Franklin, 2006). CTS job of tutoring involves a rich enough domain, making difficult to fit the elements of information in the constrained vectors as described both in Kanerva's theory and in IDA's implementation. We are exploring avenues of solution: preclassification of the information; recoding it into shorter character strings; holographic distribution of the information over the whole vector (Kanerva, 1997). That challenge

is actually under consideration by a colleague in his Master research. The SDM algorithm will be part of the next iteration of our prototype and cannot be covered here. So, I borrow a concise description of the SDM by Baars, Ramamurthy and Franklin (2006), that conveys the basic ideas that our SDM will use:

> SDM is a content addressable memory that, in many ways, is an ideal computational mechanism for use as a long-term associative memory (Kanerva, 1988). Content addressable means that items in memory can be retrieved by using part of their contents as a cue, rather than having to know the item's address in memory.
>
> The inner workings of SDM rely on large binary spaces, that is, spaces of vectors containing only zeros and ones, called bits. These binary vectors, called words, serve as both the addresses and the contents of the memory. The dimension of the space determines the richness of each word. These spaces are typically far too large to implement in any conceivable computer. Approximating the space uniformly with some manageable number of actually implemented, hard locations surmounts this difficulty. The number of such hard locations determines the carrying capacity of the memory. Features are represented as one or more bits. Groups of features are concatenated to form a word. When writing a word to memory, a copy of the word is placed in all close enough hard locations. When reading a word, a close enough cue would reach all close enough hard locations and get some sort of aggregate or average out of them. As mentioned above, reading is not always successful. Depending on the cue and the previously written information, among other factors, convergence or divergence during a reading operation may occur. If convergence occurs, the pooled word will be the closest match (with abstraction) of the input reading cue. On the other hand, when divergence occurs, there is no relation, in general, between the input cue and what is retrieved from memory.
>
> SDM is much like human long-term declarative memory. A human often knows what he or she does or does not know. If asked for a telephone number you have once known, you may search for it. When asked for one you have never known, an immediate "I don't know" response ensues. SDM makes such decisions based on the speed of initial convergence. The reading of memory in SDM is an iterative process. The cue is used as an address. The content at that address is read as a second cue, and so on, until convergence, that is, until subsequent contents look alike. If it does not quickly converge, an "I don't know" is the response. The "on the tip of my tongue phenomenon" corresponds to the cue having content just at the threshold of convergence. Yet another similarity is the power of rehearsal, during which an item would be written many times and, at each of these, to a thousand locations—that is the distributed part of sparse distributed memory. A well-rehearsed item can be retrieved with smaller cues. Another similarity is interference, which would tend to increase over time as a result of other similar writes to memory.

As a final word about memories, I would like to point out that there exists more memory structures in CTS than just TEM and AM. The Behavior Network is a procedural memory; the Perceptual Network, the Domain Model, and the event log also are memories. Learning mechanisms just as well create *implicit* memories in the form of links strength in the PN and in the BN, and in learned associations between codelets in WM. I will describe these implicit learning mechanisms in the subsection 4.3.5 below.

## 4.2 THE COGNITIVE CYCLE

Our agent's internal operations follow a continuous stream of interactions quite close to IDA's *cognitive cycle* (Baars and Franklin, 2003). This cycle offers an hypothesis by Baars and Franklin about human cognition, and is much more detailed than any provided in other agents. With convergence, divergence, competition and collaboration taking place in Working Memory, one may come to think of IDA's cognitive cycle as similar to Edelman's reentrant signaling[16]. It is generally considered to be starting with a perception and ending with an action taken.

It may not lead to an external action but to an action having internal repercussions. This cycle organizes CTS internal interactions and preserves consciousness seriality by putting conscious broadcasts as an explicit step in cognition. Human consciousness is formed from a continuous flow, an uninterrupted succession of

[16] Reentry is a dynamical and ongoing process that makes neuronal groups, and maps between them, exchange stimuli (excitatory and inhibitory) until a stable pattern emerges and is strong enough to come to consciousness. IDA's cognitive cycle includes a broadcasting step that brings unconscious resources to respond to other's stimuli. However, for broadcasting to happen, a pattern already has to be stable in Working Memory. Thus "signaling" takes place only at a higher level of organization, after stabilization, between recognized patterns. Edelman's allows for such signaling between high-level structures, but also explains the formation of low-level patterns, before they eventually reach working memory and can be "broadcast" (in Baars terms).

episodes (Crick and Koch, 2003) to which we give meaning and then broadcast throughout our brain. Accordingly, we are constantly sensing; perceiving new stimuli does not wait that the previous perceptions be completely processed. Thus many cycles overlap in a cascade fashion, as the steps of a cycle solicit different cognitive abilities of our brain. In fact, according to Baars and Franklin (2003), «We conjecture that a full cognitive cycle might take a minimum of 200 ms. But because of overlapping and automaticity, which shortens the cycle (see below), as many as twenty cycles could be running per second.» (p.3).

IDA chops the cognitive cycle in nine steps distributed in the the three parts *perceive, interpret, act*. *However, because CTS' functioning differs from IDA's in the instantiation part, CTS' cycle holds only eight operations*. I describe hereunder these steps, drawing much of the description from Baars *et al.* (in press) and Franklin (2005), again pointing out differences under *italic text*. Step number corresponds to numbers inside triangles in Figure 13.

1) ***Perception***. External sensory stimuli are received and interpreted by Perception, creating meaning. Note that this stage, as others before step 5, is unconscious. *In addition to external stimuli, IDA also re-interprets internal stimuli through its perceptual apparatus*.

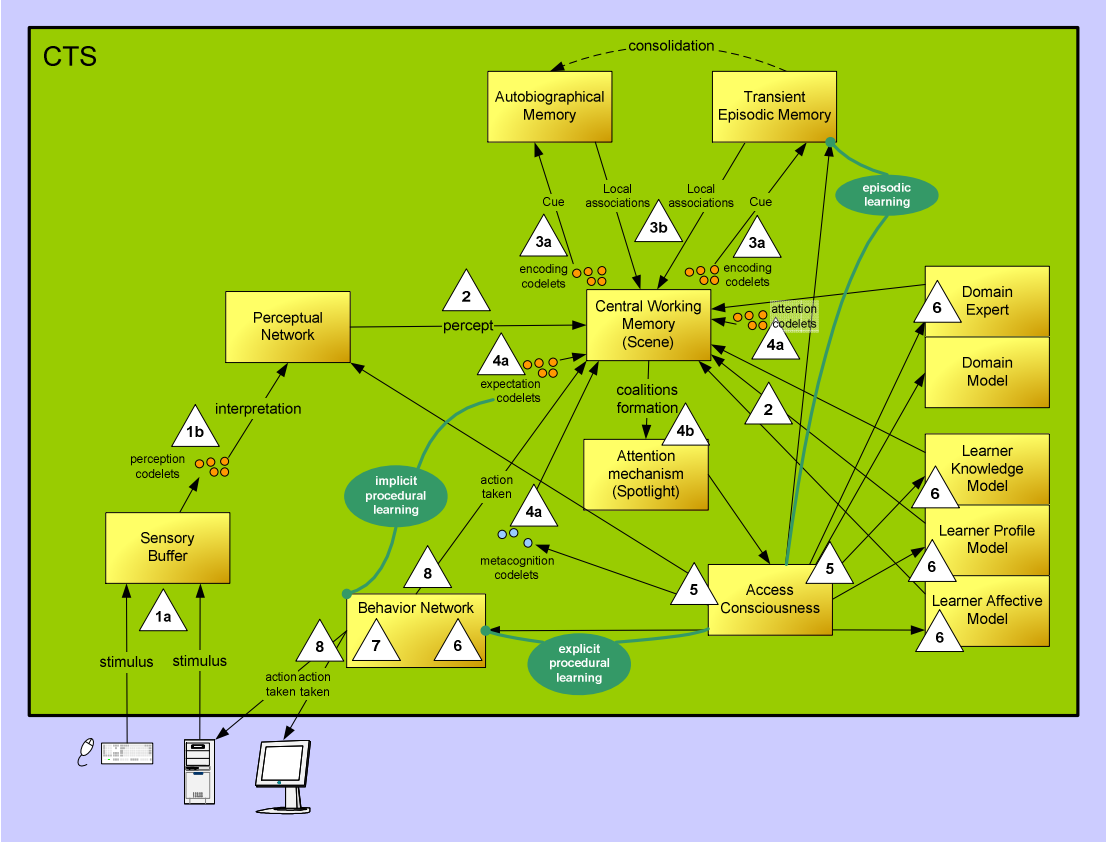



**Figure 13 CTS' cognitive cycle.** Although inherited from IDA, CTS' cognitive cycle presents originalities due to implementation differences (new modules, different implementations, different hypotheses). For instance, there is no instanciation in the Resource recruiting (6th) step.

a. Early perception: Input arrives through senses; for CTS, senses exist as communication channels with external entities (currently, CTS offers only one channel dedicated to communications with the ISS simulator and its user interface). The actual single channel funnels textual inputs to the Sensory Buffer, whereas *IDA directly looks-up the content of the e-mails it receives*. Features detectors (that are part of the syntactic analyzer) find relevant words in the string and collectively create a syntactic tree.

b. Active Perception Network creation/update: Perception codelets descend on the syntactic tree. Those that find features relevant to

their specialty activate appropriate nodes (information codelets) in CTS' Perceptual Network. *At this point, IDA sees energy flows circulate within its Perceptual Memory (Slipnet)*. The activated nodes form the percept that will be transferred to WM.

2) **Percept and other sources transfer to WM.** The percept is brought into WM as a network of information codelets that covers the many aspects of the situation, including some anterior sensing. *IDA sends the percept (the meaning plus some relevant data) directly into preconscious buffers.* As soon as our implementation renders TEM available, we will also send the percept to Preconscious Buffers, but as a copy to what is deposited in WM (the justification for this is given in the next step). Other modules in CTS send their contribution at any time after processing a previous broadcast. The information they send may mingle with the percept, or be spotted by Attention or expectation codelets at any time, preparing coalitions for the coming competition for consciousness (step 4), when the cycle allows them to act.

3) **Local Associations.** As AM and TEM for CTS are still under development, I will supply here the process to the best it is currently known. Encoding codelets react to a new percept coming into WM. They look for aspects that fit their realm, do whatever encoding they are programmed to do, and put the result in the preconscious buffer they are related to. Some do this processing for the TEM, others do it for the AM. *LIDA does not have to cover the previous step as its percepts are deposited directly into the preconscious buffers.* Depositing the new percept in WM at the same time as it is copied to the preconscious buffers allows for a faster reaction time as this new percept may be noticed by attention codelets watching for urgent information, and by expectation codelets that expect the unexpected; both teams will not wait for memories to respond, and will already create coalitions for the next conscious publication.

This new information replaces some of the residual contents of the preconscious buffers, leaving untouched the other fields. The resulting vectors are used as cues by the two long-term memories, which return local associations into their *output* Preconscious Buffers. Ericsson and Kintsch (1995) put forward the hypothesis that experts' superior memory when dealing with their field of expertise comes from their ability to refer rapidly and reliably to long-term memory for domain-specific information. Some attention codelets might exist in CTS that compare what was supplied and what was returned to decide which is most likely to be the proper information.

4) **Competition for consciousness.** Codelets that are familiar together (forming innate or learned concepts), or those that are about the same event, create coalitions in WM (4a). Attention codelets, whose job it is to bring special, urgent, or insistent events to "consciousness", may also see information codelets of interest in WM and gather the appropriate codelets in a coalition they create, or they may join a coalition to increase its likeliness of being elected "winner" (also 4a). The expectation codelets may also have supplied to WM information codelets about a problem they noticed regarding an action initiated in a previous cycle (also 4a). All these codelets forming coalitions sum up their activation in some way (see section 4.1.5) and compete to bring Attention upon them (4b). The competition may also include coalitions from a recent previous cycle. The activation of coalitions decays, making it more difficult for unsuccessful ones to compete with newer arrivals.

5) **"Conscious" broadcast.** A coalition of codelets is selected and has its contents broadcast by Access Consciousness. The broadcast is "heard" (or "seen") by memories, Learner Model sub-modules, Domain Expert, and some BN components (States and Feelings/Desires). Not all will respond, but all use information of concern to them to update their beliefs. Conscious broadcast is the only way the various modules become aware of the new information, although direct unconscious communication does happen:

between codelets when looking for associations and when assembling into coalitions, between expectation codelets and Behavior nodes, and within the BN (between States, Feelings and Behavior nodes, which are meant to represent different neuronal groups). The current content of "consciousness", as organized by the encoding codelets, is also stored in Transient Episodic Memory. At recurring times not part of the cognitive cycle, the content of TEM is consolidated into AM.

6) **Recruitment of behavioral resources.** With the broadcast, all of the system became aware of the situation, including the BN. Relevant States recognize parts of the broadcast, elevate their activation level according to the activation level of the codelet bearing their information, and start pushing energy backward into the Behavior node(s) they are connected to. Feelings and Desires also react particularly at this step, usually sensitive to the situation's type (recall that information codelets have may have either a type or a content, or both); maybe they were already stimulated by previous broadcasts and gain some more stimulation from the latest one. Note that there is no direct connection between what stimulates States and Feelings/Desires.

   Feelings/Desires have effects (energy output) that get modulated by experience in links strengths, and by the active personality of the tutor. Other modules receive the broadcast and start processing the content they recognize. So, by the mechanism of broadcasting and the specific alertness of resources, consciousness offers a nice way to bring into action only the relevant resources.

   After processing the broadcast, modules may have something to contribute about the situation. They send it as soon as they can, but it will be considered only by the next cycle.

*7)* **Selection.** This step involves only the Behavior Net. *IDA's 7$^{th}$ step regards the instantiation of a stream, with the transfer of information from priming (Behavior) codelets and the related infusion of energy. Since CTS does not*

*instantiate streams before they become active, CTS' cycle jumps to the selection of Behavior in the BN.* Among all the stimulated, *executable* Behavior nodes, the BN chooses a single one and executes it. "Executable" refers to a node that has all its preconditions met (all the States it has as preconditions are active). The choice is affected by internal motivation (activation from Feelings/Desires), by the current situation (external and/or internal environmental activation) and by the shape of the stream (length, branching, *etc.*). *LIDA (but not IDA) also sees influences from the agent's emotions, currently absent from CTS*.

8) **Action.** Action is taken in step 8, the final step of the cycle. The execution of a Behavior results in the Behavior's underlying Behavior codelets performing their tasks, which may have external or internal consequences. This part of the process is the only one that may generate an action that can be perceived by an external observer. The released codelets also include at least one expectation codelet (see Step 4), whose task it is to monitor the action taken, and to try and bring to consciousness any failure.

## 4.3 SOME INTERESTING FEATURES

The way CTS cognitive architecture works may seem quite different to what one is used to in "regular" ITS agents; I admit that the high distribution of the processing is rather disorienting at first, especially for someone used to centralized and controlled operations. I provide here explanations about emerging features that justify adapting to this new way of doing things.

### 4.3.1 Tutor is always up to date with the situation (vs. plain rule-based architectures)

One nice feature of Maes Action Selection Mechanism (upon which the BN is based) is that it does not require that a unification mechanism goes through the whole rule base at every cognitive cycle. There is only a lightweight summing process that adds all the energy sources for every Behavior node. The BN "thinking" process is differential, cumulative, with the energy making its way in the network at every cognitive cycle. Thanks to the energy flow and accumulation of activation in the succession of nodes, alternate solutions are always in preparation (if available), if not ready to fire, to take over a path that gets stuck because of effects that delay their realization. This natural alternative preparation also serves well the necessary adaptation of the agent when a solution initially favored gets stalled in the process of being chosen because one condition remains absent of its necessary context (preconditions).

### 4.3.2 Analysis and planning are holistic

CTS is based on a highly parallel architecture. Many processes are permanently kept abreast of what is happening. The Domain Expert, the Learner Knowledge Model, the Learner Profile Model, the Learner Affective State Model, numerous individual codelets, they all analyze the events from their point of view, in parallel. They may spontaneously offer an advice, an opinion, or return a feed-back on an aspect of the situation or about a hypothesis suggested by another module. When they send information to Working Memory (WM), their information codelets either combine into a coalition, bring inhibition to a coalition, or offer a competing proposition, to which other modules may react and supply complementary aspects or opposition. So a proposition might be opposed on the basis of any aspect of the situation, or it might be strongly comforted by multiple agreeing points of views.

At the same time, complementing or following the work in WM, there is the Behavior Network (BN) that also does multi-aspectual planning and decision-making. Its various Feelings may react to the same information but with different strengths; a stream may incorporate alternatives with some common preconditions (with some part of the context being common to them) and some specific States tied to a specific path. A stream may also be favored because of past experiences that increased the base-level activation of some of its Behavior nodes. Past experiences, also explicitly recorded in the Transient Episodic or Autobiographic memory, indicate what result an action has had (for instance, the user either succeeded or indicated his annoyance), complementing or reinforcing the adaptation brought about by the implicit learning stored in BN links strength. So, as you see, there are multiple parallel mechanisms that collaborate to bring on the table as many aspects as possible, at many points in the cognitive processing.

### 4.3.3 Feelings offer an intuitive contextual analysis

One nicety of the architecture is that is combines logical, explicit analysis with "intuitive" analysis. Indeed, we find *intuition* in our agent. The Merriam-Webster Online Dictionary[17] defines "intuition" as «quick and ready insight», «the power or faculty of attaining to direct knowledge or cognition without evident rational thought and inference». The way the BN works yields exactly this kind of instantaneous assessment, not going through an obvious or easily tractable reasoning process. It is always possible to follow the links in the BN, add every source of influence, and justify any implicit behavior in the standard Maes network, even in a big, complex one. But it nevertheless corresponds to the definition. With the addition of implicit learning in the BN, however, it becomes virtually impossible to tract the past sources of the

[17] (http://www.m-w.com/cgi-bin/dictionary?book=Dictionary&va=intuition)

current state of the network, yet a decision can readily be made. This strongly supports the notion of intuition.

I find that this intuitive analysis capability takes our agent one step closer to human cognitive processing.

### 4.3.4 Top-Down and bottom-up adaptation

Adaptation in CTS is not univocally driven by the external inputs, by what is happening in the external environment (mostly what the user is doing). CTS has its own professional goals and beliefs as any expert tutor does (e.g. beliefs about what didactic method yields the best result for that specific student; belief of whether past interactions have validated the efficiency of respecting user's preferences; beliefs indicating the importance of some domain content and the need to prioritize it to insure learner's advance in the curriculum; belief about the necessity to terminate the session soon, etc.). Reciprocally, CTS does not act purely to achieve the goal decided upon some time ago, as if it were blind, deaf or plainly stubborn. CTS plans on the basis of a multitude of Goal nodes simultaneously more or less activated. Decisions and actions always result from the combination of reactive mechanisms and the proactive pursuit of goals. In particular, the BN includes parameters that control the balance between them. They can be set manually by the designer to create a more spontaneous attitude of the tutor, or a more analytical stance where CTS allows more time for the energy to activate longer solution paths. Moreover, external and internal sources are always tempered by multiple internal filtering mechanisms that validate or modulate them.

#### 4.3.4.1 Adaptation with multiple personalities for the agent.

"Feelings", "Desires" and "emotions", in our architecture, aside from motivating actions, are mechanisms forming the agent’s personality. I will not discuss an emo-

tions mechanism, as emotions have not yet been addressed in our design effort (although IDA has had an emotional mechanism for some time but which has been removed, and LIDA presents a completely different approach to emotions in his conceptual architecture). Feelings and Desires nodes are the motivational mechanisms that feed the Behavior Network with activation and so orient action selection in line with the agent's high-level goals. Specific messages that stimulate a "Feeling" may appear when the Learner Knowledge Model signals some important flaw in the learner's knowledge, or may appear after the broadcast of a perceived external situation, such as the possibility of a collision while the user is manipulating Canadarm2. Depending on the tutor's (agent) "personality", the agent will be more or less inclined to take some kind of action, according to the sensitivity of its "Feelings" and "Desire" nodes to the various factors. By creating sets of parameters about the sensitivity of the Feelings towards the various events, states or aspects, and about which Goal nodes are fed and with how much energy (maybe with no energy at all), the designer may implement the various personalities he believes will be relevant in his field of application. Although not implemented yet, this capability might make the tutor a more pleasant fellow for the users. That capability could also be driven by the agent's metacognition as an automatic means of adaptation to the user.

These adjustments combine with the various parameters available in the BN that allow making the agent more jumpy or more analytical, taking more or less time to examine various ways of intervening, and showing to the user more or less promptness in reacting to his actions.

#### 4.3.4.2 Adaptation at the planning level: deliberative capabilities may kick-in

The Behavior Network is the primary means of planning in the agent. The selection of proper behavior comes from the conjunction of the drive coming from the Feelings, and the drive coming from the environment (through the States). The designer can specify the balance between the importance given to the Feelings and that given to the States. A second level of adaptation can be incorporated in the BN

by switching personality profiles for the agent, which specify the relative importance of each Feeling; this changes how it will react to various events created by the user, making the agent more rigid (less "perceptive") to some, or more friendly, *etc.*. So, the agent's adaptation (its planning) may be more or less influenced by the events (in their own rights, with their direct input to the Behaviors). In summary, the BN dictates what CTS should do, globally. But it has been surrounded in CTS with finer adaptation means that intervene to specify the details of an act. Explicit *deliberation* is often involved to complement the generic act selection and obtain the specifics of the situation at hand. For instance, hinting may have been decided upon in the BN as the most appropriate intervention in the current context but the Domain Expert is solicited to supply its specific content. Such interaction will be illustrated in scenarios (section 5.4).

Another type of planning involving deliberation may take the form of preparing lessons plans adapted to the learner (to his current knowledge, his believed deficiencies, the time available, etc.). That plan creation would be scripted in a BN specialized stream and adapted with internal deliberation that goes as follows. The first act brings into Working Memory the announcement that lesson planning is beginning. If selected, the broadcast of this information is “heard” by every sub-systems of the agent but causes only some of them to react. For instance, the Learner Profile Model (LPM) sends to WM the information about the learner’s preferences (for instance, preferred type(s) of interaction, ideal duration of a lesson, etc.). The Domain Expert (DE), on his side, sends the list of the concepts that can be seen next (based on prerequisites). The publication of this new kit of information sees the LKM send supplementary data: the probable level of expertise of the learner about each of these concepts. These are broadcast, allowing the DE to select the proper concepts (of appropriate level of difficulty, none over learner’s capabilities) and send them, including data about their intended durations and their level of priority. In this round, “sending the concepts” means sending links to the appropriate material. Now, all the needed information has been gathered in WM that will allow building a tentative outline for the delivery plan. This makes a first sub-Goal node achieved, and this should be broadcast.

Through this last broadcast, the next Behavior is made aware that all of the information it needs is available. The job of its codelet is to find the most appropriate concept as a starter. Some rules it possesses guide its choice. For instance, it may take into consideration that the presentation should concentrate first on integrating the highest priority concepts (those deemed to be so by the pedagogue who built the domain model), but, based on the learner's profile, it should moreover consider selecting one that is a grade in complexity below the learner's rated expertise: this learner needs to build confidence in its abilities at the first steps. If they happen to be all of the same difficulty level, one is chosen at random. Eventually, the codelet sets its choice on one and puts it as the head of the presentation (delivery) plan in WM. This is noted by an expectation codelet that tries to have broadcast that the first concept has been settled. This again causes a State to turn on to this effect, giving the proper precondition to the next Behavior.

The next Behavior is then executable and, provided, as always, that it is the most activated executable node in the BN, it receives a green signal and sends its codelet, who's job is of adding a major (priority) concept to the plan. As long as unassigned information codelets containing major concepts are available in WM, an attention codelet will gather them and have this information broadcast. This will reassert the preconditions for our little plan building friend, which will be solicited again to do its job of selecting and adding a major concept to the outline. When no additional major concept is to be enlisted, or when an attention codelet computes that these concepts cover most of the time available for the lesson and mentions it in WM, the next act will start and look at the minor concepts. When the plan is set into motion, the other adaptation means get involved. For instance, during the effective delivery, time may run out with respect to the allotted time for a lesson. In this case, learner will be signified of this and offered the choice of continuing the lesson or deferring the remaining concepts to the next session. Moreover, during the live presentation, CTS "professional Desires" will do their job and bring-in some interleaving of learning with complementary activities (simple questions, more elaborate questionnaires or tests, exercises) according to the implicit or active pedagogical theory (see

next sub-section). Maybe some interaction will be initiated that will help the learner to relax, refocus, or regain confidence in his (or her) achievements.

Another, quite important, example of deliberation is the possibility of some information source to oppose the propositions of another one. For instance, when a Behavior proposes taking action, the Learner Affective State Module may oppose because it believes it would bring the user's motivation (confidence) too low. This is consistent with what Libet describes (Franklin and Graesser, 2001, citing Libet, 1983) as the veto that volition can put after an action has been initiated unconsciously but before it can manifest itself to the exterior:

> Freely voluntary acts are preceded by a specific electrical change in the brain (the 'readiness potential', RP) that begins 550 ms before the act. Human subjects became aware of intention to act 350-400 ms after RP starts, but 200 ms. before the motor act. The volitional process is therefore initiated unconsciously. But the conscious function could still control the outcome; it can veto the act.

#### 4.3.4.3 Adaptation with multiple pedagogical theories

We can implement multiple pedagogical theories in the BN. However, much work remains to be done in order to translate pedagogical theories in exact BN structures. Since I concentrated first on obtaining basic coaching functionalities (in accordance to our initial concern, astronauts training on a simulator) little effort went into using pedagogical theories. However, here are a few thoughts on how to go about it.

The various aspects of the selected theories can be put under specific Desire nodes and under various Goal nodes. The "Desires" (which, as you will recall, only differ from the "Feelings" in what they are intended to do: attend to the agent's professional and personal goals) may be used to detail the various general categories of behaviors (gaining attention, planning, presenting what is to come, having student generate ideas about issues and answers, provide related cases, provide worked-out examples, pretest, offering self-assessment, presenting a concept, verifying the possession of knowledge, offering affective support, offering metacognitive stimula-

tion, submitting exercises, and so on). The many ways of performing these "professional acts" appear as Goal nodes connected to one or many Desires. Then, the designer describes how each pedagogical theory connects with each Desire (to which Desire, in what context, with what intensity/priority).

Metacognitive codelets watch how things are going with the learner. Equipped with some inference rules, they are apt at trying some modulations on the Feelings sensitivities. Others watch over them and, upon judging that insufficient progress is made, may call for a vote (a deliberation) about the necessary shift towards another pedagogical theory. Transiting to another theory might significantly change the tutor's behavior, or it may affect it in some specific areas. Each Behavior remains subject to Learner Model's vigilance, which always presents learner's preferences as support or opposition to an act (or otherwise support or opposes). For instance, if the Learner's Model states that, for the kind of learner CTS is presently interacting with, or because it is trying submitting him to a higher level of difficulty, more frequent verifications should be made about his comprehension, the activation level of the appropriate Feeling(s) will rise faster, eliciting learner's performance more frequently, modulating the actual pedagogical theory. Observation of learner's reactions indicative of a possible lack of understanding may also stimulate some diagnosis acts to kick-in on the spot, without waiting for the completion of the lesson.

#### 4.3.4.4 Adaptation at the coalition selection level

The Attention mechanism selects the most activated coalition in WM, that is, which contains the most activated codelets[18]. There is an innate value to the information about the domain, specified by the system's designer. But there is also dynamic values given to some information codelets. First, the information coming from

[18] The exact calculation is algorithmic. See section 4.1.5 for details.

the learner model is dynamic; some information sees its gravity or the certainty of its belief change with time, thus variations in its activation. Second, codelets issued by the BN receive an activation value that reflects the activation value of their Behavior node. For instance, if a Behavior is highly activated either because fed by a strong Feeling, or it is fed by more than one Feeling, or it had much time to accumulate energy, the codelets that implement the Behavior will be highly activated when they are released. Another reason for high activation is related to the learning that happens in the BN. The base-level activation of each Behavior node changes as expectation codelets confirm the success or failure of its action. The more often successful, the higher its base-level activation and, thereof, the *total* activation reached by the node. Its codelets inherit this activation level, after normalization. So, the coalition of codelets that is selected by the Attention mechanism, eventually an action towards the environment, is adapted to the learner and the situation since some codelets had activation originating bottom-up (States) and top-down (Feelings).

**Attention bias.** Attention bias can be voluntarily created with attention codelets (*which role differs somewhat from the one they play in LIDA*; see section 4.1.1.2). Doing so may be useful to change temporarily the significance (importance) of some information. For instance, in the case of the tutor trying to correct a bad habit in the way the astronaut does a certain maneuver, it becomes important to process the next maneuver, even if in itself a joint rotation does not bear much importance and could normally be easily superseded by any other information appearing in WM. That bias creation would be a remarkable adaptation to the specific need of the user.

### 4.3.5 Learning is decentralized into multiple structures

CTS' conceptual architecture includes structures specifically related to knowledge and memorization: a Transient Episodic Memory and a Semantic Memory. I already described them. There are also other ways CTS learns, in supervised and unsupervised ways that lead to explicit and implicit knowledge[19]: learning of regularities in WM that lead to new known entities or situations, and supervised implicit learning in the BN that translates into base-level activation of Behavior nodes and into BN's links' strength. I present these mechanisms now.

#### 4.3.5.1 Discovery of regularities in Working Memory

Learning of regularities in Working Memory is the first of the four types of implicit learning in CTS (the other three being links strengthening in BN, modification of base-level activation of Behavior nodes, and new associations and generalization created in TEM). It designates the associations that form between codelets that spend time together in WM, in a Hebbian learning. It is founded on Jackson's extension (Jackson, 1987) of Selfridge's (1959) Pandemonium theory that describes daemons reinforcing their associations with others while they are also found in the arena (those that are active). LIDA has the same learning of associations between codelets that spend time together in the *playing field* (according to Jackson's Pandemonium). *However, in CTS, this implicit learning of regularities applies exclusively to information codelets and functions a little differently*. It is intended for the discovery of regularities in the environment and regularities happening from the internal processing of the agent that deserve to correspond to explicit concepts. Not all initiated associa-

[19] We may be tempted to associate supervised learning with explicit knowledge, and unsupervised learning with implicit learning. However, there is no exclusive relation. For instance, explicit learning may lead to implicit knowledge, such as in acquisition of reflexes.

tions do correspond to regularities. Conservation of only confirmed regularities is assured by the principle that links have to survive their constant decay. If not reinforced soon enough, an association will decay away. Infrequent coincidences have little probability of corresponding to a regularity, or it does not happen often enough to warrant the creation of a concept about it. I hypothesize that codelets that meet often in WM correspond to features of a common idea. Their association may reach a level that designates them as a coalition. Until they become a learned concept, I call this learning implicit, as it only exists in links, and the would-be concept cannot yet be used for reasoning; it remains latent. At the threshold point, their links become permanent and, as a coalition, they may be selected by the Attention mechanism for publication. This, in turn, will allow the semantic memory to learn it, to add it to its structure of concepts. The codelets of that new concept that are common to those of other concepts will naturally create semantic links with other concepts already in memory. This last idea has not yet been thoroughly worked out, nor implemented (our team has yet to implement both long-term memories).

#### 4.3.5.2 Supervised learning in the BN: Experience as a planning factor

In any artificial agent, a behavior is designed with a goal in mind, with some effects expected. If experience reveals that some behavior does not work so well in a situation, it should be avoided, or at least, kept as a fall-back solution. We implement the means for that kind of intelligence as base-level activation in the Behavior nodes. Nodes activation is composed of the current stimulations, brought about by the energy infused by every energy source it is connected to (States and other nodes), and of its base-level activation. That last value integrates the result of past experience and measures the act reliability within its context. When a Behavior is successful, its expectation codelet(s) send(s) back a confirmation signal that brings the Behavior to elevate its activation level. In the next occasion where this node is part of the planning, it will be favored for election because it will reach the "go" threshold sooner, already having some activation.

This base-level activation grows as a sigmoid curve, as described in section 4.1.3. After the initial phase of slow increase, there is a fast learning phase that slows down when the Behavior is about as strong as it can be. Just like in human, learning is constantly subjected to forgetting. If reinforcement does not come soon enough, what has been learned might be erased by decay. Our agent will not forget its original BN sequences since they are innate but it may loose the additional benefit of experience (implicit reinforcement that it acquired). The inverse of the sigmoid function serves as the forgetting curve, but we give it a slope softer than the one used for learning. Indeed, humans generally forget more slowly then they have learned. For instance, if we learn a karate motion in half-an-hour, it is very likely we will remember some of it at the next training several days later, that is, after a period much longer than what it took to assimilate it.

#### 4.3.5.3 Unsupervised (implicit) learning in the BN: experience as an efficiency factor

Implicit learning may designate two realities: learning without awareness, and things someone learns (stores) in a form that is not explicit, that cannot be manipulated in reasoning. In any case, it generally refers to procedural learning that leads to automaticity in some gesture. Being an important part in an agent's adaptability, a complete Master thesis has been devoted to it and offers the complete description (Faghihi, 2007). I provide here the general ideas that we are implementing.

Aside from showing what gets the job done, experience has other closely related effects. The reinforcement from practicing brings both procedures to fire more easily (automaticity of decision) and to proceed more automatically (automaticity of execution). This phenomenon is quite obvious in sports, but is just as real in any activity. We want the same thing to happen in our BN: the more a stream is used, the faster it should come into a ready state, and the less it should involve consciousness for its unfolding.

We implement the phenomenon of "faster reflexes" through reinforcing the strength of the links between Behavior nodes in the BN. Stronger links modulate positively (amplify) the energy that transits through them, increasing its value. The main effect that this produces is that *anterior* nodes (the nodes in direction of the temporally first Behavior of the stream) reach the activation threshold sooner, increasing the likeliness of the stream being selected for action.

Exerted gestures or procedures need less conscious involvement to unfold. In fact, one cannot even do a golf swing or a tennis drive if at least some minimal automation has not taken place. A well exerted golf swing requires only the initial analysis that decides on the value of the parameters (distance to reach, angle, strength to apply). Then, when the move is started, little consciousness is needed. Only the visual feed-back of the ball falling off the tee or the unpleasant vibration of the club hitting the ground will come to consciousness before the end of the swing. In CTS, we considered implementing this kind of automaticity by "fusing" Behaviors together. Although this idea is rather seductive, its implementation is quite complicated and remains under study for the moment. We rather have implemented a modification in the Behavior nodes that allows them to directly stimulate States that are on their effects list; that way, the State does not need to hear the proper consciousness broadcast to become excited (turning on). The next Behavior finds its preconditions in a ready status at the next cycle and is immediately *executable* (Maes' term meaning that a competence module, a Behavior node, has its "logical" preconditions met). In that way, consciousness is not required for the stream to flow steadily. Expectation codelets continue to exist, but their role becomes to negate preconditions of Behaviors further in the chain, if expected effects do not come up; if that negative conclusion happens soon enough, it may block the continuation of the stream, just as becoming conscious of a problem would make us stop an automatic movement.

### 4.3.6 Designers may not need to do any programming

In developing our Behavior Network Editor, we have tried to put the basis of an easy-to-design agent, in the hope to put CTS within the reach of more researchers and practitioners. Its graphical user interface (see section 4.5 for an illustration) allows the design of the BN without having to type XML code. Much can be specified at the information codelets fields level. With the addition of a similar tool for the Perceptual Network, aside from field specific analyses, one may obtain a complete tutoring system without typing a single line of code.

## 4.4 A FEW WORDS ABOUT THE IMPLEMENTATION

The descriptions in this section owe much to Hohmeyer's Master thesis. The content also reflects Gaha's works for the implementation of modifications and extensions as part of his Master research. For a more detailed account of the mechanisms, refer to Hohmeyer (2006) and Gaha (2007).

We have adopted the Java language for implementing CTS, a popular platform, and the one used for IDA. The *object* philosophy of Java offers a natural parallel to Baars' idea of an audience of multiple independent specialists in the brain. Other features of Java also support the project quite well, such as the publish-and-subscribe mechanism: it resembles consciousness broadcasts and presents a very accommodating implementation vector.

Many of the ideas underlying IDA have been taken back into CTS, but the code in itself is completely new[20]. There were advantages to rewriting, but it also

[20] The reason for that is a practical one. As with any ongoing research project, IDA and LIDA are constantly being modified and improved, with temporary hypotheses, trials and hooks left in their code. With our limited knowledge and understanding of the hypotheses, we concluded that it would be easier to start anew then stitching our additions to the parts of the code that were directly applicable to our domain of a tutoring agent and readily reusable.

brought challenges, both in the implementation aspects and on the conceptual field. In its actual state, CTS encompasses nearly a thousand classes.

The whole architecture is highly modular. All modules (or rather, "packages") can do processing on their own, sensing their environment and responding to it. However, they need to use the Scene functions for their communication. The Scene creates a hub among all other modules that permits a coordinated processing in CTS. Aside from the Scene, two other modules form the essential core of CTS as a generic agent: Perception and the Behavior Network (their mechanisms are generic, although they are both mostly domain-specific in their content). Complementing the generic agent are the Semantic and Transient Episodic memories, still under development. Some modules are add-ons that extend the generic agent in its specific role: Learner Model and Domain Expert. All these need the “central” module to communicate. Since everything is contained within Baars’ theater, this will be our first point of

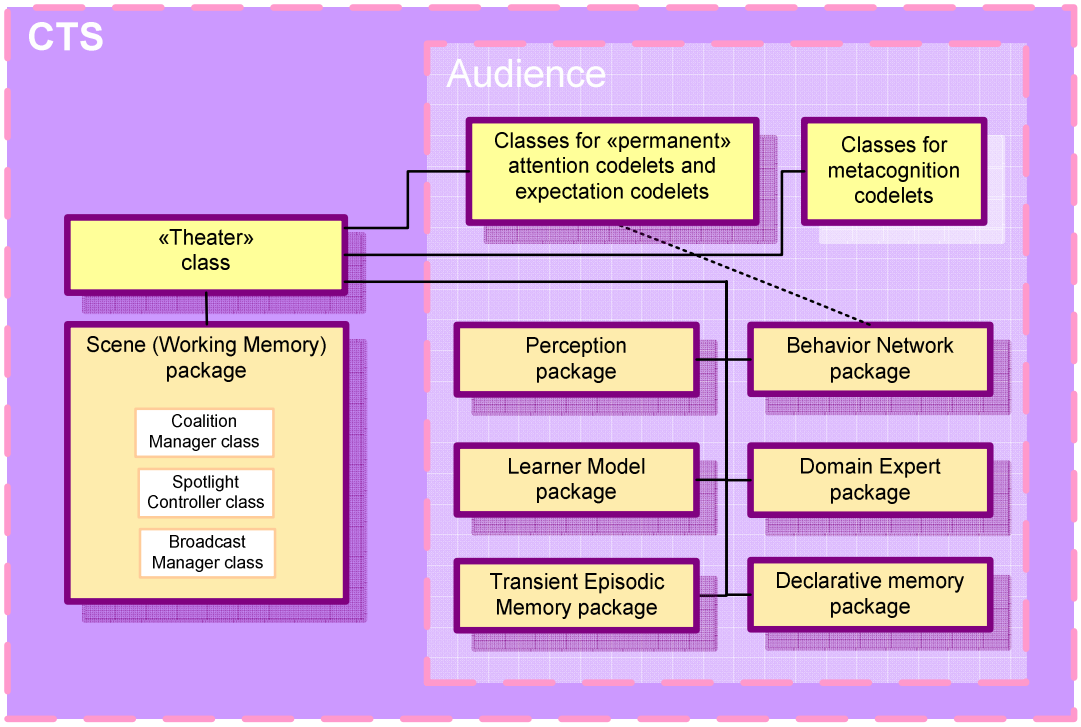



**Figure 14 CTS’ computational architecture**

interest. As you will see, much of Baars' metaphor has been kept in naming modules: Theater, Scene, and so on, and will often appear in the descriptions.

#### 4.4.1.1 The Theater within CTS

The Theater class in CTS has roles that correspond in part to the original idea of the "building". It creates the Scene and contains every codelets that is created by any entity. It incorporates the very fundamental process of the agent: it drives the loop for the cognitive cycle, which calls (directly or indirectly) every other objects for their turn of execution. Permanent codelets (codelets are objects instantiated from codelet classes) are called for and found here, such as attention codelets that need to run from the beginning to the end of CTS activation, or information codelets that need to be available before starting the cognitive cycle.

The Theater drives the cognitive cycle, calling codelets at least once in every cycle, when their group's turn comes in the list. This manner of functioning has replaced our initial idea of having a thread for each codelet[21].

[21] Although Java threads seem to correspond well to what we know about the distributed, parallel processing of the brain, using one for each codelet exceeds our debugger's capabilities, seemingly limited to 200 threads. Thus, we chose to use a loop that calls each codelet once every cycle. IDA programmers also initially thought of devoting a thread to each codelet and to most any element of the architecture (such as the BN elements: Behavior nodes, Goal nodes, etc.). However, they observed that codelets are sometimes delayed for reasons having nothing to do with the agent's architecture. Threads are being controlled by Java with its own set of priorities. Programmers concluded that threads are not a reliable way of reproducing the mind's processors.

#### 4.4.1.2 The Scene module

The Scene “module” (package) is generic and totally reusable for any application. It supports the “conscious” aspects of the architecture, that is, competition and broadcasting. It corresponds to the scene and the spotlight of Baars’ theater metaphor. As such, it offers the framework within which codelets interact.

Two auxiliary classes take on essential roles for the consciousness implementation: the Coalition Manager and the Spotlight Controller. They are needed to implement the formation of information coalitions in Working Memory and the information selection prior to the publication of the winner. At every cycle, the Scene calls these two classes for the execution of the competition in working memory (step 4 of the cognitive cycle). First, the Coalition Manager is called up for the determination of the coalitions. It returns to the Scene the collection of the coalitions, which then passes it on to the Spotlight Controller. This class simply chooses the coalition showing the highest activation. The publication of the winner to every module and every listening codelet happens by the means of the publish-subscribe mechanism, under the control of the Broadcast Manager.

A generic Java class, `Codelet`, specifies the generic behaviors of the codelets while they are alive (inside and outside Working Memory): progressive activation decay and mechanisms to join and leave the Scene. This base class also contains abstract methods for the specific behaviors of each type of codelet; this allows easier extension while enforcing stable behavior when creating new domain-specific behavior codelets.

Adding extensions can be done by calling the `setPlugInManager()` method. It is a very simple class containing only one method for each module that is added. The client application has to create a subclass of `PlugInManager` returning an operational implementation for the implemented modules. A simple but very useful extension has been the *Consciousness Viewer* (**Figure 15**). The Scene signals every arrival and departure of codelets, which is then shown in the Scene window of the viewer so that we, designers, can see what is inhabiting the Scene (WM).

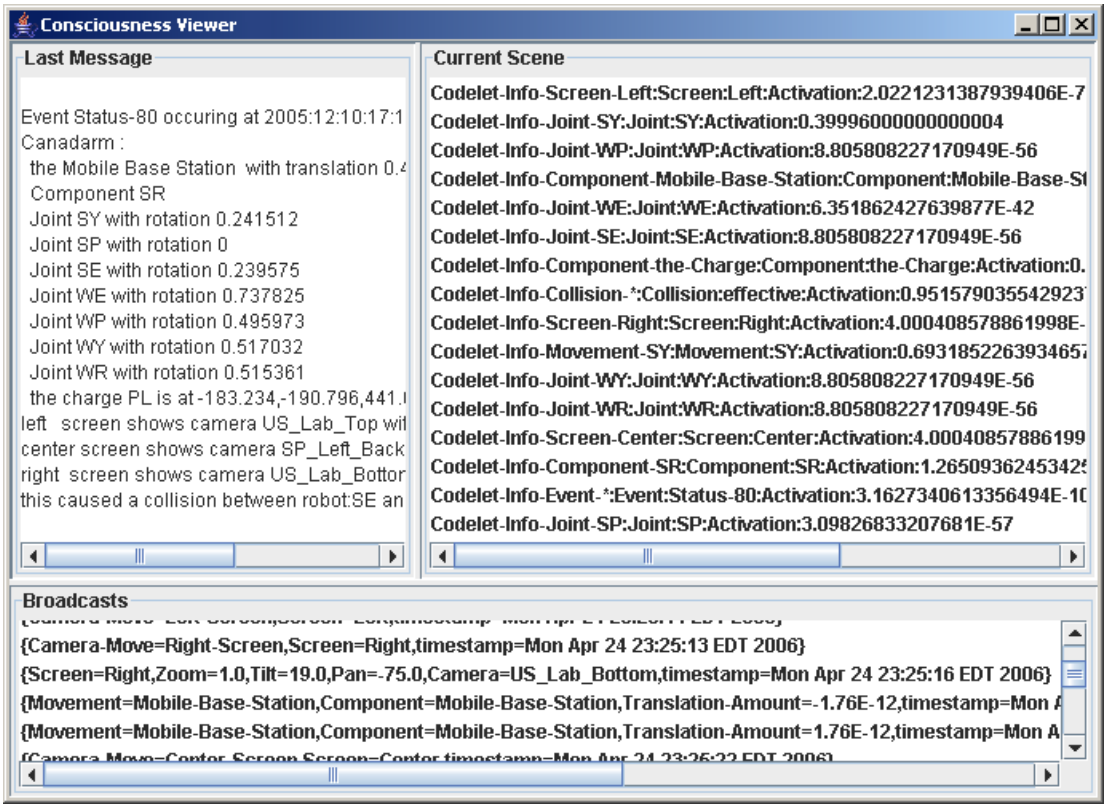



**Figure 15 The Consciousness Viewer**, a simple but very useful extension of the Global Workspace class, for comprehension and debugging needs.

## 4.5 BN EDITOR, AN AUTHORING TOOL TO HELP ELABORATING CTS

For the elaboration of the Behavior Network, we have developed the BN Editor. This graphical editor is meant to help designers think about the high-level Goal nodes, sub-Goal nodes, streams of Behaviors that can accomplish them, and codelets that realize their actions. One uses the icons to create stream boxes, put Goal node and Behavior nodes in them, link them, add their contexts (precondition and effect States). It also supplies the environment to specify the codelets that underlie the Behaviors. Specific behaviors can be indicated by telling what the implementing class is. The resulting network is saved in an xml file.

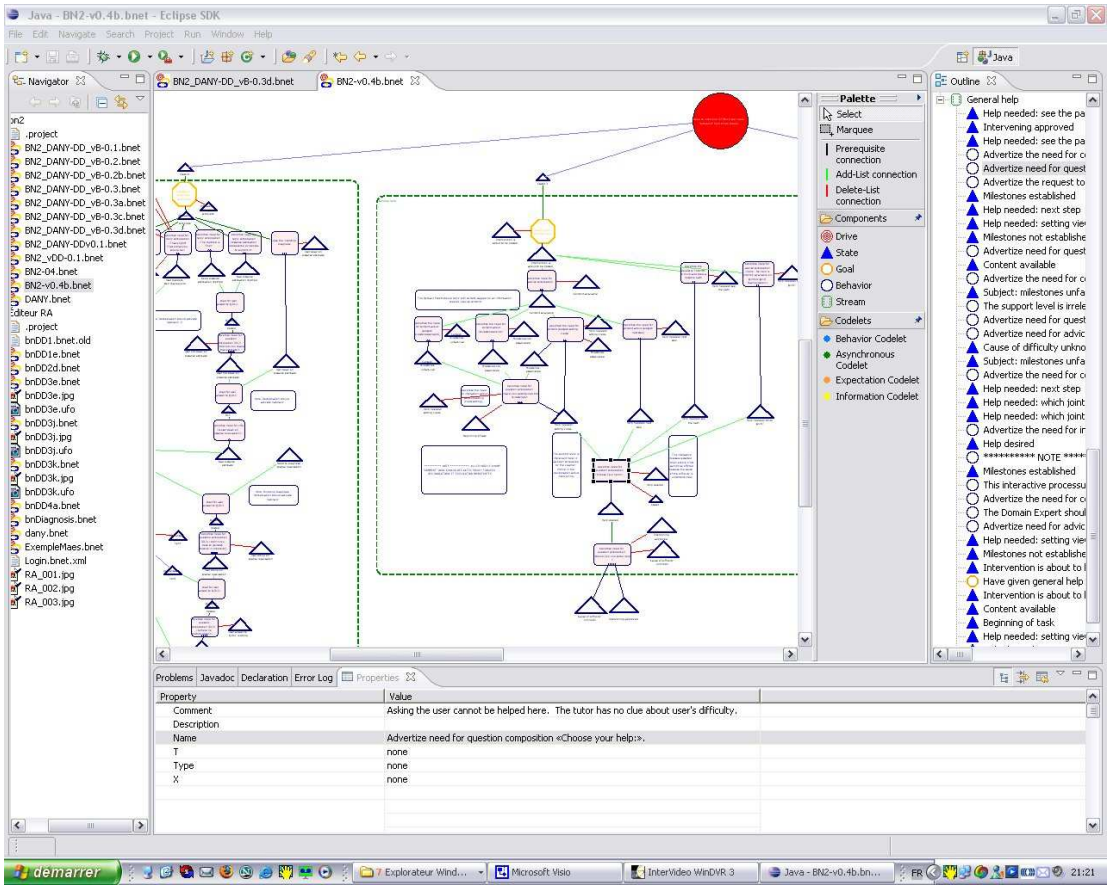



**Figure 16 The BN Editor.**

# Chapter 5

# INSTANTIATING CTS IN CANADARM TUTOR

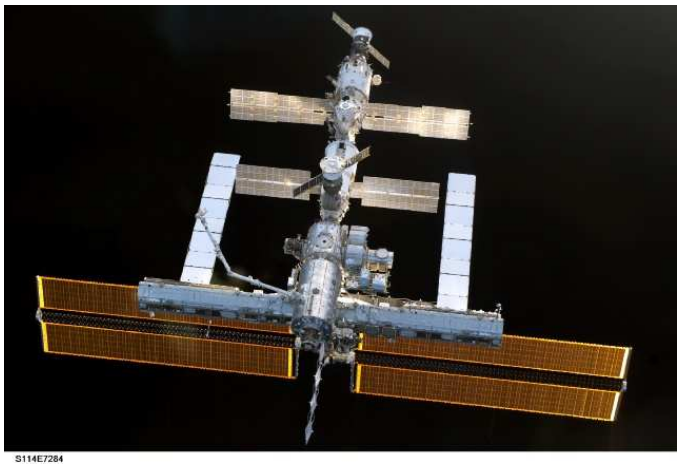



**Figure 17 The International Space Station as it was on August 6, 2005.** The most complex piece of technology ever designed by Man. Canadarm2 appears as the slim structure extending "on top" of the Truss (at the forefront in this picture) and to the left of the main axis formed by the chain of modules.

*(Source: http://spaceflight.nasa.gov/gallery/images/station/assembly/lores/s114e7284.jpg)*

## 5.1 TUTORING CONTEXT

The International Space Station is the most sophisticated piece of technology ever built. It has been designed to sustain life in space and permit scientific experiments. Thus, it needs regular replenishment in food, parts, fuel, experimental setups and other cargo. Containers return to Earth filled with used material and trash. The availability of a crane attached to the Station is a big bonus for all the manipulations

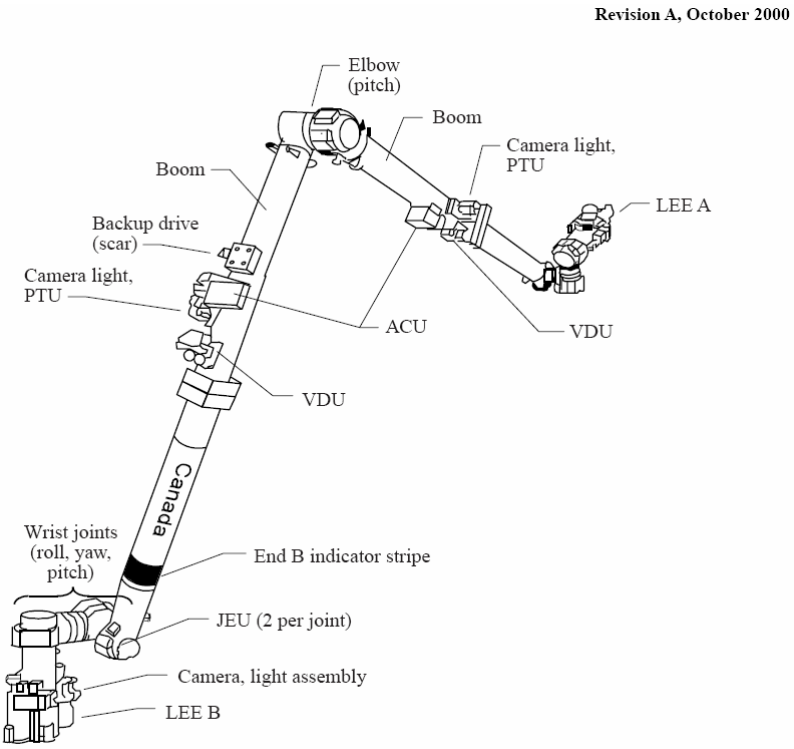



**Figure 17 Canadarm2, a complex robotic telemanipulator with seven joints.** With three joints at each end (wrist joints) and one in the middle (elbow joint), the difficulty of manipulating this device cannot be fully appreciated before trying to. Astronauts need a thorough training, and frequent refreshers. (*Source: NASA (2000) p.3-69*)

incurred, especially when they involve large size equipments such as the photo-voltaic panels (that supply the Station with electricity). Canadarm (attached to the space shuttles) and Canadarm2 (mobile around the Space Station), the Canadian contributions to the international project, have been saluted as great Canadian achievements, offering to the astronauts tools that are wonderful to use as they almost feel like a natural extension to the human arm.

However, there are constraints when dealing with the robotic arms. Even if de-signed to be controlled with just two hand controllers, their operation really is quite complex and needs serious training. With seven degrees of freedom (as the human arm), this robotic arm is much harder to operate than regular cranes used on Earth

**Figure 18 Portion of the workstation that allows controlling Canadarm2 on the ISS.** The astronaut manipulates the robotic arm with two joysticks. He has to optimize the limited views offered by the three monitors. (*Source: NASA*)

(see Figure 17). Lots of procedures and security check-ups have been put in place and need to be accomplished by the astronauts.

Compounding the difficulty of operating Canadarm2 is the fact that astronauts cannot just sit at one extremity and watch what they are doing, as is the case for any ordinary crane on Earth. The operations can only be observed indirectly, in 2D, through monitors (see Figure 18). There is no window to "look outside", and cameras must be used to see. The choice of cameras is limited: only camera ports CP3, CP7, CP9, CP12, CP13, CP14 have received cameras (see Figure 19 for their location), plus the four installed on Canadarm2 and one on its mobile base. The astronaut has to select the best choice of cameras and create the views according to very general rules such as «the central view has to show the global situation», and «The other two views have to offer a detail view and a view that permits to measure the distance of Canadarm2 elements to the ISS». A complete check list of about 20 items indicates what has to be done or verified every time before putting Canadarm2 into motion, and gone through again every time the astronaut stops in order to replan the path or needs to move away from the station, even just for a few seconds.

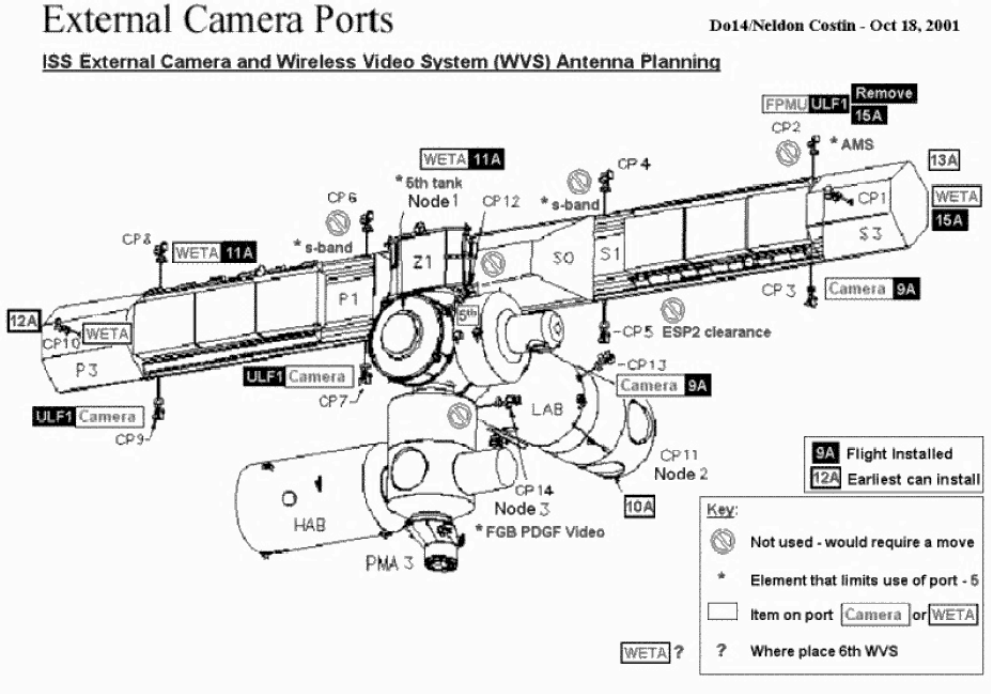



**Figure 19 Location of the ISS camera ports that can connect installed cameras to Canadarm2 workstation.** (*Source: unknown*)

Lots of concepts and procedures have to be mastered, such as the position and function of every control on the workstation (Figure 1), selecting the information sources, placing them on the different monitors, immobilizing Canadarm2, the various coordinate frames, making transformations from one coordinate frame to another, etc. Two crucial abilities that also have to be acquired deal with spatial representation and reasoning. *Spatial awareness* is about knowing where "the astronaut" is standing and where things are around "him", and at what distance – I use quotation marks around "the astronaut" meaning that the astronaut is not outside, manipulation payloads, but the views make him feel as if he were Canadarm2 himself. *Situational awareness* is about being able to understand and predict where things are going and where they will be with respect to one another. The astronaut has to be able to integrate the information from three separate views and reconstruct

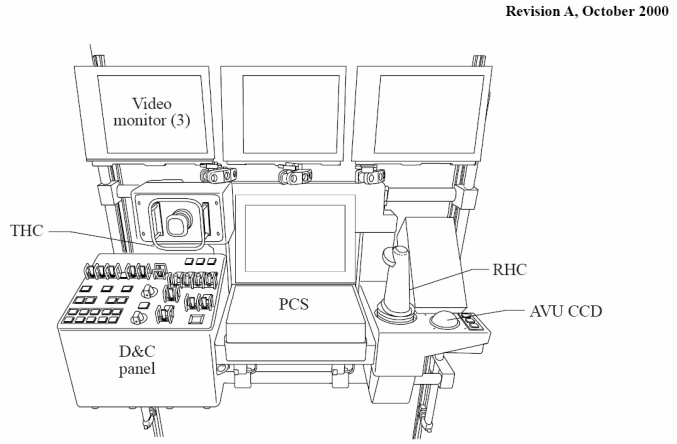



**Figure 1 The ISS robotic workstation used to control Canadarm2.** *Source: NASA (2000) p.3-71.*

a spatial map in his head, recognizing important elements, seizing distances, and extrapolating the resulting Arm displacement when he acts on one of the seven joints.

The way astronauts are presently trained involves a rather traditional class-room for the theoretical portion of their formation. After getting half a day of conceptual presentations and theoretical exercises, they spend the rest of the day in a simulation room where they are put in a pretty realistic setup with a workstation mock-up. A human tutor stands by their side, giving initial instructions and coaching while they accomplish the prescribed maneuvers. The time allotted for completing a task is pre-established, and astronauts are noted on many criteria. However, they are encouraged to ask as many questions as they please, making full use of the resources at their disposal for an optimal learning experience.

## 5.2 ACTIVITIES AND SERVICES IN CANADARM TUTOR WITH CTS

The first version of Canadarm Tutor is called *Roman* Tutor. It is a tutoring system meant to train users on the manipulation of any robotic arm through a simulator.

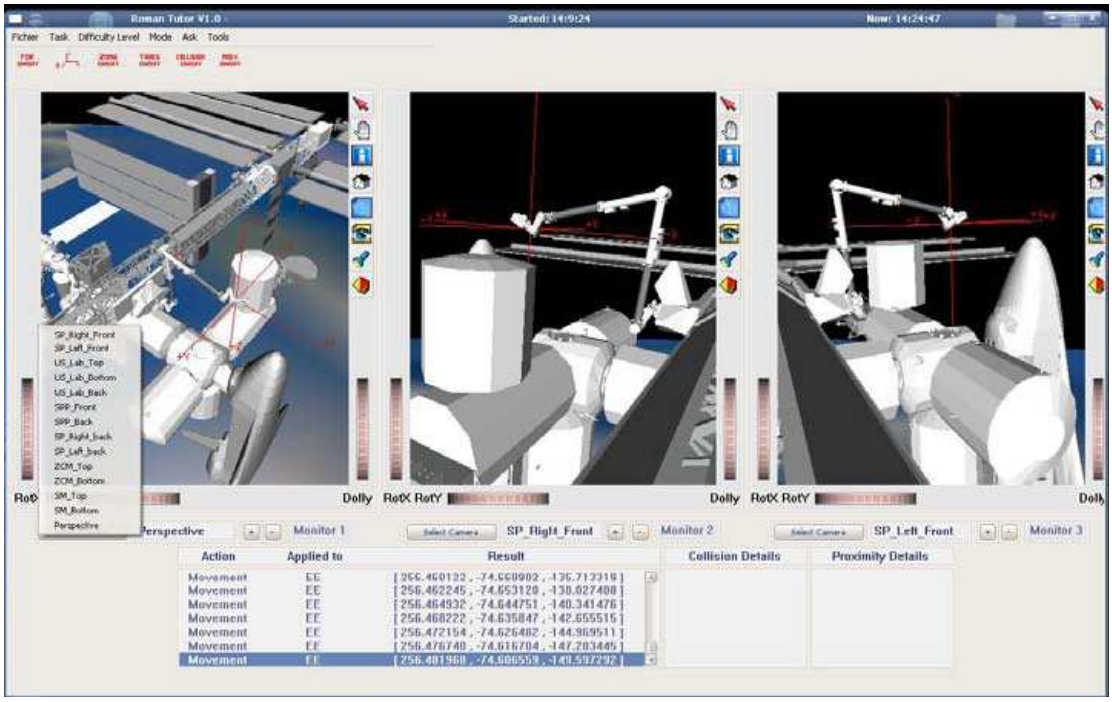



**Figure 20 The non-cognitive Canadarm Tutor.** Menus at the top allow the selection of various functions: task desired, mode of operation, level of difficulty, what-if scenario, *etc*.

In its application to the training of astronauts on Canadarm2, it received the more evocative name of *Canadarm* Tutor; I'll use that appellation from now on.

In its original version (Kabanza *et al*, 2005), Canadarm Tutor is a non cognitive tutor. I will describe its features and the main services it offers before explaining what CTS brings in.

### 5.2.1 The *non cognitive* Canadarm Tutor

The tri-dimensional problem space for Canadarm2 manipulations around the ISS is infinite. It cannot be modeled unless tasks are heavily constrained. Being able to plot a course around obstacles with the kinematics constraints coming from Canadarm2's structure is difficult enough. But in most situations, Canadarm2 is piloted by a human (Figure 18), and only cameras allow him to see how things are

going. Most of the cameras being fixed adds the constraint of restricted sight, creating *less desirable zones* to bring Canadarm2 into.

Facing these difficulties and constraints, Kabanza, Belghith and Nkambou proposed relying on a new, powerful path planner. Its FADPRM (Flexible Anytime Dynamic Probabilistic Roadmap Methods) algorithm provides a framework to support spatial reasoning within the simulator. It makes it possible to do model-tracing coaching with instant shallow feedback. I say "shallow feedback" because FADPRM can only signal observable events at the logical-physical level. The path planner is connected at the lowest logical level of the simulator (which is made up essentially of physical components in the environment, such as the robotic arm, the obstacles, the cameras, and spatial volumes). With the available information, Canadarm Tutor is able to flag proximities, dangerous zones, problematic configurations, and straying from the proposed path.

Aside from coaching, the non cognitive Canadarm Tutor offers other training facilities. There are *spatial awareness* tasks such as name-and-localize exercises that teach and train on knowing "what is where". Indeed, in real life, the views returned by the cameras cannot be "straight" all the time, and they are often upside-down: most of the cameras currently available are the "upside-down" cameras connected on ports 3, 7 and 9 (Figure 19). Consequently, they offer "upside-down" views (if there is something like "upside" things in space…) and not at all natural to interpret for humans used to living on Earth! There are also distance evaluation exercises that help the astronaut sharpen his perceptual abilities. Manipulation tasks train the astronauts on efficiently using Canadarm2. Manipulation tasks can be executed with or without assistance. With the *what-if* menu option, the astronaut can try alternatives before he effectively implements his best choice.

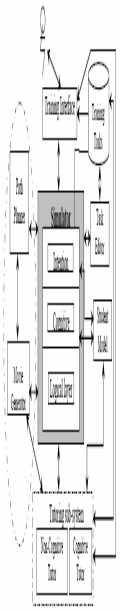


**Figure 21 Architecture of Roman Tutor, cognitive version.** *Source: Nkambou, Belghith and Kabanza, 2006.*

### 5.2.2 The *cognitive* Canadarm Tutor

In its new framework (Nkambou, Belghith and Kabanza, 2006), Roman Tutor has made room for a cognitive tutor and added higher-level knowledge about the physical environment. This supplemental knowledge is organized as a second and a third layer on top of the logical one (see Figure 21). The *cognitive layer* aggregates the logical level in terms of zones and corridors of safe operation, and annotates them with different domain knowledge elements. The *intention layer* specifies structures of predefined tasks by assembling corridors and zones in various possible global paths between a starting and an ending point. This intentional level makes it possible to better follow the evolution of the astronaut in the execution of the selected task. The whole environment is thus aggregated into various areas annotated with appropriate knowledge in order to get more semantic richness in guiding the astronaut during Canadarm2 displacements.

With CTS, Canadarm Tutor makes use of the information in the three layers, in addition CTS' own information sources (its own Learner Model, Domain Expert, declarative memories, *etc*.). CTS complements Roman Tutor by adding cognitive modeling for deeper analyses of what causes the difficulties, and for better adaptation to the learner. Just as its non cognitive version, Canadarm Tutor with CTS is meant to offer both theoretical and practical training when the astronaut is away from the training facilities. But it does so in an adaptive way, selecting material and adapting the presentations in accordance to the learner's progression and adjusting the interactions to his personality.

CTS uses the non cognitive Canadarm Tutor facilities as a foundation for its interventions. For instance, it asks the simulator to record the learner's operations so that it may ask a replay of some sequence when it detects a problematic situation. Or it occasionally asks FADPRM to verify whether there is still a way to the goal. CTS also adds some deep analyses of the learner's actions, with many levels of pattern recognizers implemented as expectation, attention and metacognition codelets. When CTS is alerted of problematic situations or behavioral patterns, it uses all aspects of its learner model to try to find the cause of the problem and then decide whether to intervene, and how.

CTS is not limited in the ways it intervenes. Although not implemented yet, many pedagogical theories and strategies can be part of its Behavior Network, switching to another strategy when a first attempt fails, and even adopting another pedagogical theory when the user does not seem to perform well under the premises of the actual one.

## 5.3 SERVICES OFFERED BY CTS

The services offered or supported are basically the same as "plain" Canadarm Tutor, to which it adds the followings (currently implemented, designed, or simply planned):

- Analyses of learner's performance (partly implemented)
- Probabilistic modeling of learner's knowledge and competencies (under design)
- Selection by CTS of the next activity, based on analysis of past performances (under design)
- Capability to create lessons plans adapted to inferred learner's knowledge (designed)
- Capacity to intervene at appropriate times with methods appropriate to the learner's profile and past performances (under design)
- Partial analysis of views on monitors (planned)
- Affective support (planned)
- All these services being at the disposal of remediation operations (planned)

With its ability to learn, CTS has a clear advantage over the non cognitive tutor. It can track what the learner is doing, anticipate the problems and plan its actions. It learns over the time what works and what does not when it intervenes (methods, timing, style). The statistics it keeps in its Learner Model indicates where the learner has weaknesses, so that CTS may chose efficient remediations (concepts and procedures that need explaining, operations that need more practice, etc.).

## 5.4 EXAMPLE SCENARIOS

I present two scenarios that allow the demonstration of CTS features and performance in different circumstances.

CTS presently has the capability to tutor either as a coach or as a more traditional ILS (Intelligent Learning System) with question-answer type interactions. The latter may be used anytime, and it may come handy as a remediation to diagnosed

problems during a coaching session. Coaching seems to be the most difficult aspect to implement, and this is the type of interactions that I will be showing.

Both our scenarios refer to the same task. The first scenario shows the astronaut at the beginning of a manipulation exercise, which requires that the astronaut move Canadarm2 from a starting point/initial configuration to a final configuration. A problematic situation happens right there, at the beginning: the astronaut takes a first action which is not the right one, forgetting to adjust views before moving the Arm. In scenario 2, after making some manipulations, the astronaut immobilizes the Arm and does nothing for a prolonged time. In both cases, CTS has to decide whether it will intervene or remain silent. CTS disposes of different ways of intervening: giving hints, stating the fact, showing the problem. Ideally, CTS will use the user's preferred way (either stated as part of its learner profile, or as explicitly indicated by him; the latter is a functionality yet to be implemented). We could even have CTS interact in a *style* appropriate to the astronaut, that is, using a style and a "tone" simulating an empathic tutor, a "straight" one, or a friendly one. For our initial prototype, we do not have natural language generation algorithms, and textual interventions can only offer pre-written textual variations for the same elements of intervention. Since I haven't prepared the necessary variations, that aspect will not be demonstrated.

In Canadarm Tutor as in Roman Tutor, manipulation tasks all use the same pattern: a) show the initial position or configuration of Canadarm2; b) show the destination or the final configuration; c) start the chronometer and coach the astronaut. To ensure that we can measure the time taken by the astronaut to plan the path, we allot a limited time for inspecting the destination/ending configuration.

I give a few more general explanations before getting to the scenarios. Preceding the manipulation of Canadarm2, there is a complete list of verifications, settings and planning that the astronaut must cover, such as setting the information sources and choosing where to display them on the monitors, checking if Earth sent new instructions, setting the right speed frame (*coarse* or *vernier*), setting the appropriate coordinates reference frame (ISSACS, OBAS, OCAS, LEE tip, etc.), checking the motors status, removing the brakes, *etc*. In our simulator, the facilities are limited,

and CTS gets strong evidence of only one type of preliminary operation: adjusting views. Planning the path might be done completely in the mind of the astronaut without any recourse to the perspective view (although this is doubtful). So, in case of trouble, CTS will have to ask the user if he has planned the path.

At all times, CTS has an attention codelet that counts the time elapsed since the last user operation (manipulating Canadarm2, answering a question, adjusting a view, using the menus, etc.). That codelet may watch the time with respect to a standard duration or to the duration indicated by the Domain Expert. If that duration is elapsed, the codelet tries bringing that observation to consciousness. In addition, CTS starts another chronometer codelet at the beginning of a session, and a third one at the beginning of any manipulation exercise, this latter one being concerned with the duration of the exercise compared to what is expected from that learner. So, interventions may be driven by the passage of time.

A last word: coalitions always compete to be selected in Working Memory by the Attention mechanism; none is certain to win. I will often simply write that a coalition is broadcast (or "published", an alternate word), or even completely forego mentioning any going into WM to avoid annoying repetitive descriptions about the selection process. But there never is any guarantee about the coming to consciousness of any coalition of information, although the designer may have granted some information a high "natural" activation value to increase its likeliness of being published. It still depends on what else occupies WM at this point in time. In a nutshell, the decision process is very dynamic, very contextual and not at all deterministic.

### 5.4.1 Scenario 1: Missing step; CTS infers the cause and offers hints

This scenario emphasizes CTS deliberative capabilities involving all of the architecture's modules.

It begins when the initialization steps of a manipulation exercise are about to be completed. The initial position and configuration and the destination have been

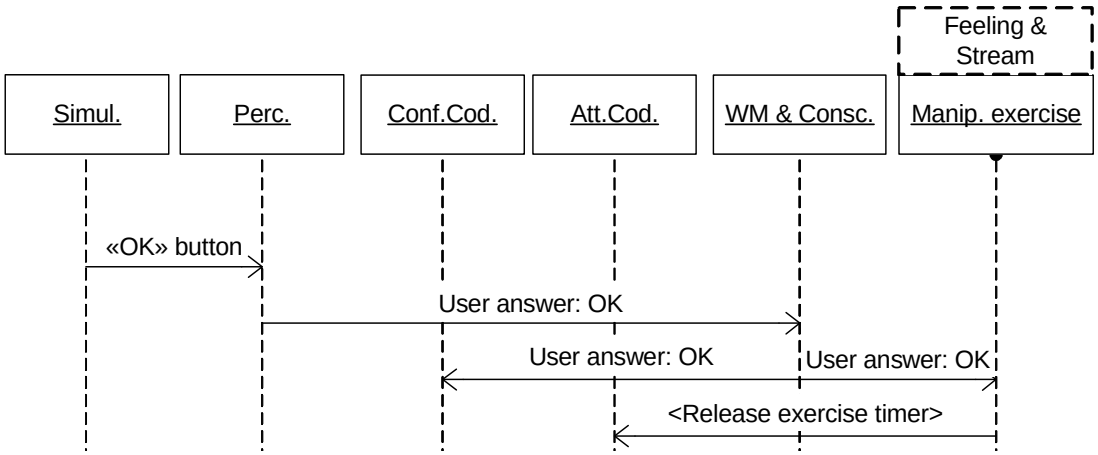



**Figure 22 Portion of the initialization of a Canadarm2 manipulation exercise** and noticing of inactivity by a "step timer" attention codelet.

shown, and the astronaut has clicked on the OK button, indicating he has memorized the task specifications and is ready to proceed (Figure 22). This «OK» was expected by an expectation codelet, and since it fulfills its expectation, it has no reason to advertise anything. That codelet will simply loose its activation and die away. As the last step of the stream that submits a manipulation exercise, an exercise timer attention codelet is released to see how close to the expected duration we have gotten.

A few moments pass, the astronaut selects one of Canadarm2 joints and starts rotating it (Figure 23). In itself, this rotation may or may not be a good choice, but this is beside the point: what is important here is that the astronaut did not adjust the views before making the manipulation. If the monitors are the only means available to the astronaut to see Canadarm2 and its environment, it may seem surprising that the scenario suggests that he does not adjust the views first.

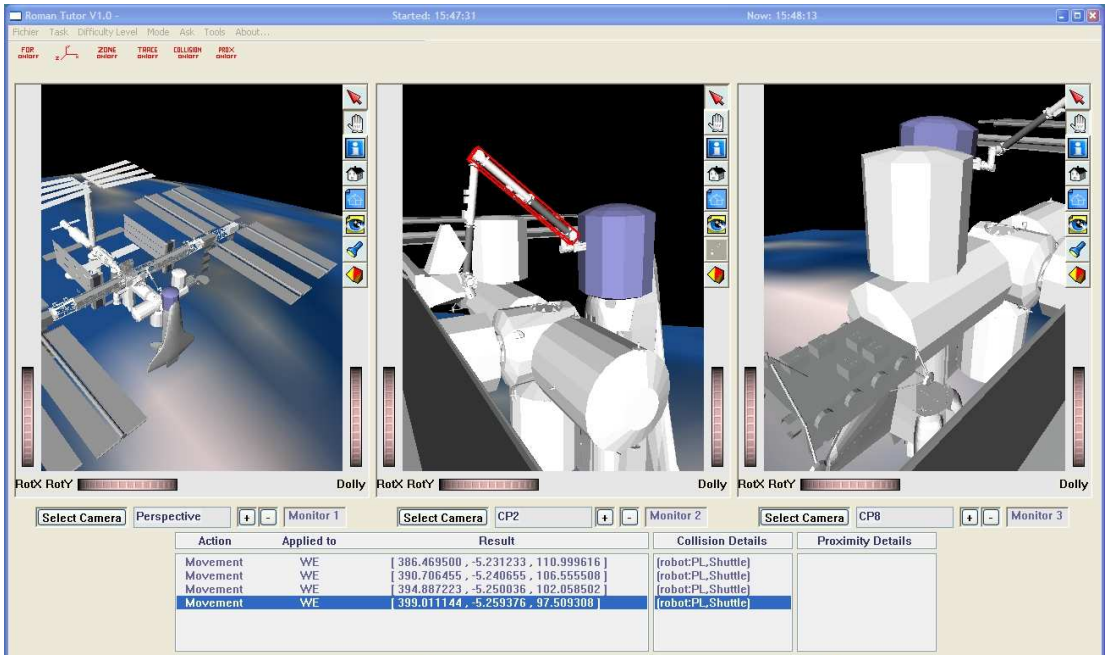



**Figure 23 The astronaut started moving Canadarm2 without adjusting the initial views.**

In fact, out of the three, there may be one view that is quite satisfactory, and makes the astronaut confortable to start the operation. However, procedure dictates that an *"optimal" combination of views* be established before any manipulation. It is very unlikely that all three monitors would offer the three best views from their default arrangement.

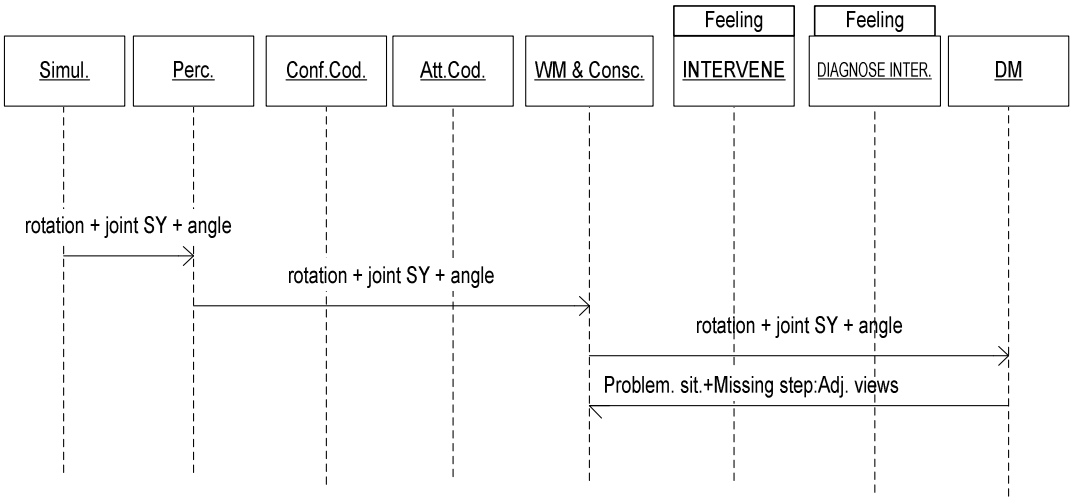



**Figure 24 Incorrect procedure followed by the astronaut.** The Domain Expert notices it and attempts to bring this to the attention of the Attention mechanism.

The Domain Expert was expecting a camera manipulation and has been made aware of a joint manipulation instead. It is able to determine that this refers to the next step in the correct procedure. So, it infers a missing step and signals it in a coalition of information codelets it sends into WM («Problematic situation: Missing step» + «Missing step: Adjusting views»).

When this new information arrives into WM, the Deliberation Arbiter notices it as describing a situation that warrants an intervention and attaches to the coalition the information codelet «Intervening: Proposed» (Figure 25). In effect, CTS is asking itself whether it should intervene; it will not just go out with its big boots and offer help to any user without thinking it through. A proposition for an intervention bears a relatively high importance (a high value); thus, the coalition containing this information codelet shows high probabilities that it will immediately get the Attention's attention (!). Subsequent broadcasting primes the Feeling of the need for an intervention; however, no action is taken now (no related Behavior in the network is fired) because all the preconditions are not met yet: causes have not yet been identified. This will come through a deliberation involving the modules that can supply a justification and approve (or oppose) intervening, *in extenso*: all the three sub-modules of the

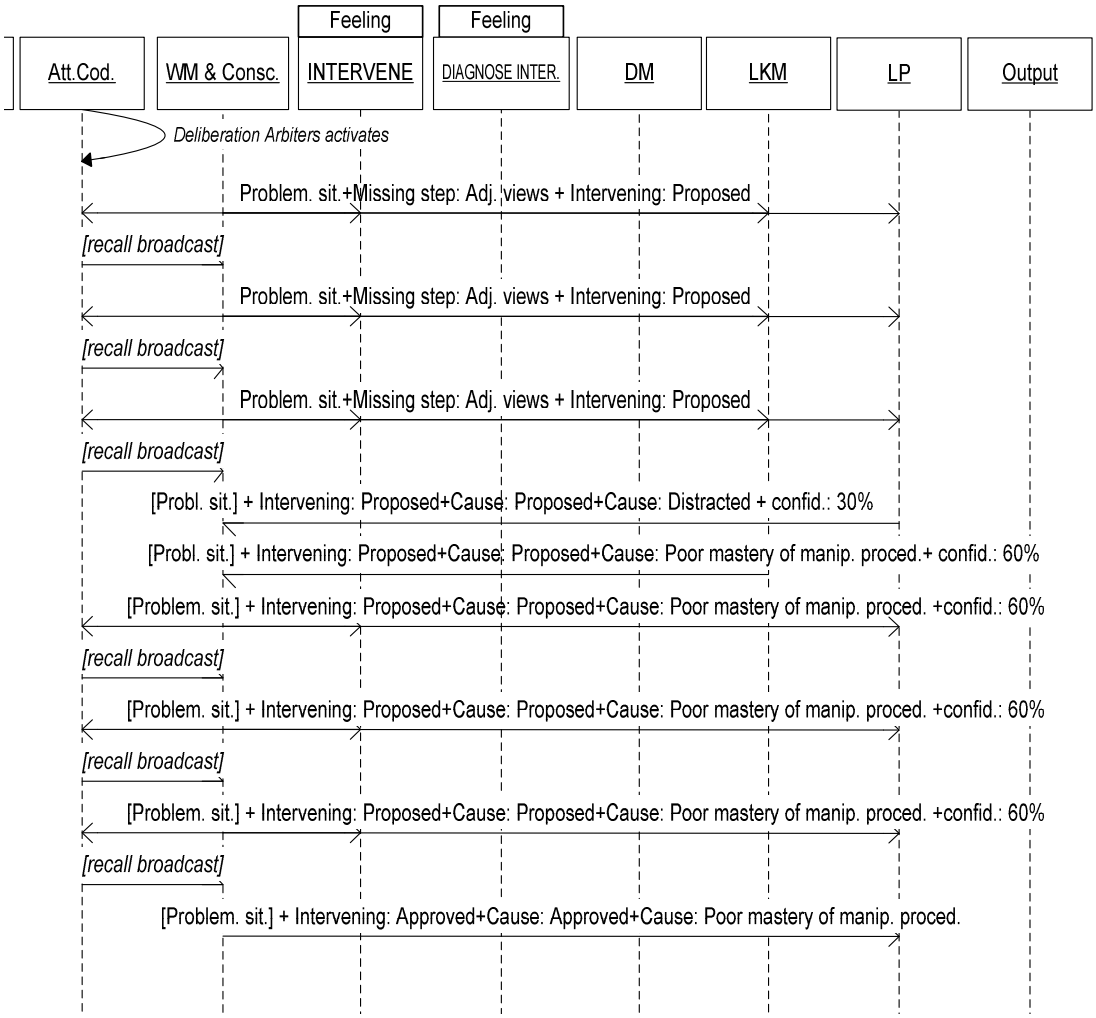



**Figure 25 Intervening is proposed.** As long as this is the most important coalition in WM, it gets published repetitively until the deliberation arbiter decides enough time has passed without any change (three cycles) or until causes are submitted. If more that one cause is offered, the Deliberation Arbiter chooses the most significant one and attaches it to the coalition under deliberation. If after a few cycles the cause proposed has not been opposed, the arbiter changes its status to «Approved».

Learner Model: the LKM (Learner Knowledge Model), the LPM (Learner Profile Model) and the LAM (Learner Affective Model). The LKM infers what the learner's knows from all the evidences it gathers from the broadcasts: what the learner does, how he performs, and what material he has been exposed to. The LPM knows what

intervention should be appropriate for the specific learner, and when. The LAM, aside from entertaining beliefs about how the learner is Feeling right now, keeps track of the impact of past interventions on the learner's motivation. It might intervene to indicate how the learner is feeling right now.

The second arrow in Figure 25 ("*Recall broadcast*") indicates that the Deliberation Arbiter reactivates the coalition just published in order to keep the context alive in WM and so that new information can be attached to it. As long as this coalition is the most important one in WM, it is broadcast repetitively until suggested causes do arrive into WM, or until the arbiter determines that enough time has passed without any change to the coalition. Here, the LKM and LPM modules have something to say about the last broadcast. After some time needed for their inference process (three cycles for both modules in our scenario), they offer their hypotheses about the cause of the overlooking: «Poor mastery of manipulation procedure» and «Distracted».

In designing the mechanism, we have elected the rule that only one cause can attach to the original coalition. When confronted with many possible causes (offered by different sources), the Deliberation Arbiter selects the most plausible cause in the current context. The plausibility of a cause is obtained by multiplying the cause's current valuation with the confidence on the hypothesis. Figure 10 gave the calculation for the cause «Poor mastery of manipulation steps» (0,53); the second cause («Distracted») computes to 0,39 (0,30 + (0,30 x 30%)). So, the poor mastery hypothesis is retained here. The Arbiter attaches that cause to the coalition, which adds new activation to it. This association mechanism implements Baars *convergence of information* phenomenon (Baars, 1997, p. 52). If this coalition is selected in WM and is broadcast (shown as the last arrow in Figure 25), the new aspects in the information should prompt new reactions in the *audience* (the modules hidden in the unconscious, in Baars' theater metaphor). Here, the Feeling for intervening gets more stimulation from it. Some module could also react and oppose the cause proposed. This would stimulate the module that got its hypothesis refused to submit a new cause, extending the deliberation process. An opposition could also aim plainly

at the idea of *intervening*. A number of reasons could justify such opposition in different scenarios:

- a module (possibly Learner Profile Module) estimates it would be damaging to intervene in the actual state of mind or affective state of the learner, as believed by the LAM;
- there is no cause (or no sufficient cause) for it;
- the support level chosen by the learner does not warrant intervening here.

An opposition to intervening simulates the experience we all have had of planning on intervening (for example, replying something nasty to someone) and just before the words went out of our mouth, refraining from doing so. It reflects James *ideomotor* theory (Baars, 1997b, chapter six).

In this simple scenario, nothing of the sort happens. The "standard" waiting time of five cycles is respected, during which the coalition is published repetitively with the proposed cause. The cause is not opposed, and neither is the idea of intervening. So, the Arbiter changes the status of both the cause and the proposition to intervene to «Approved». Then, the Arbiter knows it has completed its task.

The broadcast that ensues stimulates the «Intervening approved» State in the BN, a fundamental precondition to the whole «Hinting» stream (see Figure 26). The proposition of intervening is implicitly sustained by the LPM by not opposing it and by indicating the user's preferred way of interaction: hinting (Figure 27). We suppose the astronauts indicated sometime in the past a preference for hinting, or this indication has come with the default profile for astronauts.

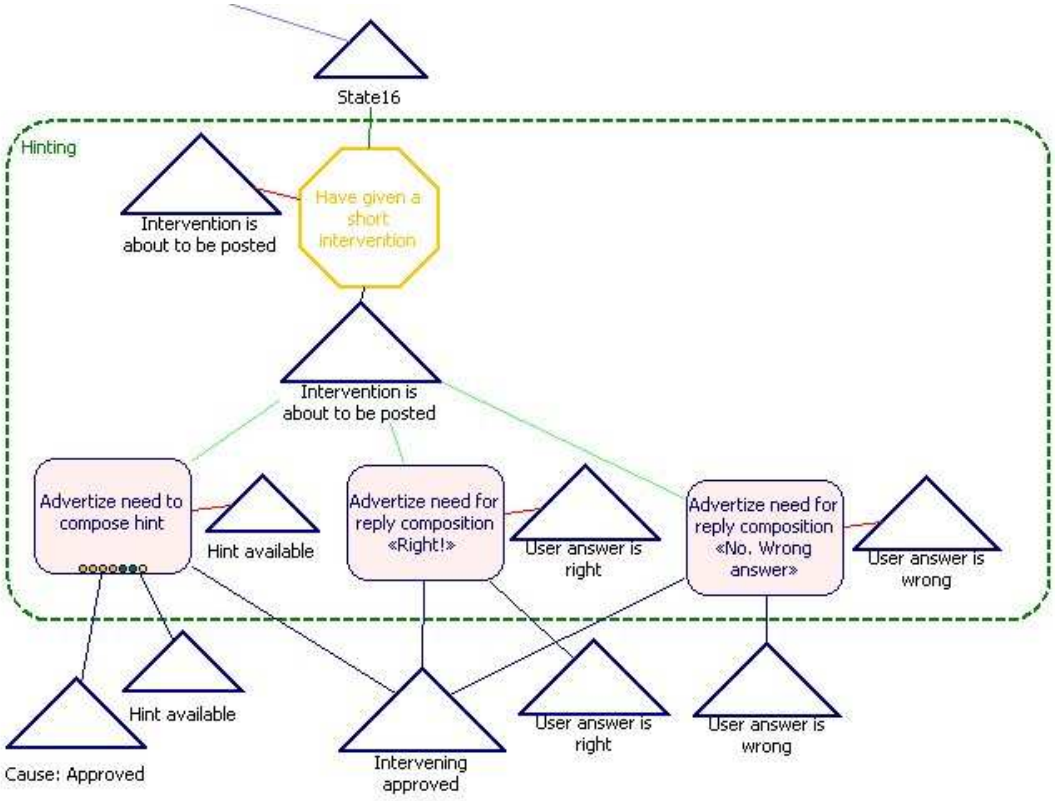



**Figure 26 The Hinting stream of the BN.** The context that renders hinting applicable is composed of three preconditions: «Intervening approved», «Cause approved» and «Hint available». Not showing here (above the stream) is the feeling it serves: «Need to intervene». When all the preconditions have been created, the behavior «Advertise need to compose hint» becomes *executable* and may fire if its activation is over the BN specified threshold, and is the most activated executable behavior in the BN. Little colored dots in the Behavior represent the codelets that implement the Behavior.

An attention codelet concerned by the hints given keeps note of the number of hints previously given in this intervention and replies with the hint number (level) to request: «Hint to give: 1». With this last information about how to interact with the astronaut, the Domain Expert is able to offer a contribution in the form of the text of a hint appropriate for the situation (based on the problem observed, the actual status of the manipulation, and the previous hint given): «Hint: Haven't you forgotten to do something?». This first-level, very general hint refers to the fact that the astronaut forgot to adjust the views before moving Canadarm2. Its content is not directly shown to the astronaut, but sent into WM.

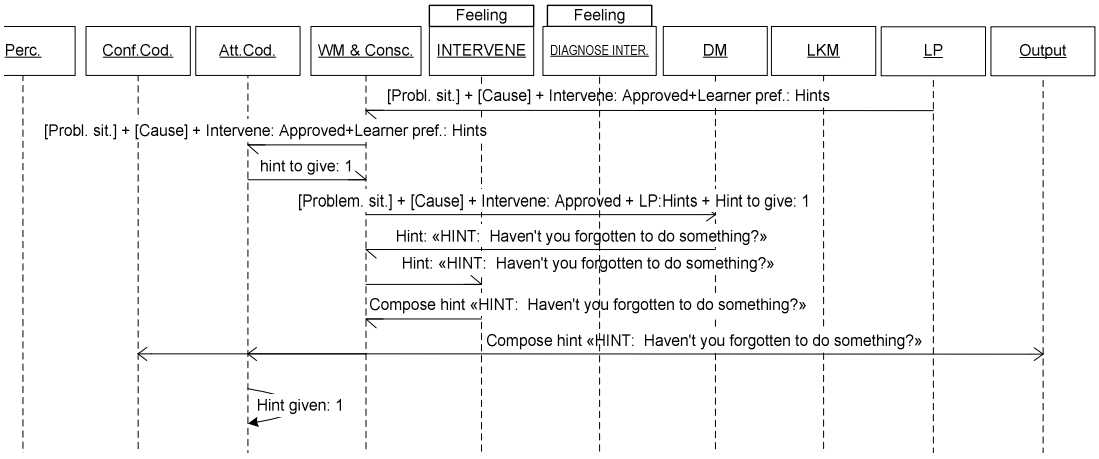



**Figure 27 Selection and presentation of a hint to the user.** An attention codelet keeps track of how many hints have been offered to the user.

When published, that content brings the final needed context in the Behavior Network for an intervention to start (see Figure 26): the activation of State «Hint available». When this information becomes available, the Behavior appearing to the left in the *Hinting* stream can send away its codelets, requesting that a hint be put into shape[22] and shown in a window on screen. A specialized process will later take care of building the window that will appear on the computer screen. The astronaut will see a message appear in the simulator «Hint: Haven't you forgotten something?».

At this point, CTS has begun an adapted interaction with the learner. It will continue with further hinting, progressively more specific and instructionalist, until the astronaut corrects the situation.

[22] The name of the Behavior, as it appears in the diagram, is generic. It allows any appropriate resource (a composition codelet or a full composition stream) to compose the textual interventions from the available bits and pieces. For now, the role is assumed by a composition codelet that simply takes the hint available and adds a «OK» button before transferring the result to the Output Buffer.

### 5.4.2 Scenario 2: Inactivity. CTS does not see the cause and offers general help.

This scenario emphasizes CTS' “unconscious” deliberative capabilities implemented in the Behavior Network. My explanations in this scenario build upon those from the previous scenario; I will mostly add only the novelties.

This scenario begins with CTS noticing (as revealed by the step timer attention codelet) an undue elapsed time since the last user action (Figure 28). In itself, that inactivity is not indicative of a problem: the astronaut may be planning the next move. This analysis can be surprisingly complex and the astronaut may simply need more time to think. Or it may be that the astronaut is unsure of the next step he should pursue, a problem that would need caring for. In any case, if published, that information about inactivity stimulates the Feeling in the Behavior Network about the need to intervene, which starts pushing energy into all the streams connected to it (through their top Goal node). But the complete necessary context has to be present for any action to be initiated by the BN.

The first time intervening "crosses CTS' mind", if I may say so, it is rejected by the LPM (Learner Profile Module). Its inferences determine that, based on the user profile and on the fact that no strong cause has been proposed, not enough time has yet been allotted. In fact, the idea of intervening has come to consciousness either before the various information sources had the time to react, or because the modules simply have no explaining cause to offer. In a traditional setup, the human tutor can see what the astronaut is doing, he can see his face, and he listens to the verbal reports the astronauts has to give about what he is doing and of what mental operation he is accomplishing. So the human tutor gets pretty good clues about what is going on. CTS cannot (yet) rely on such information sources (works are under way for visual interpretation of facial expressions and biosignals). It only has its beliefs based on past evidences to try inferring what causes the inactivity. If CTS' Learner Model can suggest a cause (for instance, erroneous or lack of knowledge, fatigue, or distraction, as in Figure 25), then CTS evaluates by an internal debate whether to intervene.

In this second scenario, not a single cause shows up in Working Memory, even the second time inactivity is signaled. But no opposition comes up either. So, after publishing again the same information a few times, the Deliberation Arbiter

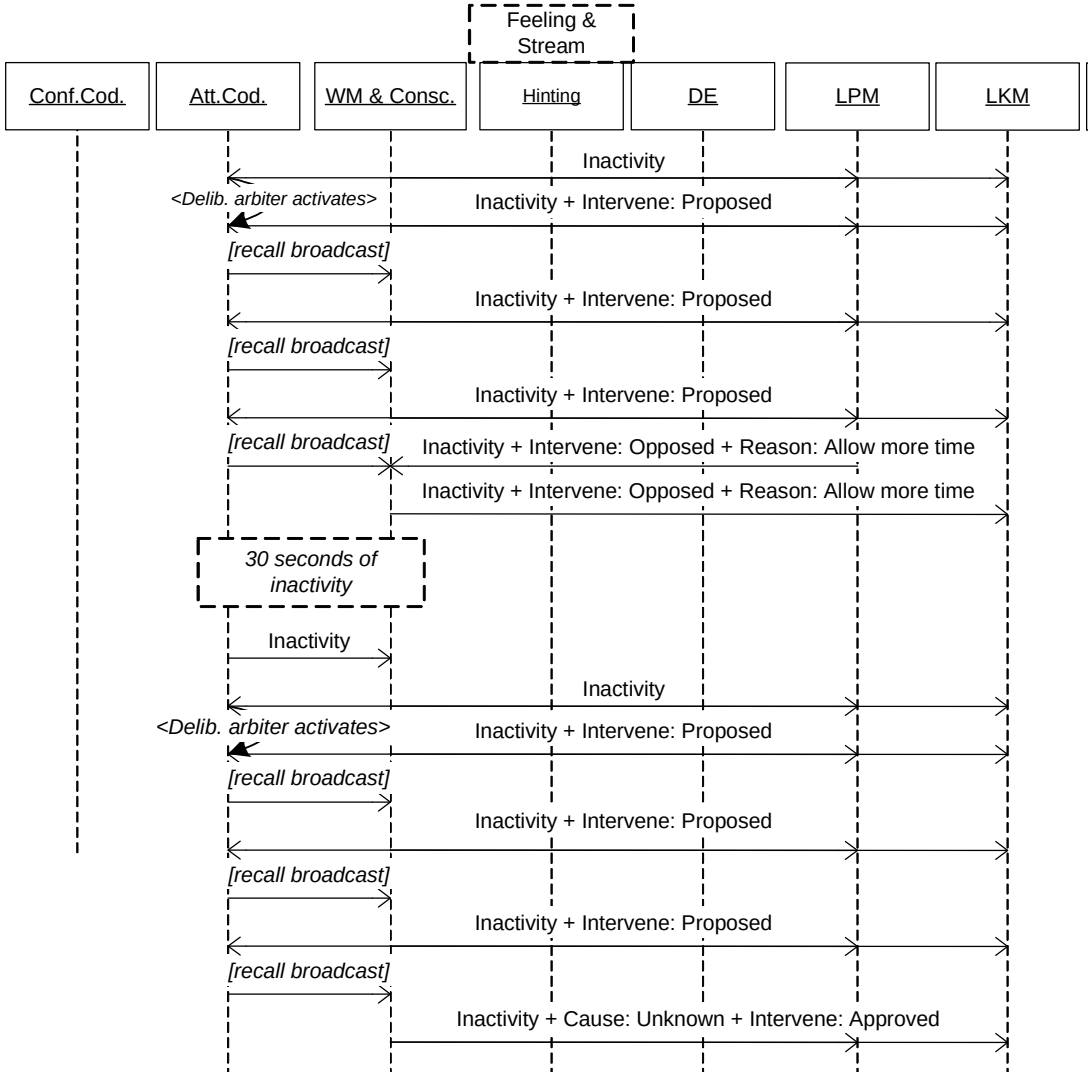



**Figure 28 CTS deliberates about intervening and about the cause of user's inactivity.** The conscious broadcast about the situation («Inactivity») brings the whole system to awareness but no cause is suggested. Nevertheless, some feelings get stimulated by this situation of inactivity, and readies the BN to react eventually. Just at the end of the standard deliberation duration, the LPM indicates its opposition to intervening without an explicit cause at this point in time: according to its beliefs about the astronaut, he probably just needs more time to think. After another waiting period, the same inactivity is signaled, but this time no one opposes an intervention.

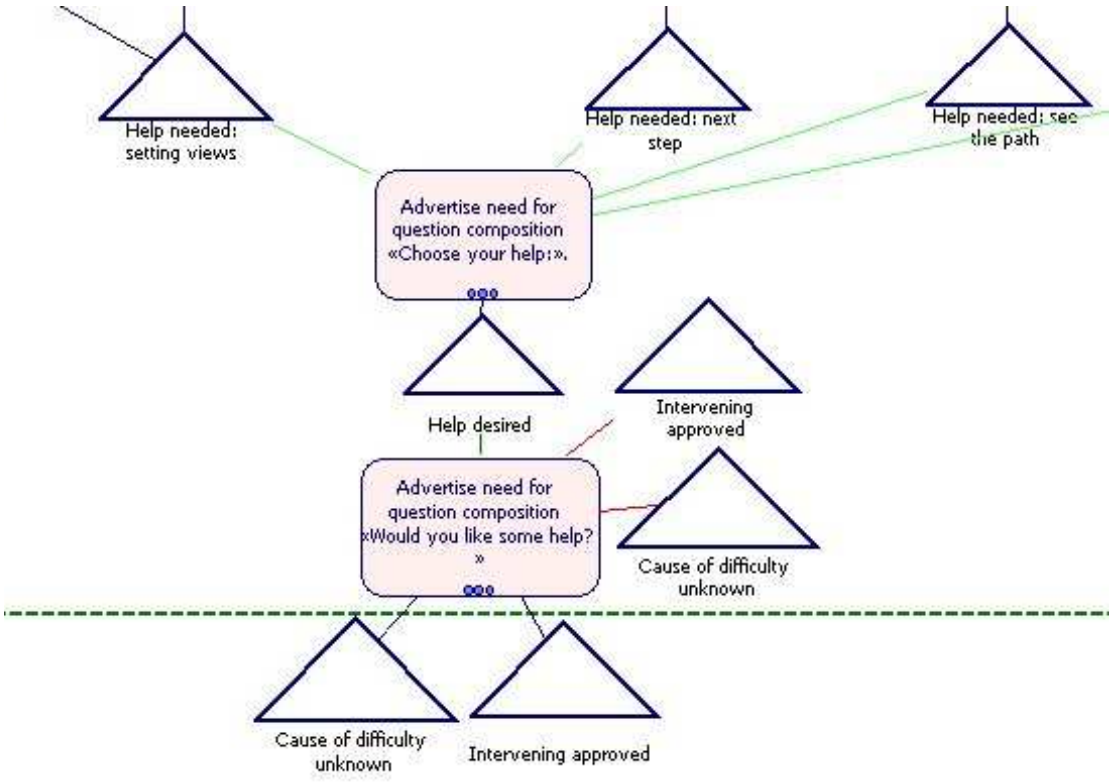



**Figure 29 Beginning of the «Give general help» stream.** It shares the precondition «Intervening approved» with the «Hinting» stream, but has the unique precondition «Cause of difficulty unknown». This stream gets involved when no module could suggest a cause to the situation. In this situation, CTS needs to interact with the user to find how it can help in what it believes to be a problematic situation.

closes the deliberation, changes the status of «Intervene: Proposed» to «Approved» and attaches an information codelet stating «Cause: Unknown». That broadcast stimulates the «Intervening approved» State in the BN, a fundamental precondition to the «Hinting» stream, but also to the «General help» stream (see Figure 29).

The States represent the context and will orient between many different ways of satisfying the Feeling of the need to intervene (CTS general intention). Indeed, that Feeling is connected to three streams that can satisfy it («Hinting», «General help» and «Interactive diagnosis»; see Figure 30), and it supplies them all with "top-down" energy. Thus, the Feeling orients generally what the agent will do (here: mak-

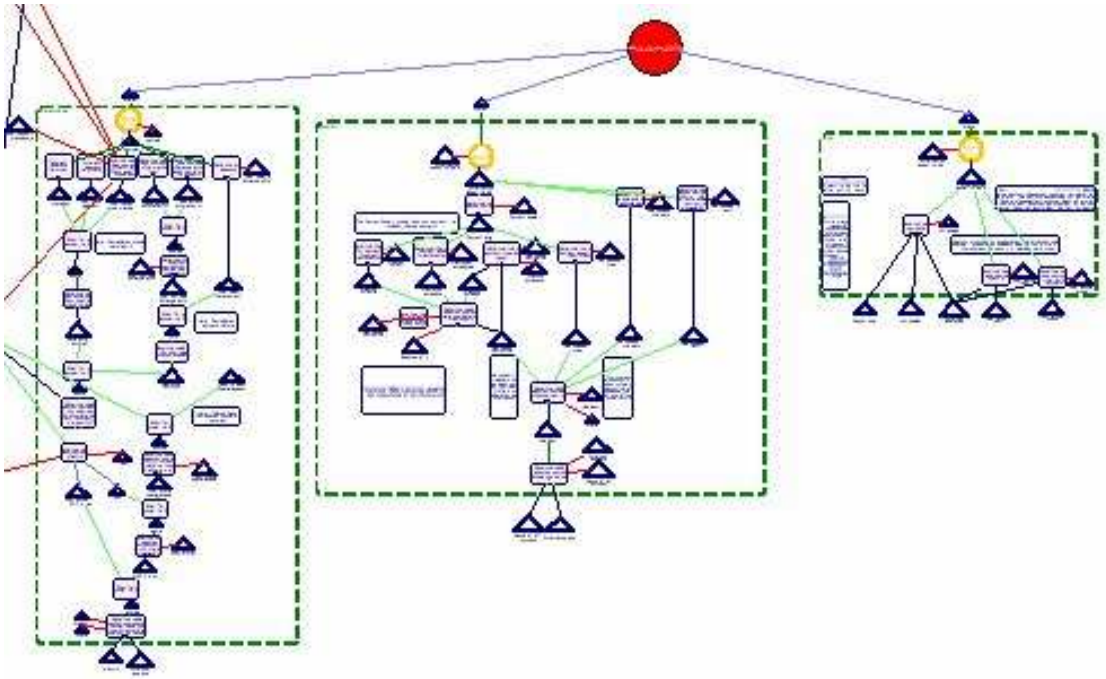


**Figure 30 Portion of the Behavior Network concerned with tutorial interventions.** The red dot at the top represents the *feeling* of the need to intervene. When stimulated, it feeds in energy the three separate streams that are connected to it (they serve its satisfaction). This illustrates that there is competition in the BN, even under the same feeling, all this serving the goal of adapting to the user.

ing an intervention about the problem), and the context decides on the precise way that will roll out. In the present situation (cause unknown), the «General help» stream is the one that gets all its preconditions activated.

For the coming explanations about the chain of actions in the BN, I will forego describing every time the loops through WM and will not mention every time that the hint or question needs to win the competition for consciousness, or that the question needs to have been joined by the appropriate answer choices and buttons before getting thrown into the Output Buffer, and so on. However, I insist that these shorter explanations should never be interpreted as though CTS behavior is deterministic, even if the BN is deterministic by its States (but not by the dynamics of energy flows) Choices through the BN depend on the combination of current and past events and

on what impact learning has had in the past. The combination of the variably activated Feelings, evolving links strengths and nodes base activation creates what can hardly be called deterministic There is always competition in WM for the most relevant information, and it is the winner that decides of the fate of the States in the BN. Moreover, a stream can at any time get interrupted by something more important appearing in WM.

The first Behavior in the «General help» stream politely and respectfully asks whether the astronaut would like some help (Figure 29). A «No» would stop the show here and now; the States and Behaviors that have been stimulated would slowly decay away (temporarily leaving a predisposition for intervening[23]). At that point, the context «Cause of difficulty unknown» and «Intervening approved» is no longer required, so the BN turns these States off (they are part of the *delete list* of that Behavior node; delete links appear in red).

Let's say the astronaut desires help and answered «Yes». That «Yes» stimulates the «Help desired» State, which creates the context for saying «Choose your help» (Figure 29). That question is generated by the codelets that support this Behavior. They are *Behavior codelets* that are capable of using variable content to adapt the questions to the specific situation. Sometimes, the questions are static (for example: «Have you established your milestones?»), sometimes they refer to the precise situation («What would you say is the structure actually nearest to Canadarm?», shown in Figure 31) and rely on options returned by the Domain Expert to propose adapted answer choices. Also part of the Behavior is the confirmation codelet that gets launched to verify that the astronaut responds to the question.

---

[23] Not "turning off" instantaneously leaves some sort of trace of what happened recently. If the need for general help was asserted again in the coming moments, that stream would fire sooner because it still bears some activation.

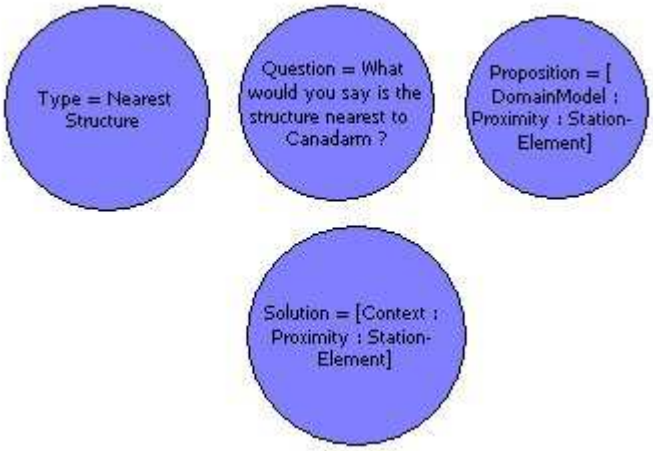



**Figure 31 Behavior codelets that implement the question proposed by the Behavior node: «What would you say is the structure actually nearest to Canadarm?».** The Domain Expert returns five names of modules to this specific request, which become the adapted answer choices presented to the user.

From there, the stream splits between many paths. The choice the astronaut makes in the proposed menu will decide which State turns on and which Behavior sees its precondition(s) come alive. Let us assume that the astronaut wishes help on setting the views. What the designer of this BN has deemed appropriate is asking then whether the astronaut has established the milestones for his path; in other words, if he has planned the path he intends to impel to the Arm. Choices are offered through information codelets that inform the composition codelet to use the propositions: «Yes», «No», and «What are milestones?» The last option triggers the Behavior that advertises the need for information about the concept of milestones, to which the Domain Expert will react, sending a text that the codelets will be able to use to inform the astronaut. Finally, when the proper Behavior has constructed the adapted material, it is assembled by the composition codelet that deposits the message, complete with the «OK» button, in the Output Buffer. An intervention has been completed.

# Chapter 6

# COMPARING CTS WITH OTHER POPULAR ARCHITECTURES

## 6.1 COMPARING CTS WITH A POPULAR AGENT ARCHITECTURE: BDI

When considering real-world applications that deal with complexity, change and uncertainty, conventional approaches falter (Georgeff and Ingrand, 1989). They are mostly designed for static worlds with perfect information. Talking in a 1999 panel (Georgeff, Pell *et al*., 1999), Georgeff reaffirmed his belief that, contrasting with conventional approaches, software agents, in particular BDI agents, provide the necessary elements to cope with the characteristics of our world. BDI aims at allowing a *resource-bound* agent to deal with an uncertain situation in a timely fashion, before the world has changed again. Jiang and Vidal (2006) explain that BDI has shown to be a very successful architecture for several reasons: first, it has widely

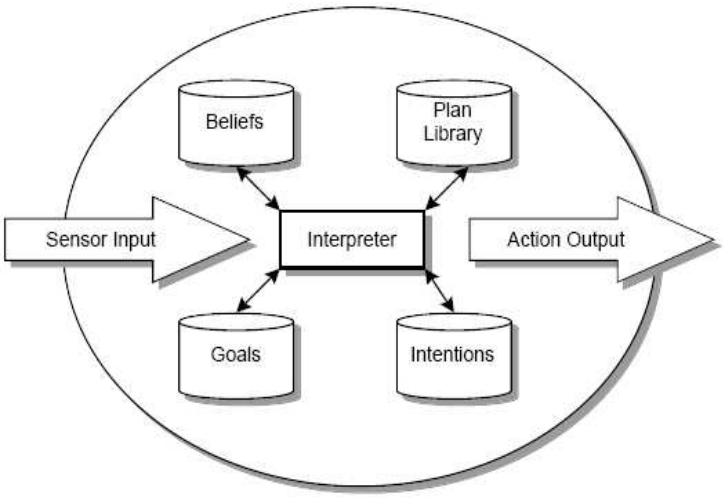



**Figure 32 The BDI architecture.** *Source: d'Inverno, Kinny, Luck, Michael, et al. (1997).*

accepted philosophical roots; second, there are logical frameworks for modeling and reasoning about BDI agents; third, there is a considerable set of software systems which employ the architecture's concepts.

Strictly speaking, there is no single software architecture that represents BDI since BDI describes high-level structures (Figure 32), constraints and mechanisms from which one can derive an architecture. The fundamental ideas include a set of beliefs about the world, as set of desires, which are possible goals about reacting and acting on these beliefs, a library of plans that may be used in reaching the selected goals, and intentions, organized in an intention structure. As originally proposed by Bratman *et al.* (1988), a practical-reasoning system inspired by the BDI principles sees these structures manipulated by various mechanisms, among which: a *Means-End Reasoner*, an *Opportunity Analyzer*, a *Filtering Process*, and a *Deliberation Process*.

PRS[24] (Figure 33) has been the first architecture implementing BDI concepts, and has been the foundation of numerous subsequent works. It implements BDI

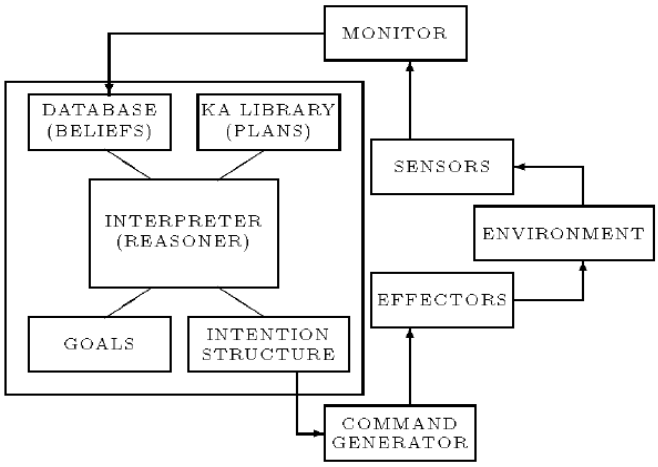



**Figure 33 An agent implementing BDI principles: PRS.** *Source: Georgeff and Ingrand, 1989*.

[24] My description of PRS relies heavily on Georgeff and Ingrand's (1989) paper.

ideas with (a) a database containing current beliefs of the agent and facts about the world; (b) a set of current goals to be realized; (c) a set of plans, or declarative procedure specifications (called Knowledge Areas, or KAs) describing how certain sequences of actions and tests may be performed to achieve given goals or to react to particular situations; (d) an intention structure containing those plans that have been chosen for (eventual) execution. I will describe in the following paragraphs how these structures, and the mechanism that manipulate them, have been implemented in PRS.

The agent interacts with its environment, including other systems, through its database, or more precisely, through monitoring mechanism that evaluate if there are changes in what is gathered by the sensors, and that feed the database. This structure, originally populated with static information about the domain of application, acquires new beliefs through its *belief-revision function* that responds to change in the environment. The agent selects goals about these beliefs, forming its *desires,* from which an *intended* goal is selected. The various ways intentions can be carried out are represented in KAs. A KA has a *body*, which describes the steps of the procedure, and an *invocation condition*, which specifies under what situations the KA is useful (applicable). BDI's filtering and filter-override mechanisms, which evaluate the options, are implemented as a *metalevel*, in special Knowledge Areas (KA) structures. Although the states descriptions that give the necessary preconditions to KAs are written in first-order logic, they can serve the unification process at the metalevel as well as at that of regular KAs. These metalevel states describe internal system states, typically the beliefs, goals and intentions of the system about its own functioning, as well as other important processing information.

The goals in PRS may be of various natures: goals of achievement, goals of maintenance, and goals to test for given conditions. And just as with state descriptions, goal descriptions may characterize the internal behavior of the system (in *metalevel goal descriptions*). Goals create constraints on what options need be seriously considered. They influence what beliefs are taken into consideration for the reasoning, and they give plans some resistance to reconsideration or abandon. That

is necessary for a certain level of consistency in the sequence of actions the agent is taking. Indeed, PRS could be subject to erratic jumping for many reasons, including its adoption of partial planning, and it continuous scan of the environment.

Partial planning brings the benefit of better reactivity, allowing the imposition of temporal constraints. It sidesteps the need to wait for a thoroughly worked-out and validated plan before getting into action. It helps avoid having to abandon plans, which might occur frequently if plans were very specific.

When a goal has been selected and PRS has committed to a plan (selected one as the most appropriate), it does not look back, unless significant changes happen in the environment. And PRS is very vigilant, with the interpreter continuously attempting to match KAs with any newly acquired beliefs or goals. The system is able to notice newly applicable KAs after every primitive action it takes. If estimated necessary, the agent reassesses its current intentions, and plans that were dismissed become subject to reconsideration, even though the new options are not means to any already intended end. PRS plans are interruptible, and can completely change its focus towards new goals when the situation warrants it.

Bratman's Opportunity Analyser is the component that keeps the agent open to changes in the environment and proposes new options to pursue the existing plan and cope with new perceptions. It exists in PRS as metalevel KAs, just as does the idea of the filter-override mechanism. But a *filter override mechanism* run in parallel allows maintaining the equilibrium between the stability of plans (keeping the focus of the reasoning) and the necessary revocability, given that plans were selected on the basis of incomplete information (the agent does not live in an idealized world of perfect information and total predictability). The override mechanism encodes the agent's sensitivities to problems and opportunities in its environment. An option that does not survive the compatibility filter may still be subject to consideration if it triggers a filter override. The surviving options are put into a deliberation process that weights them against one another and, ultimately, the deliberation process produces new intentions. So, the addition of appropriate metalevel KAs enables the system to make more informed choices (at the cost of longer decision times).

There are numerous parallels relating CTS to PRS. In fact, although CTS has not been conceived with BDI in mind, it really incorporates all of its components and principles, as I will briefly summarize below.

What CTS knows about the world, its beliefs in BDI parlance, are contained in various constructs: States, the Domain Expert, the Learner Model, *etc*. The Domain Expert contains static knowledge about the world (domain facts and procedures), and status of current operation; the Learner Model contains static and dynamic beliefs about the learner (facts and inferences); the *States* in its Behavior Network contain transient beliefs about the world and about the internal operations of the agent; mid- and long-term memories are also naturally concerned. CTS' Behavior Network bears much resemblance with PRS Plans Library: Knowledge Areas exist here as Behaviors, and plans correspond to sequences that embody partial planning with intermediate goals and partial specification with many generic behaviors that get specified through deliberation. Goals in CTS may be of various types, with interests in external actions as well as internal adaptation and operations. Behavior nodes partially depend on preconditions ("invocation conditions") to fire. The Feeling nodes represent the global goals the agent may entertain, and incorporate the idea of desires. Indeed, Feelings that are stimulated after an Access Consciousness publication show the desires of the agent, the goals that may more or less be appropriate to the situation (indicated by the activation levels). The Behavior that gets selected for action indicates which sequence and goal the agent has elected, that is, its intentions (held within an intention structure: a stream).

The Behavior Network has parameters that allow balancing the agent's sensitivity to the environment and its stubbornness (how much its Goal nodes drive its global behavior). The firing threshold for Behaviors nodes allows more or less time for the energy to reach the ending nodes of longer sequences, giving more or less time for the agent to "think" through the options. When a Behavior fires, it pushes its energy forward to the next nodes in the plan, progressively increasing the commitment of the agent to a plan. Just as BDI specifies, sequences (plans) in CTS may be

interrupted. A State may "turn off" as the result of a change in the environment; it may also happen following the arrival of an information that turns on a State or strongly stimulates another Feeling that pours a great amount of energy into another branch of a plan. CTS' Behavior Network is always listening to new perceptions, making implicit means-ends analysis. The base-level activation of Behavior nodes, modified by one of CTS' learning mechanisms, indicates how much the nodes are theoretically apt at treating a situation and deserve to be selected.

Some more connections exist. CTS Attention mechanism does a job equivalent to the compatibility filters proposed by the BDI theory. Its attention and metacognition codelets serve some of the purposes of the filter-override mechanism and all of BDI's metalevel. And both have deliberative capabilities.

According to Jiang and Vidal (2006), the main problems about BDI architectures are about finding how to efficiently implement these functions and how to reach the balance between being committed to and over-committed to one’s intentions. And, as they stand, BDI architectures ignore the influence of emotions in decision-making (Jiang and Vidal, 2006). The first two criticisms apply just as well to CTS, at least until more time is devoted to the elaboration of a clear methodology and creation of well-defined rules for the instantiation of CTS to a new domain. However, CTS exhibits here a supplemental feature: its ability to deal with feelings and emotions. Emotions have not been implemented as of now, but they are part of the conceptual model, and have already been implemented in IDA (and redesigned for LIDA), CTS' mother. They modulate learning and have influence throughout the architecture, in many aspects (see Franklin and McCauley, 2004).

CTS adds a few other features to the BDI framework. First, the Feeling nodes serve in granting the agent with a personality, which allows it to be more or less sensitive to some events, and react with a strength corresponding to such personality. Second, its Working Memory is central to additional capabilities: CTS deliberative capabilities are stronger, allowing the building of plans on-the-fly; the meeting of codelets in Working Memory permits both the learning of regularities, and the emergence of creative solutions. If a new coalition has merit in the situation, it will be se-

lected by the Attention mechanism and published by the Access Consciousness for the whole system to use and process. These possibilities are alien to the original BDI ideas.

So, with CTS, researchers may reap the advantages of BDI with its well known concepts, and explore new applications that would be hard to take on with only the native BDI theory.

## 6.2 COMPARISON OF CTS WITH A COGNITIVE ARCHITECTURE: ACT-R

Understanding what is happening in the head of a human being is a complex task that needs to be addressed if one is hoping to have his system provide the best support to the user. Reactive architectures are limited in this respect. There is a lot going on at the same time in the user's mind: recognizing symbols, memorizing new information, processing the syntax of instructions, reasoning about events, and much more. Understanding and following the evolution of each aspect is hard enough, but having them show a coherent processing that corresponds to the real user is more than a challenge. This has brought Newell to suggest constructing cognitive architectures such as SOAR (Newell, 1990).

ACT-R is such a cognitive architecture. It describes and implements cognition at the grain size of laboratory research, and is still able to put the pieces together in a model of complex cognition (Anderson, 1993, 2004). It provides a potential bridge between basic cognitive psychology and education (Anderson and Gluck, 2001). Basic assumptions of ACT theories (ACT* and its evolution, ACT-R[25]) are that human cognition emerges through an interaction between a procedural memory and a declarative memory, and it unfolds as a sequence of so-called *production rules*. New *chunks* of knowledge are added to declarative memory when goals are achieved.

[25] I am refering to version 5.0 of ACT-R in my comparison.

For instance, if a child sets the goal to add 4 to 3, counts up, and finds 7 as the answer, the goal and the answer are assembled in a knowledge chunk and stored. The goal can later be retrieved with the associated answer. Chunks may also be formed from a perception of the environment. Although ACT-R makes use of a subsymbolic level that models learning and chunks availability following learning and practice, the architecture treats cognition as a symbolic system. This and the subsymbolic level do not model the actual neural learning process; they rather model their effects by a set of equations that characterize these processes. New productions rules can be formed by compiling solutions found in declarative memory. Production rules can change the goal state.

The performance of the architecture is both parallel-based and serial-based. Many *modules* operate in parallel, with serial internal operations in each. The perceptual layer contains a number of independent modules capable of running in parallel with cognition and with each other. However, each of these modules is doing only

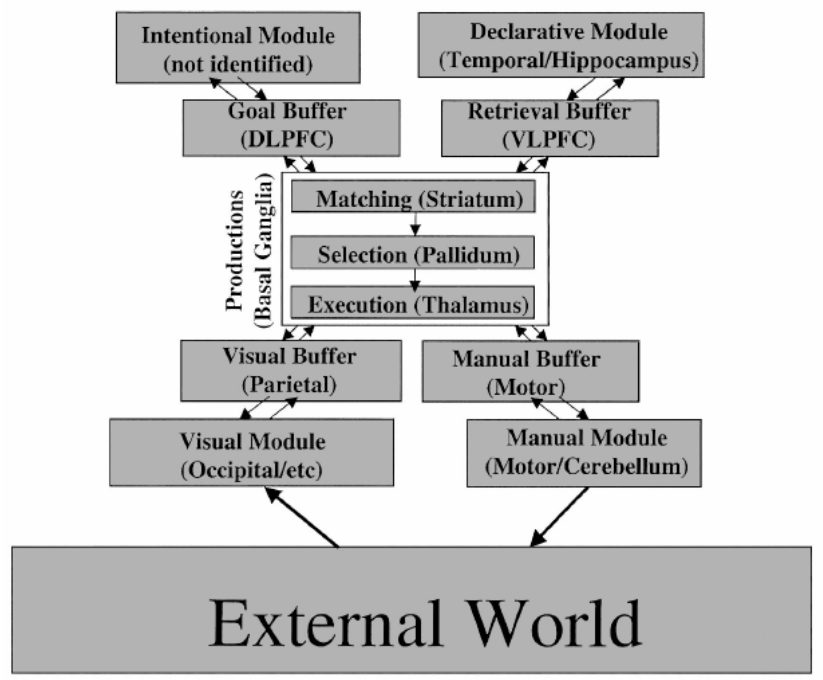



**Figure 34 ACT-R's architecture.** *Source: Anderson et al., 2004*

one thing at a time. And just as activation levels play an essential role in chunk selection, the next production rule is selected on the basis of the highest utility among all those that apply to the situation (Lebiere *et al*., 2004). The *utility* is a noisy estimate of the probability that if this production is chosen, the current goal will be achieved. The highest valued production is always selected, but on some trials, one might randomly be more highly valued than another.

Coordination in the behavior of these modules is achieved through a central production system. This central production system is not sensitive to most of the internal activity of these modules; it only responds to the limited amount of information that is deposited in the buffers of these modules. It parallels the facts that real people are not aware of all the information in their visual field but only the object they are currently attending to; they are not aware of all the information in long-term memory but only the fact currently retrieved. An example of this limited information communication is that the whole memory of the agent is not available to the rules, only of the content of the retrieval buffer, which holds information retrieved from long-term declarative memory.

The manual buffer is responsible for control of the hands. One of the visual buffers, associated with the dorsal “where” path of the visual system, keeps track of locations, while the other, associated with the ventral “what” system, keeps track of visual objects and their identity. The contents of these buffers can be determined by rather elaborate systems within the modules. For instance, the contents of the visual buffers represent the products of complex processes of the visual perception and attention systems. The goal buffer keeps track of one’s internal state in solving a problem. This is a special buffer that has links to declarative memory, making some nodes more accessible than others (Lovett, Reder and Lebiere, 1999), and which content primarily drives ACT-R. The current goal contains the information in the focus of attention.

A final word about consciousness in ACT-R framework. Lovett, Reder and Lebiere (1999) and Taatgen (2006) attempt at clarifying how ACT-R may incorporate consciousness, and they locate the bridge in declarative memory. The declarative

memory's nodes above threshold may be considered as *accessible* to conscious awareness -- although only what is retrieved and put in the retrieval buffer is *viewable* for action selection. That is, the system can be considered “aware” of the contents of all these buffers, i.e., it is aware of the currently attended visual stimulus, it is aware of the current action that is being taken, it is aware of the current goal, and the currently active fact in declarative memory. However, in the opinion of Gray, Schoelles and Myers (2003), its "consciousness capability" is limited since it cannot model the difference between the implicit, unconscious use of a strategy or acting on instinct, and the result of the act only later becoming conscious.

From this description, one can establish many parallels between ACT-R's and CTS's architectures. In fact, there are a lot of similarities: a procedural memory, a semantic memory, buffers, learning (both procedural and semantic), a rule-based functioning that considers the context, multiple specialized modules with internal seriality and with an independence that allows them to run in parallel, competition in the action selection that takes into account past utility of resources. There is also both symbolic and sub-symbolic processing, and a cognitive cycle. I could draw a detailed comparison of the two architectures. In the following subsections I will mostly limit myself to pointing out major differences.

### 6.2.1 Comparison of the cognitive cycles

Although the two systems’ cognitive cycles differ considerably and thus one could suppose that this might incur significant differences in the agents' behavioral responses, I could identify only one major consequence directly related to the cycles: the interaction of information coming from the various sources (modules) in working memory. According to Anderson (Anderson *et al.*, 2004), the cognitive cycle in ACT-R starts at the point in which the buffers hold representations determined by the external world and internal modules. Chunks in these buffers are recognized, a production fires, and the buffers are then updated for another cycle. Thus, a production

rule in ACT–R corresponds to a specification of a cycle from the cortex, to the basal ganglia, and back again.

In CTS, the cycle is more detailed, with eight steps, as described in section 4.2. One major difference resulting from the cognitive cycles stands in the place and time that is allowed for information to meet and interact and compete in Working Memory before the winning structure is selected and fed to modules. ACT-R's cycle and architecture do not allow for such natural, unsupervised interaction that could lead to spontaneous discovery of new regularities or solutions.

### 6.2.2 Buffers *vs.* Working Memory

ACT-R's buffers hold information that goes back and forth between the central production system and modules. Buffers could be thought of as holding the information in the focus of attention. They are checked at every cycle by the production rules matching system to determine what rule is most appropriate to the context. CTS also has buffers, but they only play their traditional role of temporary information receptacles for transiting information (*i.e*. Sensory Buffer, the *focus*, which holds information destined to declarative memories, and any buffer peripheral modules may need). CTS' Working Memory would be a better related structure to ACT-R's buffers.

Although it may be tempting to correlate buffers in ACT-R with its working memory (I certainly am tempted), this is an incorrect assumption according to Chuderski *et al.* (2006). The authors explain that working memory in ACT-R may be defined in two ways: as a subset of highly active elements of declarative memory or as a process of spreading source activation (i.e. attentional resource) from current goal to declarative elements strongly linked with that goal. In comparison, while CTS' Working Memory holds information returned by declarative memories related to the current context, as is the case in ACT-R, it also welcomes information from other processes. In that way, it can as well be thought of a corresponding to some of ACT-R's buffers. Contrary to these buffers, CTS' Working Memory it is not constrained, neither as to the type of information it may contain, nor as to the quantity of

information (as was explained in section 4.1.5, whereas IDA *presents the same constraints as ACT-R in this respect, LIDA is somewhat relieved from them*). CTS' WM can be described as a physical meeting place for all the information returned by all modules and of value in the context; "meeting place" is quite an apt description here. It allows a very interesting phenomenon to happen: creation of spontaneous associations between information codelets, eventually leading to new coalitions (kind of equivalent to chunks); it also permits association of related coalitions during deliberations. The unconstrained interaction of the information, which is impossible in ACT-R, enables the learning of new regularities in the environment. It also allows a powerful and rich voting mechanism through reinforcement and inhibition between the information codelets. It could also explain a part of creativity as the unforeseen association of ideas.

### 6.2.3 Representation of the context

The variety of buffers in ACT-R, doubled by the goal stack, creates a multi-aspectual context. ACT-R can involve a Learner Model just as CTS does to keep track of the learner as part of the context of decision. In addition, ACT-R keeps an implicit trace of the context in the activation level of its memory chunks and in the sub-symbolic equations that compute the utility of the rules.

CTS has similar uses of activation, but in more places. Whereas ACT-R shows activation only in its declarative memory, CTS maintains its representation of the environment as the activation of States, the activation of Behavior and Feeling/Desire nodes, as fading coalitions in its Working Memory. I would say that CTS offers a slightly richer representation of the context. When we add the capacity of CTS' architecture to take into consideration user's affective state and the agent's own affective state, then we add a dimension that currently overwhelms ACT-R's architecture. Anderson *et al.* (2004) are taking into consideration the idea of adding multiple goal structures.

### 6.2.4 Learner's goals *vs.* Tutor's goals

CTS Feeling nodes allows to clearly separate learner (or more generally, user's) goals from the tutor's (or more generally, the artificial agent's), and maintain them simultaneously. The Domain Expert tries to keep track of the operations the learner is pursuing. The Feeling nodes (in the Behavior Network) represent the various global goals the agent may entertain in as a tutor. I do not think ACT-R allows for such separation of goals.

### 6.2.5 Information selection

CTS' Working Memory is a pool inspected by the Attention mechanism that selects the most important information at the moment for system-wide publication. Not everything contributed by modules and other processes is of equal value. In CTS, the global activation value of a coalition indicates how important that information is, either intrinsically or with respect to the current situation. So, some things may be temporarily neglected to the benefit of more urgent or otherwise important information, as indicated by the activation of the coalition.

ACT-R functioning realizes something somewhat similar with its goal stack. Although buffers contain only one kit of chunks at a time, previous collections are called back when a previous goal pops back up on top of the stack. Elements in the buffers (corresponding to the left-hand side of rules) are not attributed values; there is only a plain unification taking place. However, expected utility calculations for the rules achieve the same result. What it just does not permit is the competition of goals for prioritization.

In ACT-R, the goals in the goal stack (up to version 5.0) all influence the analysis of the situation by the declarative memory, just as CTS' Feelings and Desires do influence the action selection. ACT-R's conflict resolution mechanism (which analyses at the sub-symbolic level the expected benefit of taking various actions) leads to the same prioritizing *of information* as that in CTS because the rule selected

is chosen partly on the basis of the current value (importance) of the information. From that point of view, its does a job resembling that of the attention mechanism of CTS looking over Working Memory. ACT-R conflict resolution may, just as in CTS, lead to usefully neglecting some information. That will happen when a rule with a left-hand side not considering some information, computes to a higher expected utility than a rule that integrates more buffers. However, CTS does not need to have predetermined specific rules that make use of only part of that information to select it (or ignore it). In CTS, the most globally energized coalition simply is selected and published, then the various modules decide whether they use it. There is no burden on a central coordinator and on its designer, leaving each separate module do whatever its designer has planned for that information.

What's more, although many coalitions are predetermined by the system designers or learned in the course of the operation, they do not have to be predetermined or exist in declarative memory for their usefulness to be calculated and be selected; generic rules of association allow on-the-fly coalition formation in working memory, sometimes building unexpected combinations. If such combination reflects a regularity, the implicit learning mechanism will eventually learn it as a valuable, stable coalition, creating a new element for the declarative memory to assimilate. At that point, ACT-R chunking mechanism resembles CTS' coalition creation process.

### 6.2.6 Action selection

CTS's Behavior Network is, in part, a rule-based system where States play the role of the left hand side in a production rule. A State turns on when it recognizes in a broadcast the information it corresponds to, and in so doing, it serves as preconditions (complete or partial) of the action node. The same State is also the confirmation of the effect of the previous action. This description of the context is relatively stable (a State has to be turned off, or slowly does so on its own), so the Behavior Network is always up-to-date with the situation. Then, there is the energy part of the planning mechanism that complements the "logical" one. States feed Behavior

nodes with activation, "priming" them and indicating which ones are appropriate to the context; Feelings do the same, from "atop", indicating the current global goals of the agent (many may be active at the same time with varying strength), either reacting to the environment or to an "internal professional agenda".

ACT-R has some of the same energy mechanism, with current goals giving activation to nodes in the declarative memory. The base-level activation, that effects the learning from experience in CTS, exists only in memory chunks; it nevertheless also exists for rules but under another appearance, as *rules utility* calculations. These subsymbolic processes keep a memory of past outcomes, estimating the cost (in time) of the operation and the probability of reaching the goal.

What is different in CTS is that the summation of the energy sources in each Behavior node, combining with the base level activation of each (which reflects past successes of a Behavior), is somewhat resilient and perpetually shows which is the most globally appropriate Behavior (one may think of these Behavior nodes as the candidate immediate goals or, in Baars words, the local goal context). There is no need to completely reevaluate the whole rule base at every cycle and redo the probability computations, as ACT-R needs to do. A light iterative update on the activation values (a summation process) suffices.

### 6.2.7 Consciousness in ACT-R and CTS

These two systems bear commonalities on awareness and consciousness. In ACT-R, there is *unconscious* processing at the subsymbolic level (these processes are not controlled by the "conscious" rules), and within the various modules. What modules deposit in their respective buffers is the information that could be thought of as becoming *available* to consciousness. I would say that it is brought to the "system's" awareness when it is processed by the central production system to produce a system-level actions (the agent taking an action). I put "system" in quotations marks because the "system" here is, in fact, only the central processing module, and not the whole agent, as is the case in CTS processing. The rest of the system is

made aware of only the effect of the conscious content when new stuff is deposited in the buffers by the rule.

In CTS, what is selected for becoming conscious is published *at large,* to the whole system (in the proper sense, this time) so that all the various modules become aware of it and may process it. ACT-R has a centralized processing of what has been made "conscious", whilest CTS has a distributed processing of that information.

We can point out two other differences with respect to consciousness. First, there is nothing in CTS' architecture prohibiting the firing of unconscious actions. For instance, no central rule is involved in voluntarily, "consciously" updating the buffers. Another example is that some codelets may have become part of an automated process and send some requests to a database without recourse to conscious involvement (publication). The information coming back could however be processed by the perception, eventually bringing the result of the action to consciousness. ACT-R can only take "conscious", voluntary actions towards the external world.

Second difference, CTS involves multiple levels of analysis and action-taking. Its metacognitive codelets analyse what is happening in the agent, what success plans have, find repeating information and temporal patterns. They may react and ask for the correction of some Behavior. To my knowledge, ACT-R has no specific mechanism allowing for such metacognition; once the information has been grabbed by a rule, it will very likely be modified and cannot serve a second pass for metacognitive analysis. Metacognitive rules could fire first and leave buffers untouched, but they still would not allow for temporal correlations.

### 6.2.8 Summing up

The two architectures have much in common, more than I would have anticipated at first. They both rely on a strong commitment to a cognitive approach to the mind and rest on empirical research, although one must recognize that ACT-R has an edge with respect to exact correspondence with empirical data. It has been very

much validated. However, CTS will continue to progress and may reach an equivalent validation status, especially with the continuing work of professors' Franklin and Baars team on LIDA. CTS may have an edge over ACT-R on some aspects: (a) its Working Memory allows information to simmer and create unforeseen associations; (b) this creation of associations may lead to new concepts creation, allowing for the learning of the environment; (c) its clear separation of the tutoring know-how and of the user modeling facilitates the independent design and improvement of the two aspects; (d) its generic processes for information selection in Working Memory allow for a relatively easy extension of the architecture with complementary modules; (e) CTS better reproduces at least one aspect of consciousness: it may take unconscious actions on the environment and become conscious of only the results; (f) its multiple levels of analyses incorporate autonomous metacognitive capabilities; (g) with its essentially distributed processing, CTS lets behaviors *emerge* much more than ACT-R allows with its centralized rules system. It is CTS' fundamental emphasis on consciousness mechanisms that grants it many of these advantages.

Which is best? ACT-R gives more control to the designers; CTS incorporates a multi-level, multi-aspectual analysis of the situation and allows for a more "natural", emergent behavior. All in all, it depends on what you are looking for from the system!

# Chapter 7

# VALIDATION AND EVALUATION

The fundamental goal of this research is to establish the potential of using Baars' Global Workspace (GW) principles and their implementation in Franklin's architecture, IDA (recently extended into LIDA), for building an efficient artificial tutoring agent. So what I need to establish as a proof of concept is the capability of the GW-based architecture to support valuable tutoring services such as model tracing, coaching, criticizing, diagnosing, etc. I present in this chapter a more formal evaluation of how well this goal is reached.

## 7.1 VALIDATION METHODOLOGY

The method I apply for the validation is an expert analytical evaluation that compares CTS to three elements:

- the GW theory;
- commentaries from Leo Hartman, a Technologies specialist at the Canadian Space Agency (CSA), about the needed behaviors for the tutoring agent;
- behaviors, strategies and rules inferred from field observations I made at the CSA during astronauts' training to Canadarm2 manipulation.

The analyses for the first point serve to validate CTS as reflecting appropriately the GW theory; I will simply summarize here the parallels that have been drawn throughout Chapter 4 and Chapter 6, and offer my opinion about the conformity of CTS. Being very close to IDA, which *raison d'être* is to implement Baars' theory, it

seems very unlikely that this target could have been missed. Nevertheless, our own modifications might have taken CTS out of the realm of direct implementations of GW theory, and this is something that needs to be assessed.

Mr. Hartman has inspected the model showing the proposed behaviors for CTS just before construction of the Behavior Network began. His comments were noted and integrated as much as possible in the appropriate structures of the prototype. These expert observations have been complemented by first-hand field observations of astronauts being trained by professional tutors. I observed the tutors actions, reactions, initiatives and strategies while I had the privilege of standing next to the astronauts to note their attitudes and reactions to the tutors instructions and interventions. The notes obtained from this and from collaboration with Mr. Hartman yielded many crucial artifacts:

a) a list of valid tutoring behaviors;

b) examples of reactions to be expected from the astronauts;

c) examples of tutoring situations that inspired the proposed scenarios.

The interventions that CTS made while traversing the scenarios presented in Chapter 5 can now be gauged against what is expected.

The real test will happen when we can submit CTS to "real-world" interactions with astronauts on tutoring sessions. In its current state, CTS does not include enough tutoring knowledge, pedagogical strategies, and domain knowledge to offer valuable tutoring advice in a variety of situations. Only setups corresponding to the described scenarios can be sustained.

## 7.2 RESULTS

### 7.2.1 Validating CTS against the Global Workspace theory

The main idea of the Global Workspace theory is that the brain has a way to allow separate, distributed processors (neuronal groups) to share information when needed, to collaborate and coordinate their efforts. In essence, this theory has explicit roles for consciousness. Chapter 2 enunciated the principles of Baars' Global Workspace (see sections 2.3.2 and 2.3.3) and the functions consciousness plays in the human mind (see section 2.3.4). I recall them here and show how CTS integrates them.

***Principles:***

- The distributed and decentralized nature of processing

| In Baars' GW Theory | In CTS |
|---|---|
| The brain is massively parallel, with a collection of distributed specialized networks; the processing is widely decentralized in any given task. The detailed work is done by millions of specialized neural groupings without specific instructions from some command centre. Conscious processes have a great range of possible contents, but the range of any single unconscious processor is limited. | CTS operates on the basis of collections of codelets and specialized modules that work independently from one another. Codelets are especially meant to be highly efficient at processing a simple aspect. However, there is nothing currently constraining codelets complexity; designers have to do their best to respect this line of conduct.<br>Modules operate on the same premise of specialization, each also rendering a specific service but at a higher level of organization: tracking user's knowledge, user's mood, remembering user's psychological profile, memorizing events and concepts, analyzing user's maneuvers, and so on. |

- The collaborative and competitive nature processing

| In Baars' GW Theory | In CTS |
|---|---|
| There is competition between the multiple sources to become conscious.<br>The various brain regions collaborate to deal with the situation at hand, supply information or process what is published. | There is competition in the Behavior Network for the selection of the most relevant Behavior in the current context, just as the Behavior nodes shunt those that would undo their preconditions.<br>There is collaboration from various modules to lend a hand when they can help, bringing information or processing power. Collaboration is also found between individual codelets through associations that create coalitions and that stimulate other codelets. |

- Information converges then diverges

| In Baars' GW Theory | In CTS |
|---|---|
| Global Workspace theory (GW) suggests that conscious experiences involve widespread distribution of focal information obtained from multiple sources converging and organizing themselves together. | The information parcels are sent from the various sources to Working Memory. There, they organize themselves in coalitions. Then, the one selected is broadcast, announced at large, diverging towards all the subsystems. |

- Recruiting of unconscious resources is due to consciousness

| In Baars' GW Theory | In CTS |
|---|---|
| Consciousness is needed to trigger a great number of automatic routines that make up specific actions. | What comes to Consciousness brings the modules and various attentional resources to react and either send in codelets containing some information they possess, or take charge of some aspect of the situation and contribute to the resolution. |

- Interpretation is related to multiple levels of context

| In Baars' GW Theory | In CTS |
|---|---|
| Some unconscious networks, called contexts, shape conscious contents and strongly influence conscious processes.<br>According to Baars, we continually benefit from a host of mental contexts without experiencing them as objects of conscious experience. As long as they are successful, contextual predictions give no sign of their existence (Baars, 1997b, p.116). | Many things in CTS form the context that brings meaning to what has been perceived: States, activation of Feelings and Desires, activation received by a Behavior node from connected anterior nodes, links strength, base-level activation in Behavior nodes, operation tracking by the Domain Expert, and analyses by the various modules of the Learner Model. They all propose an interpretation of the events and influence which codelets will start working.<br>Expectation codelets also take an important role here. They are by nature totally contextual, being emitted by a Behavior node. If an information element coming into WM tells of a mismatch to what it expected, the codelet puts into WM information codelet(s) stating its interpretation of the event. Otherwise, it dies away when satisfied. |

- Seriality and the limited capacity of consciousness vs. the parallel unconscious

| In Baars' GW Theory | In CTS |
|---|---|
| Conscious ideas occur one after another (serially) (Baars, 1997b, p.63). There cannot be more than one idea conscious at a time, but unconscious processors can operate in parallel. | In CTS, Consciousness publishes a single idea (coalition of information) at a time, although it may be rich.<br>But lots of processing happens in parallel, in various modules, and in each individual codelet (although this parallelism is somewhat simulated, due to technical restrictions). |

- The cognitive cycle

| In Baars' GW Theory | In CTS |
|---|---|
| | The cognitive cycle is an hypothesis set forth in IDA (Baars and Franklin, 2003) to better explain the multiple operations happening in the brain. Moreover, it is organized to preserve consciousness seriality by putting conscious broadcast as a specific step within it. But the cycle is not part of Baars GW theory in itself. Nevertheless, it has been kept from IDA as it offers a much better understanding of the theory and of the brains operations. |

- The highly structured and internally consistent nature of conscious ideas

| In Baars' GW Theory | In CTS |
|---|---|
| Selective attention always involves a densely coherent stream of events. We never mix up two streams of speech with different contents, or even with different vocal quality. It is generally true that conscious experiences are internally consistent. | In CTS, percepts are naturally structured and consistent since the Perception Network has been designed so, thus the coalitions that are formed from them. Coalitions may evolve and enrich from what is supplied by various modules; rules have been set to specify how information codelets may attach, preserving structure and consistency. |

- There is a deep level of context that is stable and guides all other processes: the Self

| In Baars' GW Theory | In CTS |
|---|---|
| Self refers to the deepest levels of context: the basic intentions and expectations we have toward the world, ourselves and each other. It is a framework that remains largely | The Self is implemented in part in CTS' Feelings and Desires. Partly formed by innate dispositions, partly by past experiences and their outcome, it also exists in the form of |

| In Baars' GW Theory | In CTS |
|---|---|
| stable across many different life situations and guides our lives. Largely unconscious, it nonetheless profoundly shapes our conscious thoughts and experiences. Different levels of self seem to work together. | links and their strength in the BN, and in base-level activation of the BN nodes. Innate dispositions are attitudes put in place within the Feelings (specific sensitivity to some events) by the system's designer, or eventually by auto-adjustment. |

- Voluntary and involuntary attention

| In Baars' GW Theory | In CTS |
|---|---|
| Most shifts of attention are not under moment-to-moment voluntary control. Mismatch-detection may trigger our attentional mechanism to direct the surprising event to consciousness.<br>But one may willingly *decide* to prioritize some source or some type of information, in other words, to focus one's attention to it. | Involuntary attention happens in CTS in each single cycle from the natural selection of the most activated coalition in WM, eventually coming from the action of an expectation codelet that suddenly puts some strong piece of information into WM.<br>Voluntary attention is created with short-lived attention codelets sent by a Behavior or by some metacognition codelet. They will eventually reinforce a coalition corresponding to some piece of information they monitor in WM. |

***Functions of consciousness:***

- Creating access to unconscious resources

  Something put into CTS' Working Memory may be selected by the Spotlight Manager (the Attention mechanism) and broadcast to all codelets and peripheral modules by the Access Consciousness. This is the only way to voluntarily send information to otherwise unreachable codelets. So, the "unconscious" resources become aware of the information by its "coming to Consciousness".

- Prioritizing

  The items of information that exist in WM as coalitions (or assemble there) add their activation (their importance) to the central node of their coalition. Their importance is key to being selected for broadcasting. Additional activation may come from an attention codelet that joins a specific coalition, augmenting is importance, thus the probability that it will be selected before other coalitions. The activation level of a coalition is how information is prioritized, and this is what happens in Working Memory with the competition among coalitions.

- Using unconscious error-detection and correcting defective perceptions

  Some of this error-detection is accomplished by expectation codelets, watching the outcomes, determining if they corresponds to what was expected. If there is mismatch, they put an information codelet about their observation into WM.

  Another part of error-detection is made by attention codelets noticing problematic situations such as when some information element returned by Declarative Memory has no relation to the current context. Metacognition codelets also do error-detection and eventually cause plans editing. One of them may observe, for instance, that the same plan, or the same strategy, has been used more than once in the current situation without success, indicative of the need for a repair in the assumptions of the system or the inference process. Metacognition codelets are not currently implemented, but are part of planned additions. The remainder of this function will be accomplished by the natural “clean-up” function offered by Declarative Memories: the associations that are returned by the *Sparse Distributed Memory* (the algorithm used to create our Transient Episodic Memory) do some fill-in-the-blanks, returning prototypical or "averaged" information which can be used to replace partial or corrupted information that it received as input (this kind of behavior is part of Kanerva’s ideas). An aspect that is not accomplished in our architecture is the clean-up that normally takes place during perception, giving meaning to the

stimulus if it can be recognized, or finding the closest match. CTS' environment being fixed in nature, we did not need to concern ourselves with this.

- Problem-solving and plans editing

  Our BN finds the best available way to solve a problem (react to the situation or act upon what is believed about the future) thanks to the information it iteratively receives from the Consciousness publications. This broadcasting organ also serves to solicit collaboration from peripheral modules so that they send needed information or do processing required to fix a problematic situation. For example, the Domain Expert sends the appropriate hint's content so that CTS may take care of the astronaut's inactivity. Another form of problem-solving is the interactive, on-line building of a delivery (presentation) plan, when CTS is asked by the astronaut to do some teaching on some concept. Initial design for a BN stream has been made for such a stream; its principles have been presented in Dubois (2005).

  "Soft" plans editing is done, for some part, in deliberations: iterations of publications-responses supply the needed specifics for the interventions. Plans editing also comes by the learning in Behavior nodes base-level activation (plans that do not work get weaker). But stronger plans editing is considered, and would be implemented in a collection of streams in the BN.

- Adapting mental structures for learning

  This adaptation refers to the time needed for declarative memory structures to reshape to accommodate new material. Without consciousness, which creates a stable and "durable" information, learning would not have the chance to occur, at least not for very different information. This role is hidden within the broadcasting in our architecture: learning in declarative memories only happens from what is broadcast; what is fed to them directly from WM only serves for the *recall* of the associations, not for learning anything.

- Reflection, self-monitoring and executive control

  Reflection is present in CTS' deliberations, where the agent becomes aware of concepts and ideas that show up in its Working Memory from various sources, including from attention codelets that try to bring up to consciousness things they notice. In some way, it resembles the *inner voice* that we experience and use to discuss with ourselves. Self-monitoring is accomplished by the metacognitive codelets that keep analyzing the agent's functioning. Executive control is present in the Feelings/Desires that drive the BN; it is found also in the *innate* attention codelets, and in the orientation of the Attention mechanism by temporary attention codelets that can be released by the BN to bias information selection.

- Creating the context for understanding

  Context is unconscious, but is brought about thanks to the conscious publications. CTS' Access Consciousness publishes information that makes all modules aware of the situation. Traces of this broadcast appear everywhere in the architecture: as States in the BN, in the statistics updated in the LKM, in the tracking made by the Domain Expert. These "priming" events influence which codelets will become active next. Consciousness also creates context by the coalitions growing *complex,* that is, include many aspects of the situation: the context. In this way, when broadcast, a whole context is presented to the listening codelets and modules.

- Optimizing the trade-off between organization and flexibility

  As long as available plans (in the BN) and solutions work, the optimized unconscious processes keep going. If unexpected results happen, though, new solutions are sought, either within the BN, or through plan repairs and modifications (not implemented yet in CTS, but feasible). Also, feedback to Behavior nodes by expectation codelets reinforce successful acts and tend to favor solutions that work. Metacognition codelets (none have been implemented yet) noticing difficulties (through conscious broadcastings) with a plan may do mi-

nor plan repairs, or request that deeper analyses be conducted. They may request that plan reparation streams (not designed, but feasible) be put to work to adapt the plans.

- Recruiting and controlling actions (James' ideomotor theory)

  This point resembles the recruiting of resources, but brings the idea that any idea (internal proposal) for an action that comes to mind (to consciousness) is acted upon unless it provokes some opposing idea or some counter proposal. James gives an illustration of not wanting to get out of bed and blocking this act, until the mind gets wandering about the load of the day and suddenly, one realizes that he got out of bed, as the automatic morning action. Volition (voluntary orientation of actions) has stopped blocking the automatic response to planning the day while in bed. We have implemented that portion of the ideomotor theory with inhibitions in the BN, and with codelets opposing a proposition during deliberations.

In summary, I conclude that CTS does implement most of Baars' theory, even if some aspects are part of future works.

### 7.2.2 Validating CTS against some behaviors that are expected from a tutor

#### 7.2.2.1 Indications received from the CSA's specialist

What is expected from CTS has been dictated by an analytical evaluation from a Canadian Space Agency specialist, Mr. Leo Hartman. He has been presented a mock-up of the proposed prototype for CTS. His observations have been taken into account when elaborating the behaviors available to CTS and its user interface. Comments referring exclusively to the simulator are not presented here, which may

give a false impression that he commented almost exclusively on the user interface. Relevant comments can be summarized as:

- Keep text boxes short; do not offer lengthy commentaries.
- Offer a recap after an exercise.
- Use a friendly style; do not use a literary style or to polite preambles.
- Do not insist on helping or offering orientation after the astronaut has been given the option to say “No, thanks!” Do not pursue the intervention to offer any "very interesting" supplementary tip.
- If the astronaut says “No, I do not want help”, he really means it. Give him the chance to express himself about this, and respect his wishes.
- Choice boxes need to be perfectly adapted to the context.
- The astronaut needs to be able to "play" with the Station and examine the situation from any angle.

#### 7.2.2.2 Indications inferred from field observations

Field observation of astronauts' training also gave me many precious indications on how actual tutors ought to behave, what type of intervention they put forward, and when they chose to remain silent. Four excerpts of the interactions appear in Appendix D. They illustrate the kind of notes that I took; not all principles enunciated below can be illustrated by this subset. Some of these principles are:

**Table 7-1 Behaviors observed from human tutors at the Canadian Space Agency.**

| Behavior observed | Related excerpt(s) | Implementation |
|---|---|---|
| Feedbacks, even words of encouragement, are offered in a very calm tone. | 1 and 4 | In future works |
| When the astronaut pauses for some time, the tutor tries to determine the source of the problem. The tutor either has an idea of the problem the astronaut locked himself into, or looks at his face to try to evaluate what is happening in his mind, or asks straight out for clarifications. | 3 | Implemented, except for "looking at learner's face" |
| When the astronaut moves Canadarm2 too close to a Station's structure, the tutor does not point it out at the outset, but rather hints at the impending problem. | none | Implemented (in a BN stream) |
| The tutor does not always react when he detects a problem. His behavior indicates that he sometimes evaluates that it would be beneficial to wait before intervening. | 2 (see scenario 2) | Implemented (with simulated Learner Model) |
| When the astronaut makes several trips back-and-forth to the scaled-down model of the Space Station behind him, the tutor understands that the astronauts has a difficulty with understanding the current views. | none | Soon (with attention codelets) |
| There is plenty of supportive feedback, probably to create an immediate reinforcement of good thinking and appropriate maneuvering (implemented but not demonstrated in the presented scenarios). (excerpt 4) | 4 (not demonstrated in scenarios) | Implemented (in a BN stream) |
| There is always recapitulation at the end of an exercise, pointing out good thinking, well-done maneuvers, and mistakes with suggestions for a better performance next time. | 1 | In future works (required TEM and more material in DE) |

| Behavior observed | Related excerpt(s) | Implementation |
|---|---|---|
| Teaching and tutoring of the astronauts use scaffolding within and over many lessons (tutors present gradually more complex concepts and maneuvers that use previously learned material). | none | In future works (required more streams, and more material in DE) |
| The coaching offers much help in the first attempts, and progressively reduces support when the astronauts manifests the capability to do more, or faster, without mistake. | none | In future works (requires LKM, or attention codelets with more complex BN streams) |
| Manipulating the physical objects is very helpful and strongly recommended. | 3 (partly observable there) | Implemented (uses a virtual camera of the simulator) |

### 7.2.3 Comparison of CTS' performances to CSA's specialist recommendations and field observations

CTS has been submitted to the scenarios set out in section 5.4. Its reactions demonstrate that, even in its prototype state, it makes correct use of the strategies that have been incorporated in it: detecting unwarranted silences or inactivity, using progressive hinting, and refraining from intervening when it is better not to. Although humans do that kind of thing all the time, it is not so straightforward to accomplish it right. Some human tutors talk too much, some offer too much help when they should just give a few hints. Some intervene all the time, or conversely do not offer enough support.

The initial trials of CTS show that it reacts in the expected ways when submitted to the proposed situations. It is able to choose an appropriate action when it detects inactivity on the astronaut's part. For that, it makes use of the cause proposed by the Learner Model; if it cannot think of a reason for the silence (the sub-modules of the Learner Model supply no probable cause), *and if the idea of intervening does not get opposed*, then it makes an offer for help (Figure 35), which will then be followed by choices if the astronaut accepts. Since the LM is still under development, it has been simulated. But when it becomes available, I am confident that the architecture will process its richer contributions correctly.

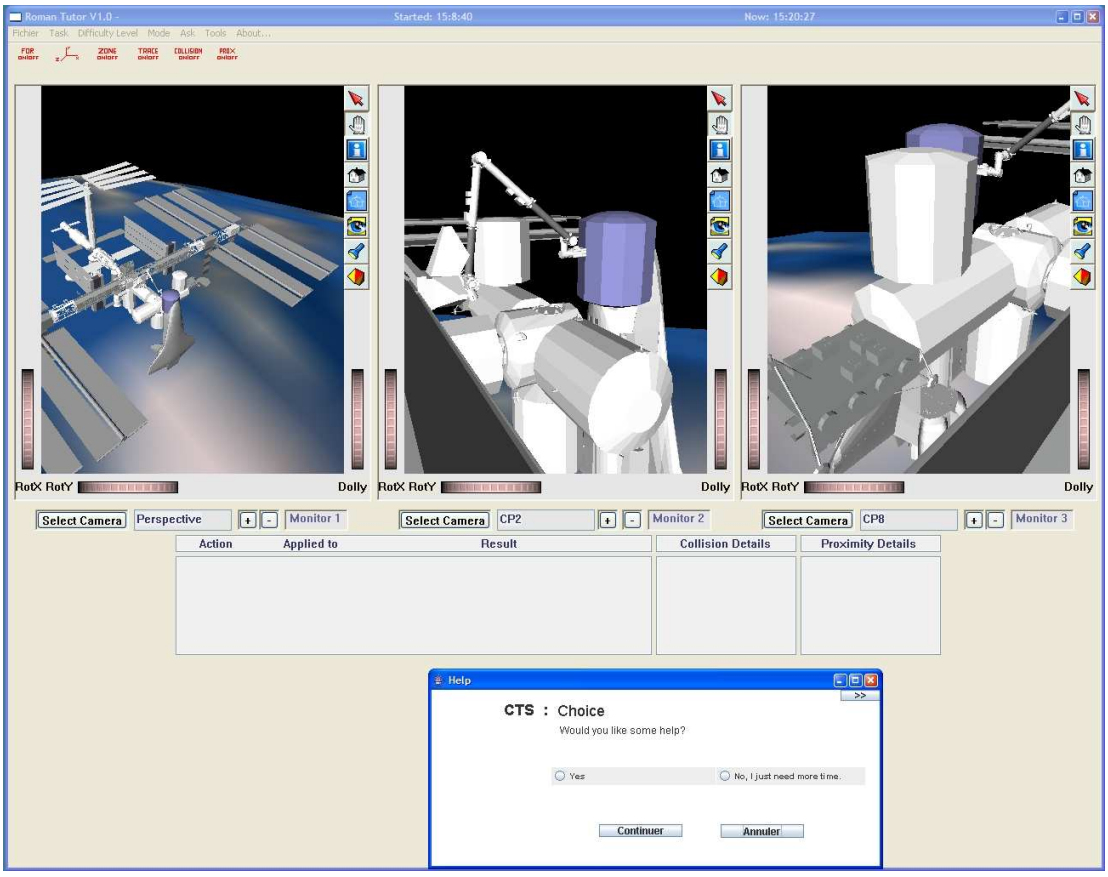



**Figure 35 When CTS cannot think of a cause, it simply offers help** (if the idea of intervening is not opposed during the deliberation). If the astronaut accepts, then choices are offered for selecting the help desired.

CTS also detects incorrect procedures and gives feedback that solicits the astronaut's metacognitive thinking. CTS can choose the appropriate way for giving feedback among those available: hinting (Figure 36), stating the fact, highlighting the problematic element on screen, or replaying the last maneuver. Here, again, the Learner Model has been simulated, but information returned by the *stubs* has been properly used by the rest of the architecture.

CTS is also capable of diagnosing a situation. Its primary mechanism for this,

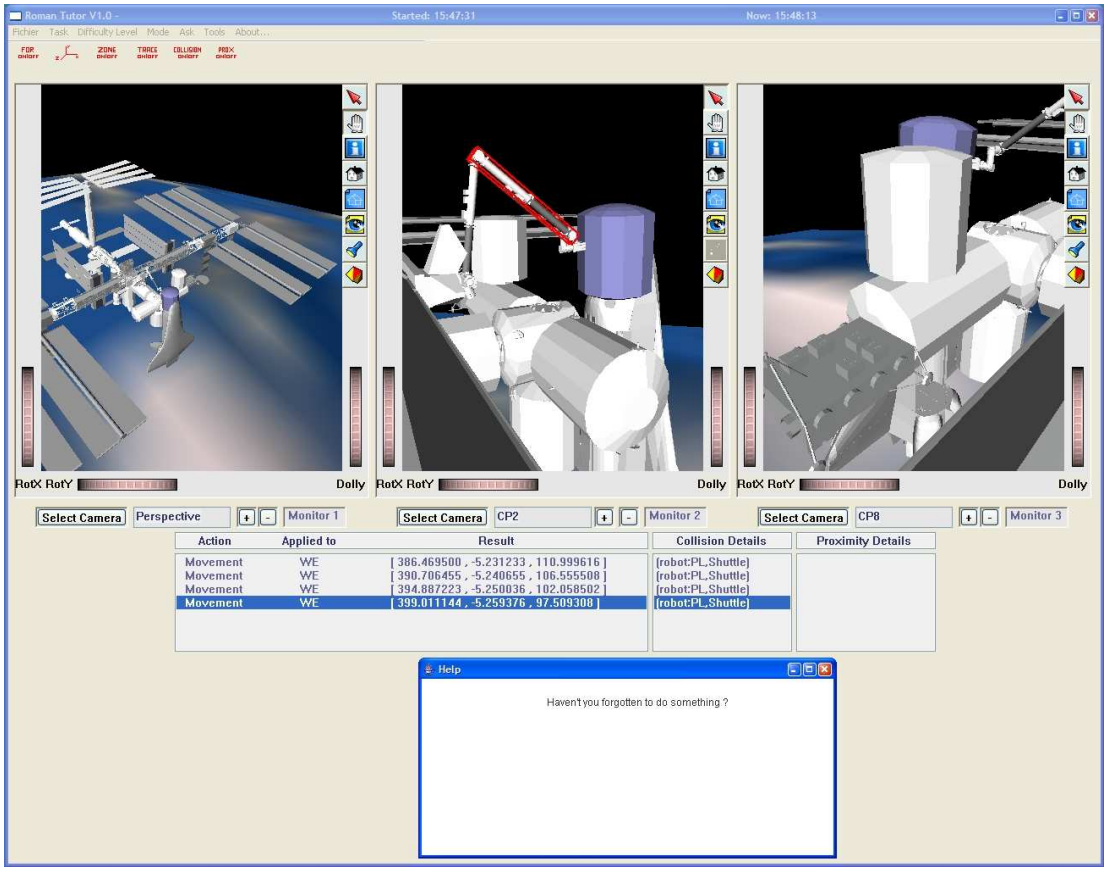



**Figure 36 CTS reacts to an incorrect procedure.** Here, the astronaut started moving Canadarm2 before creating all the necessary views on the three monitors. Admittedly, the help windows presented could use some polishing with choices like «No», «Why do you ask?». In any case, if the astronaut does not answer this message by adopting the needed corrections, CTS presents a more specific hint, then the plain fact, and, if nothing helps, the appropriate operation. Four levels of hints are the current remediation available.

the Learner Model, still under development, is complemented by an interactive diagnosis sequences found in the Behavior Network. One has been elaborated for diagnosing and remediating proximity situations (see Appendix F). It performs quite well, reorienting the line of questioning and the actions proposed on the basis of the astronaut's answers. The publications (the elected information becoming conscious) set properly the context in the BN.

These simple evaluations demonstrate CTS capability to take into account many sources of information, to combine their specialized abilities and prioritize what occupies its Working Memory. They also demonstrate its ability to choose an action appropriate to the context, deliberate to adapt the intervention, and even decide whether it should intervene or remain silent. The agent's resulting behavior emerges as a natural process that even incorporates intuitive processing.

# Chapter 8

# CONCLUSIONS AND FUTURE WORK

## 8.1 CONTRIBUTIONS OF THIS RESEARCH

I believe the results of my research efforts contribute both to cognitive sciences and to the field of artificial intelligence in education (AIED). In particular, it should be of interest as:

1. a new implementation of Baars' theory (we completely reimplemented IDA), presenting at the same time an exploration of other ways to implement Franklin's ideas.
2. a new cognitive tutor infrastructure with an architecture based on a theory of cognition that incorporates an explicit consciousness mechanism as its core. It offers an exploratory tool for cognitive scientists (mostly for philosophers, psychologists, linguists) and an alternative platform for ITS designers to build cognitive tutors. This is a significant contribution in AIED field where current cognitive tutors are all based on ACT-R.
3. a framework for developing other "conscious" cognitive agents with new insights about how to think about and use a cognitive architecture. This opens the door for other agents and other applications to make use of the most powerful adaptive means human kind exhibits: consciousness. Its modular architecture should be of interest to learning systems designers.

4. a project that already offered opportunities to many students for learning, for gaining experience in research and obtaining their diplomas; a project that offers many open avenues for other researches.

The second point deserves a few lines here, as it has rather been left alone in my document. Indeed, I find myself very excited at the prospect that ITS designers may use CTS as a foundation to be completed and extended. The reasons I see for them doing so is that the architecture offers a powerful holistic approach to analyses, adaptation and planning. CTS' decision processes and actions can take into account multiple factors naturally: learner state in its varied facets (knowledge, learning trends, psychological profile, mood and emotions), tutoring knowledge, and CTS goals as a proficient tutor that wishes to pursue the objectives of the pedagogical theory he currently "believes" in. Its parallel processing of all the aspects, their iterative addition through deliberation and their combination in Working Memory make for a rich decision process that is very flexible, very adaptive. The prioritization accomplished by the Attention mechanism helps cut through too various possible actions, through too much information, and concentrate on what is of paramount importance.

Without the capacity to learn and improve, a tutor may become very annoying to learners, with unavoidable twitches and irritating personality traits. Happily, CTS can learn and can adapt to the context. Attention and metacognitive codelets make it relatively easy to add and refine specific behaviors, even if CTS does not yet have a full-blown transformation mechanism for its Behavior Network, with only "soft" adaptation capabilities. The Feelings can be easily molded to support new attitudes, new personalities. Its modularity permits the addition of new capabilities that may be developed and perfected separately, for instance new processing options based on supplemental pedagogical theories.

When I start elaborating on the potential of the architecture, my mind becomes on fire and I can easily get carried away. Much work remains to be done, even just to complete the cognitive architecture and to enrich the pedagogical capabilities, and even more to reach this architecture's full potential. But I perceive these as very mo-

tivating prospects. I know I will witness this framework becoming a better tool, and CTS progressively turning into a convincing interlocutor. I have a strong desire of offering a cognitively interesting architecture, a framework for powerful agents, and reusable facilities for new levels of tutoring systems.

Here is a list of works I intend to pursue in the coming months (and years!):

**Making CTS application to astronauts training more powerful**

- Adding more conceptual nodes in the Perception Network (PN) and enrich the formal grammar accordingly.
- Designing exercises that can be used autonomously or as remediation following a diagnosis.
- Completing the diagnosing streams, and adding new ones.
- Elaborating more capabilities in the Behavior Network (BN), such as affective support, flexible textual interactions, styles and tone for these interactions, a variety of personalities, and developing the existing prototype's streams. More pedagogical strategy streams have to be created, with some dedicated to the delivery of the subject matter. Multiple pedagogical theory streams are also needed, with capability to take charge when a theory is declared ineffective by a metacognitive codelet.
- Creating the linkage to an ontology of pedagogical theories so that the BN can be validated while it is being designed or while it auto-adapts.
- Augmenting the simulator so that it sends more information and becomes able to offer more services to CTS (highlighting objects on the monitors or aspects of the user interface, replaying sequences of actions, presenting exercises generated on-the-fly by CTS, etc.).
- Augmenting the simulator for analysis of the views selected by the astronaut.

**Adaptation capabilities**

- Implementing metacognitive codelets to modulate, modify and correct the agent's behaviors.
- Adding metacognitive codelets that do analysis for temporal pattern recognition.
- Designing and implementing the capacity for CTS to modify its BN with new nodes following an analysis by metacognition codelets (for instance).
- Adding more concepts and generalization capabilities to the Perception Module; examining how LIDA's slipnet capabilities could be inserted.
- Adding the Transient Episodic Memory module, currently under development.
- Designing and implementing emotion capabilities for CTS.
- Adding some user interface widgets that would allow the user to express its reactions to the latest intervention, or how he now feels about the task (so that CTS may take appropriate action, pedagogical or affective).

**Perception Module**

- Examining the linkage of the Perception Network to the Declarative Memories.

**Tools to make CTS a framework for "conscious" cognitive agents**

- Preparing specifications and methodologies for the application of the agent to a new field. For instance, how one does disseminates the expert knowledge throughout the architecture (in the BN, in the Learner Model (which one could call more generally *User* Model), in the Domain Expert, in the attention codelets of the various types).
- Improving the actual BN Editor's capabilities and services (ex.: complete linkage to CTS actual code, automatic generation of codelets, libraries of reus-

able codelets), and developing formal and uniform specifications for codelet/class naming, for usable fields (and their role in the codelet), etc.

- Developing a way to automatically create parts of the BN from learning theories ontologies; this may lead to multi-strategy tutors that can switch from one instructional theory to another in the same learning session.
- Developing a tool that allows one to see and analyze BN's reactions and state in correspondence to the conscious broadcasts.
- Creating a mechanism that would allow CTS to explain its decisions.
- Augmenting the BN with management features such as specification of personality profiles (groups of parameters for the Feeling nodes), analysis of consistency, lists of States with their connections, etc.
- Writing specifications that indicate how and in what form modules receive publications from the Access Consciousness, how and in what form they are expected to react and respond.
- Specify an open standard for the Learner Model so that it becomes easy to connect to existing models or import them.

## 8.2 SCALABILITY OF THE ARCHITECTURE

This point is of interest since I claimed that complexity is the beast to tame when considering a tutoring agent (or any agent with human capacities, for that matter). Can CTS' architecture take the load of an ever richer BN? Won't it get bogged down by many modules sending information? Can it really consider the multiple aspects of a situation?

The initial works are encouraging but call for some caution. The BN is, in part, a rule-based system and, as such, meets the same difficulties of rules complexity. However, since the analysis is distributed over different specialized structures and modules, the rules may not need to get as complex. Metacognitive codelets could be

created to alleviate the need for human designers involvement: they could analyze CTS performances and its internal operations and then bring changes to the BN, either on States, or on precondition links, or on effect links, or in the Behavior nodes.

Now, this partial answer may in fact just move the difficulties to a new focus: how do we reassemble the pieces of more distributed and complex analyses? My answer to this is that the principles of the architecture, mainly *the convergence of the information to Working Memory and the coalitions creation*, assure that the bits and pieces get together automatically. Each piece of information sent by a module joins the coalition that stimulated the reaction. So, there can be no confusion there.

So, if the basic mechanisms are designed correctly, CTS should be able to become a multi-talented expert with some social graces too.

## 8.3 REUSABILITY OF THE ARCHITECTURE

How easy is it to reuse CTS and apply it to another field? How feasible does it come for ordinary people? I see its reuse as quite feasible, especially when some of the items enumerated in the "Future works" section will have been accomplished to facilitate the manipulation of the architecture.

Some of the mechanisms are completely generic: Working Memory, coalitions creation, selection by the Attention mechanism, broadcasting. Some mechanisms are generic as a shell but need a field-specific content: the BN structure is generic, and some of its streams can be reused, especially if they have been designed as generic Behaviors or as partially specified actions. For instance, the hinting stream can be reused for any field of application, as it has been created to rely on the Domain Expert for the content it will display. But the General Help stream is very field specific. In future works, I will base the design of the Learner Model on open standards and look for a generic core that can be extended to correspond to specific needs.

## 8.4 IS THERE A FUTURE FOR THIS FUNCTIONAL APPROACH?

Asking this last question after years of research and development is troubling. In my opinion, there is a trend towards more biologically-oriented architectures. But these are further away for general use. It will take good tools that insulate the designers from the intricacies of undecipherable neural networks and "chemical" bonds. I think CTS-like architectures have a pretty long future ahead of them because they are accessible to a wider public, being partly symbol-based.

We still do not understand most of the human body functioning. Bio-chemists still try to unravel some aspects of the myriad of the body's enzymes and other compounds. Neuroscientists are still baffled by much of the brain. Functional and hybrid architectures have the potential of offering much of the macro-level advantages of biological mechanisms without waiting for our full understanding of the human body and mind complexities.

# APPENDIX A

# CONSCIOUS ACCESS THEMES FROM THE PAST 20 YEARS

## Source: (Baars, 2002)

Presented here are some conscious access themes, from various authors. The frequency of such themes in the science and philosophy of consciousness has increased in recent years.

• **Baars, 1983** 'Conscious contents provide the nervous system with coherent, global information.' [a].

• **Edelman, 1989** 'Global mapping in a reentrant selectionist model of consciousness in the brain.' [b].

• **Damasio, 1989** 'Meaning is reached by time-locked multiregional retroactivation of widespread fragment records. Only the latter records can become contents of consciousness.'[c].

• **Freeman, 1991** 'The activity patterns that are formed by the (sensory) dynamics are spread out over large areas of cortex, not concentrated at points. Motor outflow is likewise globally distributed…. In other words, the pattern categorization does not correspond to the selection of a key on a computer keyboard but to an induction of a global activity pattern.' [Italics added] [d].

• **Llinas et al., 1998**'… the thalamus represents a hub from which any site in the cortex can communicate with any other such site or sites. … temporal coincidence of specific and non-specific thalamic activity generates the functional states that characterize human cognition. [e].

• **Edelman and Tononi, 2000** 'When we become aware of something … it is as if, suddenly, many different parts of our brain were privy to information that was previously confined to some specialized subsystem. … the wide distribution of information is guaranteed mechanistically by thalamocortical and corticocortical reentry, which facilitates the interactions among distant regions of the brain. ' [f] (pp. 148–149).

• **Dennett, 2001** 'Theorists are converging from quite different quarters on a version of the global neuronal workspace model of consciousness … On the eve of the Decade of the Brain, Baars (1988) had already described a "gathering consensus" in much the same terms: "Consciousness", he said, is accomplished by a "distributed society of specialists that is equipped with a working memory, called a global workspace, whose contents can be broadcast to the system as a whole."' [g] (p. 42).

• **Kanwisher, 2001** '…in agreement with Baars (1988), it seems reasonable to hypothesize that awareness of a particular element of perceptual information must entail not just a strong enough neural representation of information, but also access to that information by most of the rest of the mind/brain.' [h].

• **Dehaene and Naccache, 2001** 'We propose a theoretical framework … the hypothesis of a global neuronal workspace. …We postulate that this global availability of information through the workspace is what we subjectively experience as the conscious state.' [i].

• **Rees, 2001** 'One possibility is that activity in such a distributed network might reflect stimulus representations gaining access to a 'global workspace' that constitutes consciousness.' [j] (p. 679).

• **John, 2001** 'Evidence has been steadily accumulating that information about a stimulus complex is distributed to many neuronal populations dispersed throughout the brain.' [k].

• **Varela et al, 2001** '…the brain… transiently settling into a globally consistent state … [is] the basis for the unity of mind familiar from everyday experience.' [l].

**References**


a Baars, B.J. (1983) Conscious contents provide the nervous system with coherent, global information. In *Consciousness and Self-Regulation* (Vol. 3) (Davidson, R.J. *et al.*, eds), Plenum Press

b Edelman, G.M. (1989) *The Remembered Present*, Basic Books

c Damasio, A.R. (1989) Time-locked multiregional retroactivation: a systems-level proposal for the neural substrates of recall and recognition. *Cognition* 33, 25–62

d Freeman, W.J. (1991) The physiology of perception. *Sci. Am*. 264,78–85

e Llinas, R. and Ribary, U. (2001) Consciousness and the brain: the thalamocortical dialogue in health and disease. *Ann. N. Y. Acad. Sci.* 929, 166–175

f Edelman, G.M. and Tononi, G. (1999) *A Universe of Consciousness*, Basic Books

g Dennett, D. (2001) Are we explaining consciousness yet? *Cognition* 79, 221–237

h Kanwisher, N. (2001) Neural events and perceptual awareness. *Cognition* 79, 89–113

i Dehaene, S. and Naccache, L. (2001) Towards a cognitive neuroscience of consciousness: basic evidence and a workspace framework. *Cognition* 79, 1–37

j Rees, G. (2001) Seeing is not perceiving. *Nat. Neurosci*. 4, 678–680

k John, E.R. *et al.* (2001) Invariant reversible qeeg effects of anesthetics. *Conscious. Cogn.* 10, 165–183

l Varela, F. *et al.* (2001) The brainweb: phase synchronization and large-scale integration. *Nat. Neurosci.* 2, 229–239

# APPENDIX B

# RELATIONSHIPS BETWEEN WORKING MEMORY THEORY, GLOBAL WORKSPACE THEORY, AND IDA

Source: (Baars and Franklin, 2003)

| Extended WM | GW theory | IDA model | Some plausible brain bases |
|---|---|---|---|
| Preconscious visuo-spatial and auditory-phonological analysis. | Unconscious input analysis. | Early preconscious perception | Early visual and auditory cortex |
| Preconscious identification of objects, words and other single chunks. | | Late preconscious perception (using slipnet) | Visual/auditory object and word recognition areas of cortex, reentering widely via gamma coherence. |
| Perceptual input into WM storage. | | Percept to preconscious buffers.<br><br>Local associations. (retrieved from transient episodic memory and long term memory). | |
| The following involve multiple GW and IDA cycles: instructions to rehearse. Rehearsal (retrieval, repetition/manipulation, and storage). Instructions to retrieve and report. Retrieval and report. (Instructed tasks are under the control of the Central Executive.) | For each conscious event: competition for global workspace until one input processor (or coalition) gains access and becomes conscious. | For each cognitive cycle that involves a conscious event: competition for consciousness. (attention codelets). | First stable re-entrant organization of perceptual and immediate association areas. |
| | Broadcast of conscious perceptual or internal contents, such as conscious images and inner speech. | Broadcast of conscious contents. | Correlated firing from sensory projection areas to target areas: parietal, frontolimbic and medial-temporal cortex, hippocampus and basal ganglia. |
| | Recruitment of resources (processors).<br>Setting of goal context hierarchy. | Recruitment of resources (behavior codelets).<br>Behavior stream. | Re-entry between target areas and sensory cortex.<br>Frontolimbic re-entrant processing to prepare action. |
| | Action is chosen and prepared.<br>Internal or external actions taken by specialized processors (networks). | Action chosen.<br>Internal or external actions taken by behavior codelets (possibly writing to preconscious buffers). | Motor efference from motor/premotor cortex. |

# APPENDIX C

# MY HYPOTHESES ABOUT

# INFORMATION'S RELATIVE IMPORTANCE

The following tables indicate the values I have given to domain aspects. There are two types of values in the table, those that refer to the type of the information, and those that add a supplement based on the specifics of the situation. These values will probably be modified when more experimentation takes place.

| | |
|---|---|
| Proposition to intervene | 0,40 |
| Opposition | -2,00 |
| Inhibition (not implemented yet) | variable |
| Hint : | |
| • First | 0,10 |
| • Second | 0,15 |
| • Third | 0,25 |
| • Fourth | 0,35 |
| Effective Collision | 0,85 |
| Movement | 0,70 |
| Proximity | 0,6 + ad-justment |
| Cause | 0,30 |
| • defective knowledge | +0,10 |
| • distraction | +0,30 |
| • fatigue | +0,30 |
| Inactivity | 0,40 |
| Problematic situation | 0,55 |
| • Missing step: Views setup | +0,50 |
| • Missing step: Steps planning | +0,40 |
| • Poor view: Monitor x | +0,50 |
| • Views: Poor combination | +0,60 |
| • Hesitation | +0,45 |
| • Milestones not planned | +0,30 |
| • Milestone incorrect | +0,30 |

| | | |
|---|---|---|
| • Error in distance evaluation | | +0,60 |
| • Error in controls manipulation | | +0,60 |
| • Error in element recognition | | +0,40 |
| • Error in localization | | +0,40 |
| Defective knowledge | 0,20 | |
| Manipulation steps | | -- |
| • poor | | +0,40 |
| • average | | +0,20 |
| • good | | +0,10 |
| Neighborhood | | -- |
| • poor | | +0,35 |
| • average | | +0,15 |
| • good | | +0,10 |
| Help request | 0,50 | |
| User answer | 0,60 | |
| Dynamic element | 0,40 | |
| General information about user | 0,20 | |
| Static element | 0,00 | |

# APPENDIX D

# A FEW EXAMPLES TAKEN FROM TUTORING SESSIONS

# WITNESSED AT THE CANADIAN SPACE AGENCY

In the following transcripts I made of the coaching sessions on the simulator, I call the astronaut “A” and the tutor “T”.

**Excerpt 1**

*In this interaction, we can see that initial coaching is about basic concepts, and that the astronaut is tightly conducted through the necessary reasonings. Even if the astronaut offers a good answer, the tutor takes the opportunity to remind the astronaut of alternatives. Once the goal has been reached to the satisfaction of the tutor, he makes explicit what was good and well done by the astronaut.*

T: On the simulator, T creates three views (one per monitor) and asks A to evaluate the distance between two specified points (LEE tip and a nearby Space Station module.

T: After receiving an answer, he asks A «Which view is most useful for the task?»

A: points at one of the monitors.

T: «Good. I like what you did: using your fingers to…»

T: «One other thing you could do is […]»

A: explains his line of reasoning, why he made that choice.

T: gives other hints about what could have be used.

T: Then, T concludes «Excellent. You used […] to obtain […]»

**Excerpt 2: Operating the Portable Computer System (PCS) and the Display and Control Panel (DCP)**

*This excerpt shows that there can be quite a bit of "practical theory" presented during a field training.*

T : Announces the general content of the session, then presents a detailed overview.

T : Describes PCS' interface, which is the output means for the DCP. He explains that A has to decide what he wants to appear on the PCS, and where.

T : Poses questions about the buttons, and the possible commands that can be created here;

T : Explains the rules that need to be respected, operations preliminary to operating cameras. The he asks:

T : «Where would you go to see the last system message?»

A : presses the appropriate button on the PCS.

T : «Excellent»

T : explains the content of the evens log, where messages come from (ex. : messages transferred by the physical equipments, such as ASK, ACCEPTED, COMPLETED).

T : suggests how to keep the log window always visible ; he also points out which information need always be visible on screen.

T : lists the steps to follow, then points with his finger to the next step on the paper list nearby the right monitor (orienting the astronaut's first steps in such a procedure)

T : asks A to find on the DCP what button would obtain some specific information.

T : presents and explains the MSS operations check-list.

**Excerpt 3:**

*We see here that the tutor may use silence as a mean of letting the astronaut understand what he proposed in incorrect. The tutor may also detect that the astronaut's answer is taking time to come, indicative of the need for a helping hand. It also shows that sometimes, after an exercise, the tutors make jokes.*

T : «Now, I would like you to effectuate a rotation of the LEE»

A : suggest a maneuver to accomplish this, but it is incorrect. T remains silent, not reacting to the answer. The astronaut understands he needs to reevaluate his answer. But he cannot offer any better way.

T : Not receiving any new suggestion, T brings out a set of arrows representing the coordinate system, along with a mock-up of Canadarm2's Hand. A will be able to see in 3D, in "real" how things look. This seems to enlighten him.

T : After the maneuver is successful, T looks at one of the simulator's monitors showing space and says, very seriously: «Now, for an extra credit, what is the name of the star at the center of the aim?» Then he starts laughing and says «It's a joke!». A starts laughing.

**Excerpt 4: Hinting**

*Often, the tutor will not come out with the correct answer; he will rather try to suggest that there is a mistake, or offer a hint to bring the astronaut to understand there is something missing.*

T : «Now, a straight Station Forward in ISSACS.» ISSACS is one of many coordinate systems used around the Space Station.

A : «Easy. A simple X motion.»

T : «Right. Now, what would it be in Internal coordinates?»

A : suggests an incomplete answer.

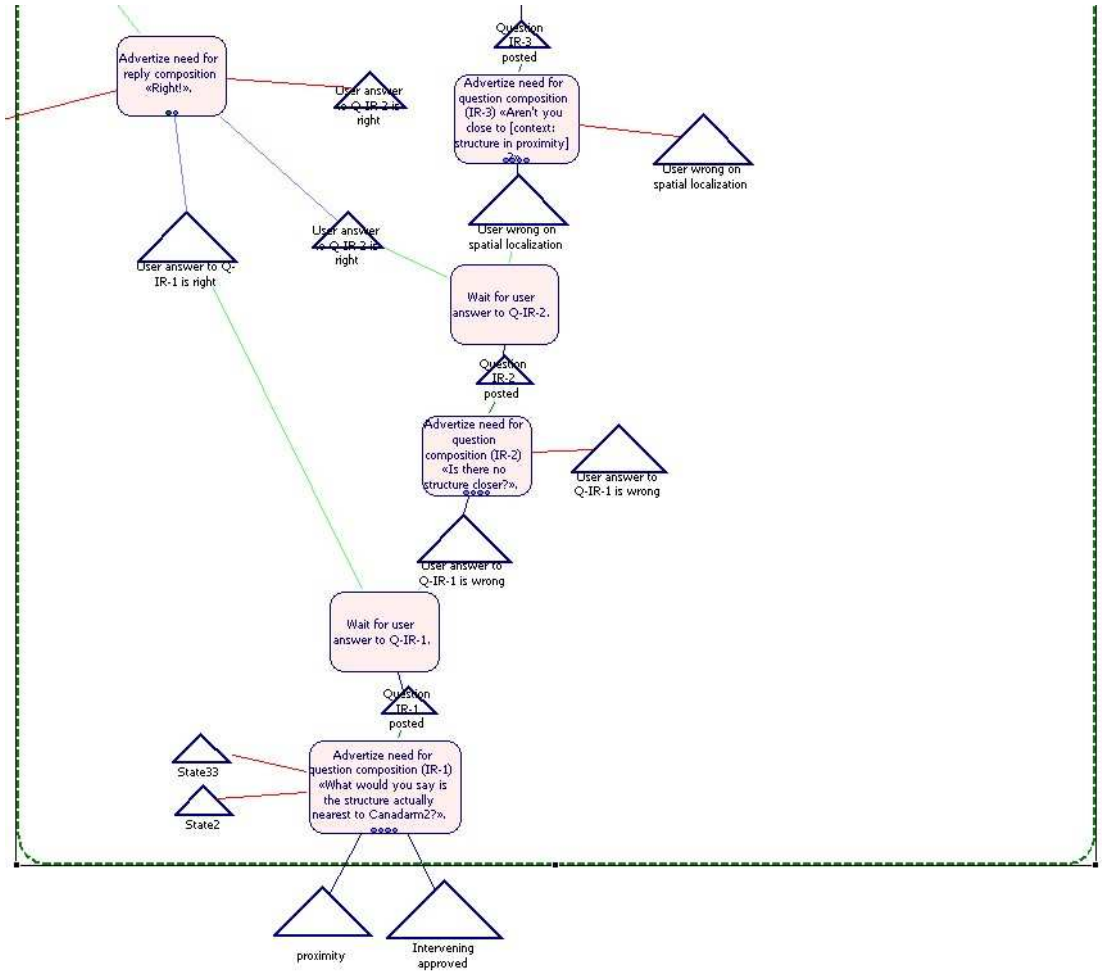
T : «Feel free to use the axes model.» (Visibly, A really needs to manipulate the physical model at this point). «Basically, you have to align the Y axis to O.»

T : «Nicely done. Excellent.»

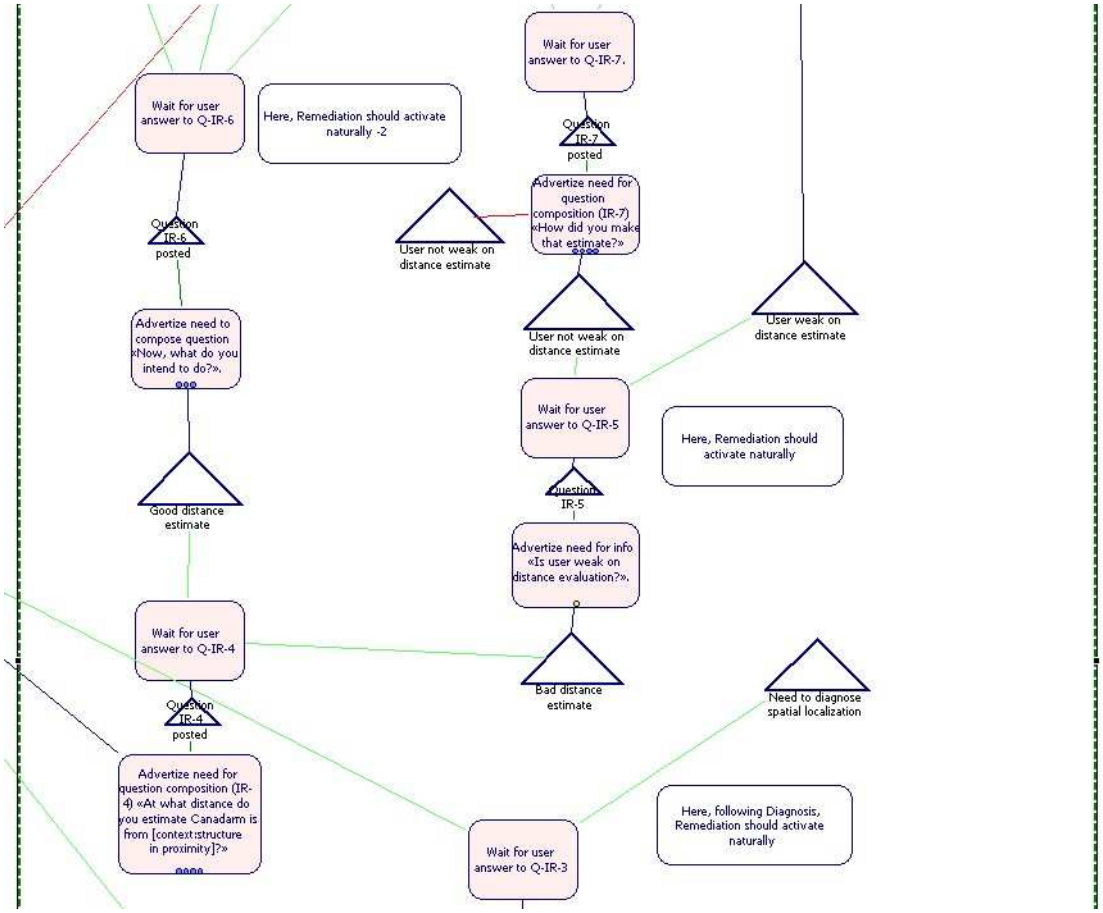
In summary, we can see that the tutors prefer to use hinting that offering the right answer straight out. They sometimes use silence as an implicit hint. Sometimes, just a short bit of theory is *plugged* at an appropriate moment. And they offer plenty of feedback, mostly as positive reinforcement.

# APPENDIX E

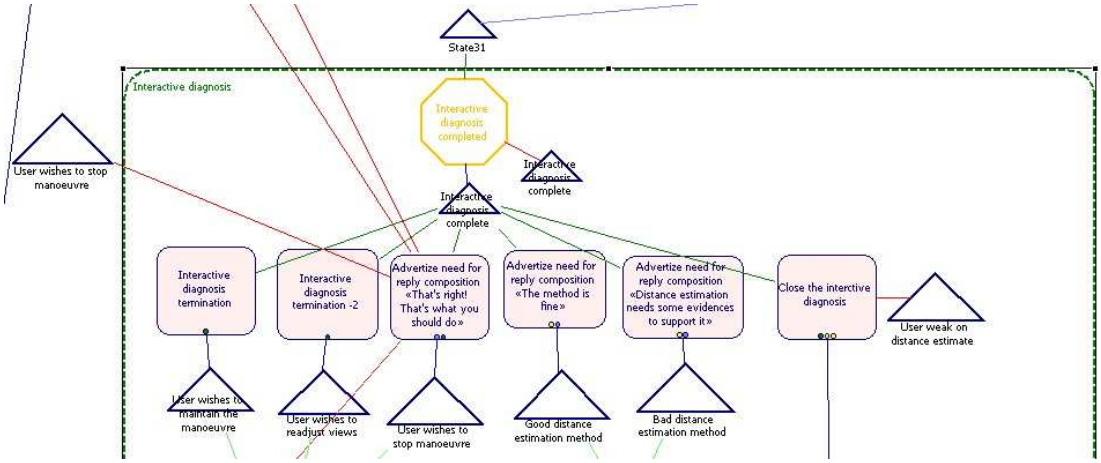
# THE INTERACTIVE DIAGNOSIS STREAM – LOWER PART



# THE INTERACTIVE DIAGNOSIS STREAM – MIDDLE PART



# THE INTERACTIVE DIAGNOSIS STREAM – UPPER PART



# REFERENCES


1. Anderson, John R. (1993) *Rules of the Mind*. Hillsdale, NJ: Erlbaum.

2. Anderson, John R., Dan Bothell*, et al.* (2004) "An integrated theory of the mind" *In Psychological Review*, vol. 111, no. 4, pp. 1036-1060.

3. Anderson, John R. and Kevin Gluck. (2001) "What role do cognitive architectures play in intelligent tutoring systems?" *In Cognition & Instruction: Twenty-five years of progress*. D. K. S. M. Carver Erlbaum, pp. 227-262.

4. Baader, F. & Nutt, W. (2003) Basic description logics. *In* F. Baader, D. Cavanese, D. McGuinness, D. Nardi, & P. Patel-Schneider (Eds.), *The description logic handbook: Theory, implementation and applications*, Cambridge University Press. pp. 47-100.

5. Baars, Bernard J. (1988) *A Cognitive Theory of Consciousness*. New York: Cambridge University Press.

6. Baars, Bernard J. (1997a) "In the theater of consciousness: Global Workspace Theory, A Rigorous Scientific Theory of Consciousness." *In Journal of Consciousness Studies*, vol. 4, no. 4, pp.292-309.

7. Baars, Bernard J. (1997b) *In the theater of consciousness: The Workspace of the Mind*. New York, NY: Oxford University Press.

8. Baars, Bernard J. (2002) "The conscious access hypothesis: origins and recent evidence" *In TRENDS in Cognitive Sciences*, vol. 6, no. 1, pp. 47-52.

9. Baars, Bernard J. and Stan Franklin (2003) "How conscious experience and working memory interact". *In TRENDS in Cognitive Sciences*. vol. 7, pp.166-172.

10. Baars, Bernard J., Uma Ramamurthy and Stan Franklin (In Press (2006)) "How deliberate, spontaneous and unwanted memories emerge in a computational model of consciousness". *In Involuntary Conscious Memory*. J. H. Mace (Ed.). Malden, MA: Blackwell Publishing.

11. Baars, Bernard J., Thomas Z. Ramsoy and Steven Laureys (2003) "Brain, conscious experience and the observing self" *In TRENDS in Neurosciences*, vol. 26, no. 12, pp. 671-675.

12. Baddeley, A. D., and G. J. Hitch (1974) Working memory. *In The Psychology of Learning and Motivation*, ed. G. A. Bower. New York: Academic Press.

13. Bechtel, William (1995) "Consciousness: Perspectives from symbolic and connectionist AI". *In Neuropsychologia*, vol. 33, pp.1075-1086.

14. Block, Ned (1995) "On a Confusion about a Function of Consciousness". *In The Behavioral and Brain Sciences*, vol. 18.

15. Block, Ned (2002) Some Concepts of Consciousness. *In Philosophy of Mind: Classical and Contemporary Readings* (David Chalmers, Eds), Oxford University Press.

16. Block, Ned (2003) *Consciousness, Philosophical Issues about*, nature publishing group.

17. Bratman, Michael E., D. Israel*, et al.* (1988) "Plans and Resource-Bounded Practical Reasoning." *In Computational Intelligence*, vol. 4, no. 4, pp. 349-355.

18. Brown, A. L. (1987) "Metacognition, executive control, self-regulation, and other more mysterious mechanisms". *In Metacognition, motivation, and understanding*. F. E. Weinert and R. Kluwe (Ed.). Hilsdale (Jew Jersey): Lawrence Erlbaum Associates, pp. 65-116.

19. Carbonell, Jaime G., Craig A. Knoblock, and Steven Minton (1990) "PRODIGY: An integrated architecture for planning and learning". *In Architectures for Intelligence* (Kurt VanLehn, ed.).Hillsdale, NJ: Erlbaum.

20. Carpenter, Gail and Stephen Grossberg (1987) "A massively parallel architecture for a self-organizing neural pattern recognition machine." *In Computer Vision, Graphics, and Image Processing,* no. 37, pp. 54-115.

21. Carpenter, Gail and Stephen Grossberg (2003) "Adaptive resonance theory". *In The Handbook of* Brain *Theory and Neural Networks, Second Edition* (Michael A. Arbib, Eds). Cambridge, Massachusetts:MIT Press, pp. 87-90.

22. Cazenave, Tristan (1998) "Machine Self-Consciousness More Efficient Than Human Self-Consciousness?" *In European Meeting on Cybernetics and Systems Research*, Vienne.

23. Chalmers, David J. (1995) "Facing up to the problem of consciousness." *In Journal of Consciousness Studies*, vol. 2, no. 3, pp. 200-219.

24. Chuderski, Adam, Zbigniew Stettner, et al. (2006) "Modeling individual differences in working memory search task". *In Seventh International Conference on Cognitive Modeling*, Trieste, Italy, pp. 74-79

25. Cotterill, Rodney M.J. (2003) CyberChild: a simulation test : a simulation test-bed for consciousness bed for consciousness studies. *In Journal of Consciousness Studies*, vol. 10, no 4-5, pp. 31-45

26. Crick, Francis and Christof Koch (1990) "Towards a neurobiological theory of consciousness". *In Seminars in the Neurosciences*. Academic Press, vol. 2, pp. 263-275.

27. Crick, Francis and Christof Koch (2003) "A framework for consciousness" *In Nature neuroscience*, vol. 6, no. 2, pp.119-126.

28. Damasio, Antonio R. (1990) "Synchronous activation in multiple cortical regions: a mechanism for recall" *In Seminars in Neuroscience*, vol. 2, pp.87-296.

29. Davis, Darryl N. (2002) "Architectures for cognitive and a-life agents". *In Intelligent Agent Software Engineering*. V. Plekhanova and S. Wermter (Ed.) Idea Group Publishing.
http://www2.dcs.hull.ac.uk/NEAT/dnd/papers/IASE-Final.pdf

30. Dehaene, Stanislas and Lionel Naccache (2001) "Towards a cognitive neuroscience of consciousness: Basic evidence and a workspace framework". *In Cognition*. vol. 79, pp. 1-37.

31. Dennett, Daniel. (1991) *Consciousness Explained*. (1st edition) New York: Little Brown & Co. 511 p.

32. d'Inverno, M., D. Kinny*, et al.* (1997) "A Formal Specification of dMars". *In Agent Theories, Architectures, and Languages*(Ed.), pp.155-176.

33. D'Mello, Sidney K., Uma Ramamurthy and Stan Franklin (2006) "Realizing Forgetting in a Modified Sparse Distributed Memory System". *In 28th Annual Meeting of the Cognitive Science Society*, Vancouver, Canada, pp. 1992-1997.

34. du Boulay, B. and R. Luckin (2001) "Modelling Human Teaching Tactics and Strategies for Tutoring Systems." *In International Journal of Artificial Intelligence in Education*, vol. 12, no. 3, pp. 235-256.

35. Dubois, Daniel (2005) "Consciousness for an ITS: It's a Naturel". *In AI-CBT'2005: Proceedings of the Workshop on Formal AI Techniques in Computer-Based Training*, Victoria, BC, Canada, May 8, 2005, pp. 7-17.

36. Edelman, Gerald M. (1989) *The Remembered Present: A Biological Theory of Consciousness*. New York: Basic Books

37. Edelman, Gerald (1992) *Bright Air, Brilliant Fire: On the Matter of the Mind*. New York: Basic Books, 280 p.

38. Elinas, Pantelis, Jesse Hoey and James J. Little (2003) "HOMER: Human Oriented MEssenger Robot", *In Proceedings of AAAI Spring Symposium on Human Interaction with Autonomous Systems in Complex Environments*, Stanford CA, March 2003.

39. Engel, Andreas K. and Wolf Singer (2001) "Temporal binding and the neural correlates of sensory awareness" *In TRENDS in Cognitive Sciences*, vol. 5, no. 1, pp. 16-25.

40. Ericsson, K. Anders and W. Kintsch (1995) "Long-term working memory" *In Psychological Review*, no. 102, pp. 211-245.

41. Evens, M. W., S. Brandle, *et al*. (2001). "CIRCSIM-Tutor: An Intelligent Tutoring System Using Natural Language Dialogue." *In Twelfth Midwest AI and Cognitive Science Conference, MAICS 2001*, Oxford, OH, pp. 16-23.

42. Faghihi, Usef (2007) *Adding new modes of learning in CTS, a "conscious" cognitive agent*. Master Thesis. UQAM.

43. Feldman Barrett, Lisa, Michele M. Tugade and Randall W. Engle (2004) "Individual Differences in Working Memory Capacity and Dual-Processing Theories of the Mind" *In Psychological Bulletin*, vol. 130, no. 4, pp. 553-573. http://www2.bc.edu/~barretli/pubs/2004/WMC.PsychBull.pdf. Retrieved on November 22, 2006

44. Flavell, J. H. (1979) "Metacognition and cognitive monitoring". *In American Psychologists*. vol. 34, pp.906-911.

45. Fournier-Viger, Philippe, Mehdi Najjar*, et al.* (2006) "From Black-Box learning Objects to Glass-Box Learning Objects". *In 8th International Conference, ITS 2006*, Jhongli, Taiwan, Springer, pp.258-267.

46. Franklin, Stan (1995) *Artificial Minds*. Cambridge, Massachusetts: MIT Press.

47. Franklin, Stan (2003a) "A computer-based model of Crick and Koch's Framework for Consciousness" *In Science&Consciousness Review*, no. 1.

48. Franklin, Stan (2003b) "IDA: A Conscious Artifact?" *In Journal of Consciousness Studies*, vol. 10, no. 4-5, pp. 47-66.

49. Franklin, Stan (2005) "A "Consciousness" Based Architecture for a Functioning Mind". *In Visions of Mind*. D. Davis (Ed.). Hershey, PA: IDEA Group, Inc.

50. Franklin, Stan (2005b) "Cognitive Robots: Perceptual associative memory and learning". *In 14th Annual International Workshop on Robot and Human Interactive Communication (RO-MAN 2005)*, Nashville, TN, pp.427-433.

51. Franklin, Stan, Bernard J. Baars*, et al.* (2005) "The Role of Consciousness in Memory". *In Brains, Minds and Media*. vol. 1.

52. Franklin, Stan and Art Graesser (1999) "A software agent model of consciousness"*. In Consciousness and Cognition*, no. 8, 285-301

53. Franklin, Stan and Lee McCauley (2004) "Feelings and Emotions as Motivators and Learning Facilitators: Architectures for Modeling Emotions". *In AAAI Spring Symposia : Architectures for Modeling Emotions*.

54. Franklin, Stan and Uma Ramamurthy (2006) "Motivations, values and Emotions: 3 sides of the same coin". *In Sixth International Workshop on Epigenetic Robotics*, Paris (France), Lund University Cognitive Studies, pp. 41-48.

55. Gaha, Mohamed (2007) Implementing modifications and extensions to CTS. Master Thesis (forthcoming).

56. Gamma, Claudia (2000) "The Reflection Assistant: Investigating the Effects of Reflective Activities in Problem Solving Environments". *In ED-MEDIA 2000*, Montreal, pp. 316-322.

57. Gray, W. D., M. J. Schoelles*, et al.* (2003 (in press)) "Meeting Newell's Other Challenge: Cognitive Architectures as the Basis for Cognitive Engineering." *In Behavioral & Brain Sciences*. On line. http://www.cogsci.rpi.edu/cogworks/publications/116_BBS_rsp-to-A&L_v9.pdf. Retrieved on November 2, 2006.

58. Georgeff, Michael, Barney Pell, *et al*. (1999) The Belief-Desire-Intention Model of Agency*. In 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures, and Languages (ATAL-98),* Paris: Springer-Verlag.

59. Georgeff, M. P. and F. F. Ingrand (1989) "Decision-Making in an Embedded Reasoning System". *In Eleventh International Joint Conference on Artificial Intelligence*, Detroit, Michigan.

60. Graesser, A., P. Chipman, *et al*. (2005). "AutoTutor: An intelligent tutoring system with mixed-initiative dialogue." *In IEEE Transactions in Education* no 48, pp. 612-618.

61. Graesser, A. C., G. T. Jackson, *et al*. (2003) "Why/AutoTutor: A Test of Learning Gains from a Physics Tutor with Natural Language Dialog"*. In 25th Annual Conference of the Cognitive Science Society*, Mahwah, NJ: Erlbaum.

62. Graesser, A. C., N. Person, *et al*. (2005) "Learning while holding a conversation with a computer". *In Technology-based education: Bringing researchers and practitioners together*. L. PytlikZillig, M. Bodvarsson and R. Bruning. Greenwich, CT: Information Age Publishing**,** pp. 143-167.

63. Grossberg, Stephen (1976) "Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors & II: Feedback, expectation, olfaction, and illusions". *In Biological Cybernetics*, no. 23, pp.121-134 & 187-202.

64. Grossberg, Stephen (1999) "The Link between Brain Learning, Attention, and Consciousness". In *Consciousness and Cognition*, vol. 8, pp. 1-44.

65. Grossberg, Stephen (2005) Linking attention to learning, expectation, competition, and consciousness. *In Neurobiology of attention* (L. Itti, G. Rees and J. Tsotso, Eds). San Diego, Elsevier, pp. 652-662.

66. Harnad, Steven (2003) "Can a Machine Be Conscious? How?" *In Journal of Consciousness Studies*, vol. 10, no. 4-5, pp. 69-75.

67. Hexmoor, Henry H., Johan M. Lammens, *et al*. (1993) "An autonomous agent architecture for integrating "unconscious" and "conscious", reasoned behaviors". *In Computer Architectures for Machine Perception*, IEEE Computer Society Press., pp. 328-336

68. Hofstadter, D. R. and M. Mitchell (1995). "The copycat project: A model of mental fluidity and analogy-making". *In Advances in connectionist and neural computation theory, vol. 2: Logical connections*, ed. K J Holyoak and J A O. Role of Editor eds Barnden:205–267. Norwood N.J.: Ablex.

69. Hohmeyer, Patrick (2006) *Développement d'une architecture d'agent conscient pour un système tutoriel intelligent.* Master Thesis. Université du Québec à Montréal.

70. Jackson, John V. (1987) "Idea For A Mind". *In SIGART Newsletter*. vol., pp.23-26.

71. James, William (1892) "The Stream of Consciousness". *In Psychology (chap. XI)*.

72. Jennings, N. R. and M. Wooldridge (1998) "Application of Intelligent Agents". *In Agent Technology: Foundations, Applications, and Markets*. N. R. Jennings and M. J. Wooldridge (Ed.). Heidelberg, Germany: Springer-Verlag, pp. 3-28.

73. Jiang, H. and J. M. Vidal (2006, in press) "From Rational to Emotional Agents". *In AAAI Workshop on Cognitive Modeling and Agent-based Social Simulation*, Menlo Park, California, AAAI Press.

74. Johnson-Laird, P.N. (1983) "A computational analysis of consciousness". *In Cognition and Brain Theory*, no. 6, pp.499-508.

75. Johnson-Laird, P. N. (1988) "A computational analysis of consciousness". *In Consciousness in Contemporary Science*. A. E. Marcel and E. Bisiach (Ed.). Oxford: Clarendon.

76. Johnston, V. S. (1999) “Why We Feel: The Science of Human Emotions”. Reading, MA: Perseus Books.

77. Kabanza, Froduald, Roger Nkambou, Khaled Belghith and Leo Hartman (2005). “Path-Planning for Autonomous Training on Robot Manipulators in Space.” *In Proceedings of IJCAI’2005*.

78. Kanerva, Pentti (1988) *Sparse Distributed Memory*. Cambridge, Mass.: MIT Press.

79. Kanerva, Pentti (1993) "Sparse Distributed Memory and related models". *In Associative Neural Memories: Theory and Implementations*. M. H. Hassoun (Ed.). New York: Oxford University Press, pp. 50-76.

80. Kanerva, Pentti (1997) "Fully distributed representation". *In 1997 Real World Computing Symposium (RWC'97)*, Tokyo: Real World Computing Parthership, pp. 358-365.

81. Katz, S., A. Lesgold, E. Hughes, D. Peters, G. Eggan, M. Gordin, and L. Greenberg. (1998) "Sherlock 2: An intelligent tutoring system built on the LRDC framework". *In Facilitating the Development and Use of Interactive Learning Environments*. C. P. Bloom & R. B. Loftin (Eds.), Mahwah, NJ: Erlbaum, pp. 227-258

82. Langley, P., K. B. McKusick, J. A. Allen, W. F. Iba, and K. Thompson (1991) "A design for the Icarus architecture", In SIGART Bulletin, no. 2, pp. 104-109.

83. Lebiere, C., M. D. Byrne, *et al*. (2004) "An integrated theory of the mind." *In Psychological Review*, vol. 111, no. 4, pp. 1036-1060.

84. Lesgold, A. M., S. P. Lajoie, *et al*. (1992). “Sherlock: A coached practice environment for an electronics troubleshooting job”. *In Sherlock: A coached practice environment for an electronics troubleshooting job*. (J. Larking and R. Chabay, Eds.). Hillsdale, N. J., Lawrence Erlbaum Assoc.

85. Libet, Benjamin, C. A. Gleason*, et al.* (1983) "Time of Conscious Intention to Act in Relation to Onset of Cerebral Activity (Readiness-Potential)" *In Brain*, vol. 106, no., pp. 623-642.

86. Maes, Pattie (1989) "How to Do the Right Thing" *In Connection Science Journal*, vol. 1, no. 3, pp. 291-323.

87. Mayers, A., B. Lefebvre*, et al.* (2001). "Miace: A Human Cognitive Architecture." *In sigcue outlook*, vol. 27, no. 2, pp. 61-77.

88. McCarthy, John (1959) "Programs with common sense". In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, On line. London: Her Majesty's Stationary Office, pp.75-91. http://www-formal.stanford.edu/jmc/mcc59.html. Retrieved on September 30, 2006.

89. McCarthy, John. (1995) and (2002). "Making Robots Conscious of their Mental States". [on line]
1995: http://citeseer.ist.psu.edu/cache/papers/cs/3566/http:zSzzSzwww-formal.stanford.eduzSzjmczSzconsciousness.pdf/mccarthy95making.pdf 2002: http://www-formal.stanford.edu/jmc/consciousness.ps

90. Minsky, Marvin (1961) "Steps Towards Artificial Intelligence". *In Proceedings of IRE*, On line. vol. 49, no. 1, Jan. 1961, pp. 8-30. http://web.media.mit.edu/~minsky/papers/steps.html. Retrieved on September 30, 2006.

91. Minsky, Marvin (1965) "Matter, Mind and Models". *In Proceedings of the International Federation of Information Processing Congress*, vol. 1, pp.45-49

92. Minsky, Marvin (1985) *The society of mind*. New York, NY: Simon & Schuster/Touchstone.

93. Minsky, M. (1998) "Consciousness is a Big Suitcase: A Talk with Marvin Minsky." *In Edge*, February 1998. On line. http://www.edge.org/3rd_culture/minsky/minsky_p2.html. Retrieved on November 8, 2006.

94. Nagel, Thomas (1974) "What is it like to be a bat?" *In The Philosophical Review*. vol. 83, pp.4 35-450.

95. NASA (2000) "International Space Station Evolution Data Book: Volume I. Baseline Design: Revision A". FDC/NYMA, pp.222.

96. Negatu, Aregahegn S. and Stan Franklin (2002) "An Action Selection Mechanism for "Conscious" Software Agents". *In Cognitive Science Quarterly*. vol. 2, pp. 363-386.

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

98. Nkambou, Roger, Khaled Belghith and Froduald Kabanza (2006) "An Approach to Intelligent Training on a Robotic Simulator using an Innovative Path-Planner". *In 8th International Conference, ITS 2006*, Jhongli, Taiwan, Springer-Verlag, pp. 645-654.

99. Paillard, Jacques (1999) "L'approche neurobiologique des faits de conscience : vers une science de l'esprit", *In Psychologie française*, no 44-3, pp. 245-256.

100. Picard, Rosalind W. (2000) "Towards computers that recognize and respond to user emotion" *In IBM Systems Journal*, vol. 39, no. 3 & 4.

101. Piaget, Jean (1970) *L'épistémologie génétique*. Paris : Presses universitaires de France. Coll. Que sais-je?, no 1399, 127 p.

102. Ramamurthy, Uma, Sidney K. D'Mello and Stan Franklin (2006) "LIDA: A Working Model of Cognition". *In 7th International Conference on Cognitive Modeling*, Trieste, Italy, Edizioni Goliardiche, pp. 244-249.

103. Revoy, Nicolas and Philippe Chambon (2006) "La science aux portes de la conscience*"*. *In Science & vie*, no. 1062, pp. 56-71.

104. Searle, John (1980) "Minds, Brains, and Programs." *In Behavioral and Brain Sciences*, no 3, pp. 417-424.

105. Selfridge, Oliver G. (1959) "Pandemonium: A paradigm for learning". *In Symposium on Mechanisation of Thought Processes*, H. M. Stationary Office, pp. 511-529.

106. Shaw, E., W. L. Johnson, *et al*. (1999) "Pedagogical Agents on the Web." *In Third International Conference on Autonomous Agents*.

107. Sloman, A. (1999) "What sort of architecture is required for a human-like agent?" *In Foundations of Rational Agency* (Michael Woolridge and Anand Rao, Eds), Kluwer Academic Publishers, Portland, Oregon.

108. Sloman, Aaron and Ron Chrisley (2003) "Virtual machines and consciousness." *In Journal of Consciousness Studies*, vol.10, nos (4-5), pp. 133-172.

109. "Stottler Henke to enhance intelligent tutoring system for U.S. Navy". *In Military & Aerospace Electronics Magazine*. On line. PennWell Corporation. http://mae.pennnet.com/Articles/Article_Display.cfm?ARTICLE_ID=252989&p=32. Retrieved September 30, 2006.

110. Sun, Ron (1997) "Learning, action, and consciousness: a hybrid approach towards modeling consciousness" *In Neural Networks, special issue on consciousness*. vol. 10, no 7, pp. 1317-1331.

111. Taylor, John G. (2000) "The Enchanting Subject Of Consciousness (Or Is It A Black Hole?): Review of Enchanted Looms: Conscious Networks In Brains and Computers By Rodney Cotterill". On line. <http://psyche.cs.monash.edu.au/v6/psyche-6-02-taylor.html>. Retrieved on September 30, 2006.

112. Taatgen, N.A. (submitted). *Consciousness in ACT-R*. Submitted to the OUP companion to Consciousness. On line. http://www.ai.rug.nl/~niels/publications/ACT-R-consciousness.pdf. Retrieved on November 2, 2006.

113. Tononi, G. and Gerald M. Edelman (1998) "Consciousness and complexity" *In Sciences*, vol. 282, pp. 1846-1851.

114. VanLehn, K. and W. Ball. (1991) "Goal Reconstruction: How Teton Blends Situated Action and Planned Action", In Architectures for Intelligence. (K. VanLehn Ed.). Hillsdale, NJ: Lawrence Erlbaum Associates. pp. 147-189

115. VanLehn, K., C. Lynch, et al. (2005). "The Andes physics tutoring system: Five years of evaluations." *In* Artificial Intelligence in Education, Amsterdam, IOS.

116. Veloso, M., J. Carbonell, A. Perez, D. Borrajo, E. Fink, and J. Blythe, J. (1995) "Integrating planning and learning". In *Journal of Experimental and Theoretical Artificial* Intelligence, vol.7, no1.

117. Voss, Peter (2004) "Essentials of General Intelligence: The direct path to AGI". *In Real AI: New Approaches to Artificial General Intelligence*. Ben Goertzel and Cassio Pennachin (Ed.). On line. http://www.adaptiveai.com/research/index.htm#different_approach . Retrieved on November 11, 2006.

118. Yam, Phil (1998) "Intelligence Considered: What does it mean to have brain-power?: A search for a definition of intelligence". *In Scientific American Quarterly*. (Winter).