

# Evaluating Spatial Representations and Skills in a Simulator-Based Tutoring System

Philippe Fournier-Viger, *IEEE Member*, Roger Nkambou, *IEEE Member* and André Mayers

**Abstract**— Performing exercises in a simulation-based environment is a convenient and cost-effective way of learning spatial tasks. However, training systems that offer such environments lack models for the assessment of learner's spatial representations and skills, which would allow the automatic generation of customized training scenarios and assistance. Our proposal aims at filling this gap by extending a model for representing learner's cognitive processes in tutoring systems, based on findings from research on spatial cognition. This article describes how the model is applied to represent knowledge handled in complex and demanding tasks, namely the manipulation of the robotic arm Canadarm2, and more specifically, how a training system for Canadarm2 manipulation benefits from this model, both by its ability to assess spatial representations and skills and to generate customized assistance and exercises.

**Index Terms**—Computer-assisted instruction, Intelligent tutoring systems, Cognitive modeling, Spatial cognition

## 1 INTRODUCTION

Numerous activities, such as driving a car, involve spatial abilities, representations and skills. Spatial abilities and skills represent both ends of a continuum. The former is more general and can be applied to various domains such as mentally performing object rotations or change of perspectives (for example, see [1], [2]). The second refers to the specialized skills that involve correct manipulations of spatial representations to achieve specific goals. Although spatial skills and representations are generally learned with hands-on activities, they can often be reified as declarative knowledge that can be explicitly manipulated in reasoning and communicated through descriptions such as road maps or route instructions [3]. Learning such skills or representations can be a time consuming task. To reduce training costs and reproduce complex scenarios, training can be conducted with simulations. For this purpose, very realistic immersive environments have been developed: to train soldiers for coordinated group tasks, for instance [4]. However, simulator-based training fails to provide an optimal learning experience, as training scenarios and feedback are not tailored to learners' spatial representations, abilities and skills. Such a situation can be improved in two different ways. The first way consists of having experts observe learners being trained and asking them to intervene in the training session to create customized learning activities. The second way, which is explored in this paper, consists of enhancing a training system with techniques derived from the field of Intelligent

Tutoring Systems (ITS) to evaluate learners' spatial representations and skills, and generate customized scenarios and assistance. Although software that provides customized assistance for virtual learning activities has been investigated for over 20 years [5], the problem of building simulator-based training systems that provide assistance which is customized to learners' spatial representations, abilities or skills remains to be addressed by researchers.

The goal of this paper is (1) to stress the importance and raise the challenge of building tutoring systems to acquire spatial skills and representations in an effective manner and (2) to show how a specific cognitive model for student modeling in tutoring systems was extended to assess spatial representations and skills. The paper is organized as follows. First, it introduces RomanTutor, a tutoring system for learners being trained to operate the Canadarm2 robotized arm attached to the International Space Station (ISS). Then, it reports relevant findings from the field of spatial cognition. Next, the paper describes a cognitive model, its extension to model the recall and use of spatial knowledge, and how the model applies to RomanTutor to support tutoring services. Then, results from an experimental evaluation are reported. Finally, the paper discusses related work, presents conclusions and previews our future work.

## 2 THE ROMANTUTOR TUTORING SYSTEM

Manipulating the Canadarm2 attached to the ISS consists of a complex task that relies on complex spatial representations. Canadarm2 is a robotic arm equipped with seven degrees of freedom (depicted in Figure 1). Handling such technology represents a demanding responsibility since the astronauts who control it have a limited view of the environment. The environment is rendered through only three monitors, each showing the view obtained from a single camera while about ten cameras are mounted at different locations on the ISS and on the arm. Guiding a robot

- P. Fournier-Viger is affiliated with the Department of Computer Science, University of Quebec in Montreal, Montreal, Canada. Email: Fournier\_viger.philippe@courrier.uqam.ca.
- R. Nkambou is affiliated with the Department of Computer Science, University of Quebec in Montreal, Montreal, Canada. Email: nkambou.roger@uqam.ca.
- André Mayers is affiliated with the Department of Computer Science, University of Sherbrooke, Sherbrooke, Canada. Email: andre.mayers@usherbrooke.ca.



perspective. In order to be useful, egocentric representations must be constantly updated, according to one's perception. This process, called "path integration", is supported by robust experimental evidence [12], [13]. For instance, researchers have observed that ants can return directly to their initial position, even after traveling over 100 meters on a flat field [12]. In the context of route navigation, egocentric representations describe relative landmarks along a route to follow, and navigation consists of performing the correct movements when each landmark is reached [3]. Egocentric knowledge is usually gained from experience, but it can also be acquired directly from descriptions, such as textual route instructions, for instance. Route navigation is extremely inflexible and leaves little room for deviation. Indeed, choosing correct directions with landmarks depends on a person's relative position. Consequently, path deviations can easily disturb the outcome of the entire navigation task. Incorrect encoding or recall can also seriously compromise the achievement of the goal.

According to Tversky [3], egocentric representations may be sufficient to perform tasks such as navigating through an environment, but they are inadequate to perform complex spatial reasoning. For reasoning that requires inference, humans build cognitive maps that do not preserve measurements, but instead retain the main relationships between elements. Such representations do not encode a single perspective, yet they make it possible to adopt several perspectives. Cognitive maps are also prone to encoding or recall errors. However, recovering from errors is generally easier when relying on cognitive maps than egocentric representations. Moreover, it is more efficient to update an allocentric representation, since only a single self position changes. Note that some researchers suggest that humans can also rely on a "semi-allocentric" frame of reference [14], where the position of an object is encoded according to a salient axis of the environment, such as the one formed by two distant buildings, providing such axes are available.

Recently, place cells have been discovered in human hippocampi [15]. Observing rat nerve cells also lead to the discovery of head-direction cells [16], [17], speed modulated place cells [18] and cells with periodic place fields [19]. In light of such work, as well as other research carried out over the last few decades in neuroscience, neuropsychology [20], [21], [22], experimental psychology and other disciplines, there is no doubt that humans use both allocentric and egocentric space representations [23], [24].

### 3.2 How can spatial representations be implemented in a computational model?

The second important question that must be considered to develop a cognitive task description that considers spatial representations and skills is how to implement spatial representation in a computational model. Part of the answer to this question can be found in models derived from experimental spatial cognition research. However, such models generally specialize in a particular phenomenon, such as visual perception and motion recognition [25], navigation in 3D environments [19], [26] and mental

imagery and inference from spatial descriptions [27]. The few models that attempt to give a more general explanation of spatial cognition have no or only partial computational implementation as in [28] for example. Most cognitive models of spatial cognition can be categorized as providing structures to model cognitive processes either at a symbolic or a neural level. Models based on neural networks can simulate low level behavior such as the activation patterns of head-direction cells [29] and the integration of paths with cognitive maps with considerable success [19]. However, connectionist models are not convenient for tutoring systems, wherein knowledge must be explicit. Alternatively and with certain particularities, symbolic models that rely on allocentric representations [25], [27], [28] usually represent spatial relations as links of type "a r b" where "r" depicts a spatial relation such as "is on the left of" or "is on top of" and where "a" and "b" translate into mental representations of objects. Each allocentric representation is encoded according to a reference frame that is either implicit or explicit: a coordinate system, for example. This representation of cognitive maps reflects the work of psychology researchers such as Tversky [3], who suggests that cognitive maps are encoded as sets of spatial relationships in semantic memory. On the other hand, an egocentric representation is usually represented by a relation of type "s r o" where "r" denotes a spatial relation, and "s" and "o" denote mental representations of the self and of an object, respectively.

After studying symbolic models, it was decided not to choose models that explain specific phenomena of cognition, such as [25] and [27], as the foundation of our work in RomanTutor. It was decided rather to rely on a unified theory of cognition, as it provides a more global understanding of cognitive performance. We rejected the Gunzelmann and Lyon model [28], which is based on a global theory of cognition, since it lacks computational implementation. Another important issue regarding the aforementioned models is that they have yet to be implemented within a tutoring system. In fact, they would require major changes to support the acquisition of spatial representations and skills in a tutoring system, as this context generates particular challenges. The main difficulty consists of developing mechanisms to evaluate learners, given the limited communication channels offered by tutoring systems (usually a keyboard, a mouse and a monitor) [5]. Therefore, as a starting point, it was decided to select an existing cognitive model designed for tutoring systems [30], which is based on a global theory of cognition, and to adapt this model according to results obtained in the field of spatial cognition.

## 4 THE PROPOSAL

Our proposal is described in three parts. First, the initial cognitive model is introduced. Second, we present its extension. Then, software modules for simulating the dynamic aspects of the model and exploiting it in a tutoring context are described.

#### 4.1 The Initial Model

For the initial model, we chose a model for describing cognitive processes in tutoring systems that we have developed in previous research [30]. Inspired by the ACT-R [31] and Miace [32] cognitive theories, this symbolic model organizes knowledge in two distinct categories: semantic [33] and procedural [31]. This subsection describes the cognitive theory behind this model and its computational representation.

Semantic knowledge represents declarative knowledge which is not associated with the memory of events [34]. The model regards semantic knowledge as concepts taken in the broad sense. According to recent research [35], humans can consider up to four instances of concepts to perform a task. However, through the process of chunking [35], the human cognitive architecture can group several instances together and treat them as a single one. These syntactically decomposable representations are called “described concepts”, unlike “primitive concepts” which are syntactically broken up. For example, the expression “PMA03 isConnectedToTheBottomOf Lab02” consists of a decomposable representation that contains three primitive concept instances, representing the knowledge that the “PMA03” ISS module is connected at the bottom of the “Lab02” ISS module on the ISS, assuming a default three dimensional Cartesian coordinate system. This way, the semantics of a given described concept are revealed by the semantics of its components. While concepts are stored in the semantic memory, concept instances, which are manipulated by cognitive processes, are stored in working memory, and they are characterized by their mental and temporal contexts [32]. Thus, each occurrence of a symbol such as “Lab02” is viewed as a distinct instance of the same concept.

Procedural knowledge encodes knowledge pertaining to ways of automatically reaching goals by manipulating semantic knowledge. It is composed of procedures which are triggered one at a time, according to the current state (goal) of the cognitive architecture [31]. Contrary to semantic knowledge, the activation of a procedure does not require attention. For instance, one procedure could be recalling which camera is best located to view a specific ISS module. Another procedure would consist of selecting a camera from the user interface of RomanTutor. Just as the model of Mayers et al. [32], this work differentiates primitive from complex procedures, whereas primitive procedures embody atomic actions, activating a complex procedure instantiates a set of goals, which may be reached through either complex or primitive procedures.

Goals are defined as intentions that humans have, such as the goal of solving a mathematical equation, drawing a triangle or adding two numbers [32]. At every moment, the cognitive architecture has a goal that represents its intention. This goal is chosen among the set of active goals [36]. There are numerous correct and erroneous ways (procedures) to achieve a goal. The model considers goals as a special type of described concepts, instantiated with zero or more concept instances, which consist of goal parameters. For example, the concept instance “Cupo-

la01” could be a component of an instance of the goal “GoalSelectCamerasForViewingModule”, which represents the intention of selecting the best camera to view the “Cupola01” ISS module. The parameters of a goal instance can restrict the procedures that can be triggered. Goal parameters also represent ways to transfer semantic knowledge between the complex procedure that creates a goal and the procedure that will achieve the goal.

A computational structure [30] was developed to describe the cognitive processes of a learning activity, according to the cognitive theory described above. The computational structure defines sets of attributes to describe concepts, goals and procedures. The value of an attribute can consist of a series of concepts, goals, procedures, or arbitrary data such as character strings.

A primitive concept has an attribute called the “DL Reference”. This last attribute contains a logical description of the concept. It allows inferring a subsumption relation between concepts (see [30] for more details). Described concepts bear an additional attribute, called “Components”, to specify the concept type for each of its components.

Goals have two main attributes. “Parameters” indicate the concept types of the goal parameters. “Procedures” enumerate a set of procedures that can be used to achieve the goal.

Procedures have six main attributes. “Goal” indicates the goal for which the procedure is defined. “Parameters” specifies the concept type of the arguments. For primitive procedures, “Method” points to a Java method that executes an atomic action. With regard to primitive procedures, the “Observable” attribute specifies whether the procedure corresponds to a user’s action in the interface or a mental step. For complex procedures, “Script” indicates a set of sub-goals to be achieved with one or more optional constraints in the order that they should be achieved. Lastly, “Validity” is used to determine whether the procedure is valid.

An authoring tool was developed. It was used to represent the learners’ cognitive processes when using a Boolean reduction rule tutoring system [30]. Although the model was successfully used to provide customized assistance to university students carrying out learning activities, the model emphasizes the acquisition of procedural knowledge rather than semantic knowledge. The reason for this is that the model assumed a perfect recall of semantic knowledge and, therefore, all mistakes are explained in terms of missing or erroneous procedural knowledge. The model did not include a process to simulate the retrieval of knowledge from semantic memory, a key feature of many cognitive theories. As a consequence, it was impossible to specify that in order to achieve a given goal, for instance, one must correctly recall the concept “CameraCP5 AttachedTo S1” (the spatial knowledge that Camera CP5 is attached to the ISS module called S1) and use it in a subsequent procedure. Evaluating semantic knowledge is essential to evaluating spatial representations and skills, if we consider that cognitive maps are encoded as semantic knowledge, as suggested by Tversky

[3] or as in models of spatial cognition such as [28], and that these spatial representations have to be recalled during procedural tasks.

## 4.2 The Extended Model

To address this issue, the model was extended. In particular, the extension adds pedagogical distinctions between "general" and "contextual" semantic knowledge. General knowledge is defined as the semantic knowledge (memorized or acquired from experience) that is valid in all situations of a curriculum. For instance, such knowledge includes the fact that the end effector of Canadarm2 measures approximately one meter. General knowledge comprises described concepts, as it must represent relations in order to be useful. To be used properly, general knowledge must be (1) properly acquired, (2) recalled correctly and (3) handled by valid procedural knowledge. When general knowledge is recalled, it is instantiated with predetermined components. On the other hand, contextual knowledge refers to the knowledge obtained by interpreting situations. It is composed of described concept instances which are dynamically instantiated in a situation. For example, the information stating that the rotation value of joint "WY" on the Canadarm2 currently equals 42°, consists of contextual knowledge obtained by reading a display.

Three attributes are added to described concepts. The "General" attribute indicates whether or not the concept is general. For general concepts, "Valid" specifies whether the concept is correct or erroneous, and optionally, the identifier of an equivalent valid concept (the model allows for the encoding of common erroneous knowledge). Moreover, for general concepts, the "Retrieval-Components" attribute specifies a set of concepts to be instantiated to create the concept components when the concept is recalled. Table 1 presents a concept encoding the knowledge that the spatial module "MPLM" is connected below module "NODE2" on the ISS (according to the default coordinate system). The "Valid" attribute reveals that it consists of erroneous knowledge and that the valid equivalent information consists of the concept "MPLM\_TopOf\_Node2" (c.f., Table 2). The "DLReference" attribute specifies that these concepts are subconcepts of "SpatialRelationBelow" and "SpatialRelationTopOf", respectively, which are themselves subconcepts of "SpatialRelationBetweenModules", the concept of spatial relation between two ISS modules.

A retrieval mechanism was also added to connect procedures and general knowledge, thus modeling the recall process. It functions similar to the retrieval mechanism of the ACT-R theory of cognition [31]. ACT-R was selected since the initial model is already based on that theory. With the addition of this retrieval mechanism, a procedure can now ask to retrieve a described concept by specifying one or more restrictions on the value of its components. In the extended model, this is accomplished by a new attribute called "Retrieval-request". It specifies the identifier of a described concept to be recalled and zero or more restrictions on the value of its components. Table 3 shows the procedure "RecallCameraForGlobalView". The

execution of this procedure requests the knowledge of the ISS camera that provides the best global view of a location taken as parameter by the procedure. The "Retrieval-request" attribute states that a concept of type "ConceptRelationCameraGlobalView" (a relation stating that a camera offers a global view of a given area) or one of its subconcepts is needed, and that its first component should be a location whose concept type match the type of the procedure parameter, and the second component needs to be of type "ConceptCamera" (a camera). A correct recall following the execution of this procedure will result in the creation of an instance of "ConceptRelationCameraGlobalView" that will be deposited in a special buffer that accepts the last recalled instance. Then, the procedures subsequently executed can access the concept instance in the buffer to achieve their goal. In the extended model, a procedure can be asked to trigger if and only if the retrieval buffer contains a specific type of concept instance. This is achieved by a "Retrieval-match" attribute that can specify constraints on general knowledge that should have been recalled. Although the extended model only allows retrieving general knowledge, it also allows the definition of primitive procedures that read information from the users' interface to simulate the

TABLE 1  
PARTIAL DEFINITION OF THE CONCEPT  
"MPLM\_BELOW\_NODE2"

Attributes	Values
DLReference	SpatialRelationBelow
Components	Module, Module
RetrievalComponents	MPLM, Node2
General	True
Valid	False (Valid=MPLM_TopOf_Node2)

TABLE 2  
PARTIAL DEFINITION OF THE CONCEPT  
"MPLM\_TOPOF\_NODE2"

Attributes	Values
DLReference	SpatialRelationTopOf
Components	Module, Module
RetrievalComponents	MPLM, Node2
General	True
Valid	True

acquisition of contextual knowledge perceived visually.

## 4.3 Tools to Simulate the Dynamic Aspects of the Cognitive Model

To simulate the dynamic aspects of the cognitive model, the "Cognitive Model Interpreter" module was developed. Furthermore, to exploit the model in a tutoring context, the "Model Tracer" module was created. This section describes these modules, which work closely together, and are the foundations for supporting elaborated tutoring services.

The interpreter consists of a software module that can simulate behaviors described by the cognitive model. Run by the Model Tracer module, its state is defined as a set of goals. At the beginning of a simulation, it contains a single goal. At the start of each cycle, the interpreter asks the Model Tracer to choose a current goal amongst the set of instantiated goals. The interpreter then determines which procedures can be executed for the goal. The Model Tracer must choose a procedure to continue the simulation. To execute a primitive procedure, the interpreter calculates the result before removing the goal from the set of goals. To execute a complex procedure, the interpreter instantiates the procedure subgoals and adds them to the set of instantiated goals. A goal achieved through a complex procedure is only removed from the list of goals once all of its subgoals have been reached. When executing a procedure request to retrieve semantic knowledge, if two or more options are possible, the interpreter asks the Model Tracer to select the most general concept to be retrieved. The interpreter stops when there are no goals left.

TABLE 3  
PARTIAL DEFINITION OF THE PROCEDURE "RECALLCAMERA-FOR-GLOBALVIEW"

Attributes	Values
Goal	GoalRecallCameraForGlobalView
Parameters	(ConceptPlace: P)
RetrievalRequest	ID: ConceptRelationCameraGlobalView A1:ConceptPlace: P, A2: ConceptCamera
General	True
Valid	True

In the context of learning activities, an author must specify one main goal for each problem-solving exercise. Simulations to solve exercises with the interpreter give rise to structures such as the one shown in Figure 4.a, where the main goal, "G", is achieved with a complex procedure named "CP1", which instantiates three subgoals. Whereas, the first goal, "G1", is reached by executing the complex procedure "CP2", goals "G2" and "G3" are achieved by the observable primitive procedure "OP1" and the complex procedure "CP3". The two subgoals of the procedure "CP2" are achieved by primitive procedures "PP1" and "PP2", respectively, and those of "CP3" are attained by the observable primitive procedures "OP2" and "OP3", respectively. Only primitive procedures tagged as observable correspond to actions conducted within the user interface. Figure 4.b shows the list of procedures that would be visible for the structure presented in Figure 4.a. The following paragraph explains how the "Model Tracer" module makes it possible to follow a learner's reasoning using its visible actions.

The main task of the Model Tracer consists of finding goal/sub-goal structures such as the one shown in Figure 4.a to explain a sequence of learner's actions such as the one depicted in Figure 4.b. The input consists of a list  $S=\{p_1, p_2, \dots, p_n\}$  of observable primitive procedures executed along with their arguments. The algorithm proceeds as follows: It first launches the interpreter starting from the exercise main goal. When the interpreter offers a choice of several procedures, goals or retrieval options for different units of semantic knowledge, the

algorithm records the current state of the interpreter. Then, a single possibility is explored and the others are placed on a stack to be explored subsequently. In other words, the algorithm explores the state space of the various possibilities in a depth-first way. Every time the algorithm tries a possibility that matches a subsequence  $s$  of  $S$  (for example  $\{s_1, s_2\}$ ), without finding the subsequent action, or in cases where the following action is tagged as erroneous, the algorithm notes the current structure along with the subsequence  $s$ . After trying all possibilities, if a structure for  $S$  is not found, the goal/sub-goal structures that match the longest correct subsequence of  $S$  is returned. To make the search for goal/sub-goal structures more manageable, the algorithm does not explore the possibilities that match subsequences which are not subsequences of  $S$ , nor does it explore beyond the last correct step taken by the learner.

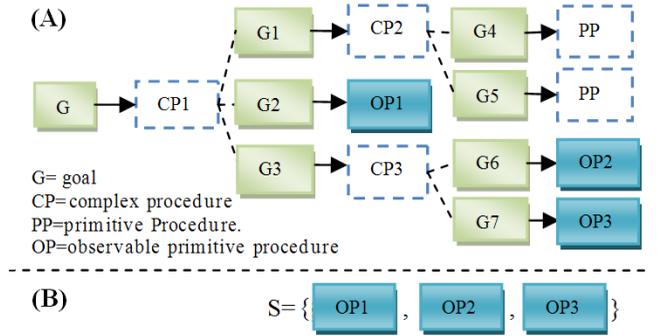


Fig.4. A goals/sub-goals structure for an exercise

## 5 HOW THE EXTENDED MODEL SUPPORT TUTORING SERVICES IN ROMANTUTOR

This section describes how the Canadarm2 manipulation task was modeled. Then, it explains how the cognitive model supports four important tutoring services in RomanTutor.

### 5.1 Modeling the Canadarm2 Manipulation Task

To model the Canadarm2 manipulation tasks, a cognitive task analysis was performed. The available documents pertaining to the ISS and Canadarm2 were studied. In addition, a member of our team had the opportunity to observe a week-long training session at the Canadian Space Agency, located in St-Hubert, Canada, in March 2006. Using his observations of training sessions on a simulator, a first draft of the cognitive steps required to move a load from one position to another with Canadarm2 was produced. Then, we listed the elements missing from the user interface of our simulator, such as options to select various speeds to move the arm.

We then describe the task with the cognitive model. To achieve this, the 3D space was first split into 3D sub-spaces called Elementary Spaces (ESs). The model incorporates 30 ESs. For the current exercise, we selected the hypothesis that the arm was locked into its mobile base and that the base could not move. Even with such a framework, manipulation tasks remain complex. After examining different possibilities, it was determined that the most realistic types of ES for mental processing are

ESs configured with an arm shape. Figure 5 illustrates 6 of the 30 ESs. For example, from ES 1, it is possible to obtain ES 2, ES 4 and ES 6. Each ES is composed of three cubes. The goal of the exercise and the initial state must be defined through an ES.

Nearly 120 concepts were defined, such as the various ISS modules, the main parts of Canadarm2, the cameras on the ISS and the ESs. Spatial relations were then encoded as described concepts such as (1) a camera gives a global or a detailed view of an ES or an ISS module, (2) an ES comprises an ISS module, (3) an ES is placed next to another ES, (4) an ISS module is next to another ISS module or (5) a camera is attached to an ISS module. This representation of allocentric spatial representations as relations encoded in semantic memory is in agreement with both the research on spatial cognition described in section 3 and computational models of spatial cognition where spatial knowledge is typically represented as relations of type "a r b".

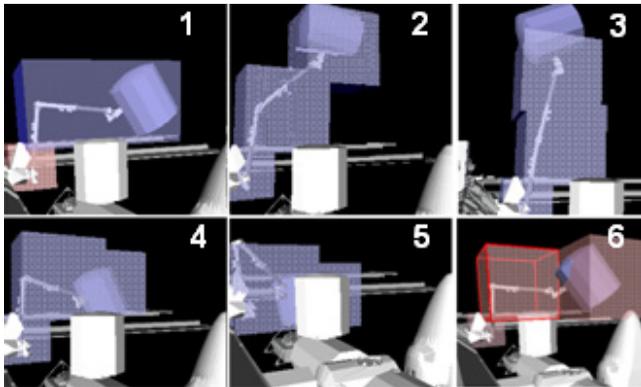


Fig.5. Six elementary spaces (ES)

Procedural knowledge to move the arm from one position to another is modeled as a loop, where learners must recall a set of cameras to view the ES that contains the arm, select a camera that provides a global view for the second monitor and then select cameras that offer a detailed view for the first and third monitor. They must also zoom in or out, pan and tilt each monitor in that order, retrieve an ES sequence to reach a goal from the current ES, before moving to the next ES. The model does not go into finer details such as choosing the right joint to move in order to go from one ES to another. In this task description, we consider the procedural knowledge required to manipulate spatial representations as a spatial skill. This is in accordance with our definition in section 1 of a spatial skill as the ability to correctly manipulate spatial representations to achieve specific goals, in contrast with spatial abilities, which are more general and can be applied in many domains (for example, mental object rotation and perspective change). Spatial abilities are not represented in this model.

Finally, textual hints and explanations were annotated to provide didactic resources to be used for procedures, concepts and goals [30]. We also added certain frequent erroneous knowledge elements to the model as a result of confusing two cameras that offer similar views, but are from different modules, for example. There are 15 erroneous general knowledge and 5 erroneous procedures. A

more detailed cognitive task analysis to identify more common erroneous knowledge is planned.

The next subsections described the tutoring services provided by RomanTutor based on the task description.

## 5.2 Evaluating Semantic and Procedural Knowledge during Problem-Solving Exercises

The first tutoring service offered in RomanTutor consists of an evaluation tool to assess semantic and procedural knowledge in problem-solving exercises. In RomanTutor, the main problem-solving activity encompasses the Canadarm2 manipulation task. The evaluation is performed during an exercise as follows: after each learner action, considered to be a primitive procedure execution, the Model Tracer tries to find goal/subgoal structures to explain the learner's current partial solution. Then, by inspecting the goal/subgoal structures, two types of procedural errors can be discovered: (1) the learner applied an erroneous primitive/complex procedure for the current goal or (2) the learner did not respect the order constraints between subgoals in a complex procedure. For example, a learner could forget to adjust a camera zoom before moving the arm. In this case, the learner forgot to achieve the subgoal of adjusting the zoom. Errors concerning semantic general knowledge, which pertain to recalling erroneous knowledge, can also be detected by examining a goal/subgoal structure, given that it reveals whether general knowledge is recalled.

Such error detection processes allow for updating a student profile that consists of probabilities to detect the acquisition of valid or erroneous knowledge procedures or general concepts. For example, if the goal/subgoal structures indicate that a learner applies a procedure to retrieve a general concept several times, the system increases its level of confidence that the learner can recall that knowledge. Conversely, when learners are likely to have recalled an erroneous general concept, the system increases the probabilities of retrieving errors for such knowledge and decreases its level of confidence that the learner has mastered valid concepts.

## 5.3 Evaluating General Knowledge with Questions

The second tutoring service permits evaluating general concepts with direct questions. These questions can first appear in the form of "fill in the blanks". For instance, to test the knowledge pertaining to "CameraCP9 GivesGlobalViewOf JEM", RomanTutor can generate the following questions: "? GivesGlobalViewOf JEM", "CameraCP9 ? JEM" and "CameraCP9 GivesGlobalViewOf ?". Concretely, RomanTutor asks "CameraCP9 GivesGlobalViewOf ?" by showing learners a view of the "JEM" module and asking them to identify the camera used (c.f., Figure 6). Other "fill in the blank" questions are also implemented in RomanTutor, such as those that ask for the name of the closest modules of a given unit, or requesting the learner to select the best suited camera to view one or more specific modules. RomanTutor also generates multiple-choice questions on general knowledge, where valid general knowledge information is selected as the right answer and different erroneous semantic knowledge units are selected or generated for the incorrect answers. After each

question, the learners' probabilities of having mastered the assessed semantic knowledge are updated.

Note that certain general knowledge evaluated with questions in RomanTutor is not handled by the procedural knowledge modeled for the problem-solving activities. Indeed, RomanTutor also offers questions in the context of a "Space Station Knowledge Builder Quiz" where learners' general knowledge regarding the ISS and Canadarm2 are tested. These questions are primarily related to the relative position of modules and cameras, since ESs are invisible to users.

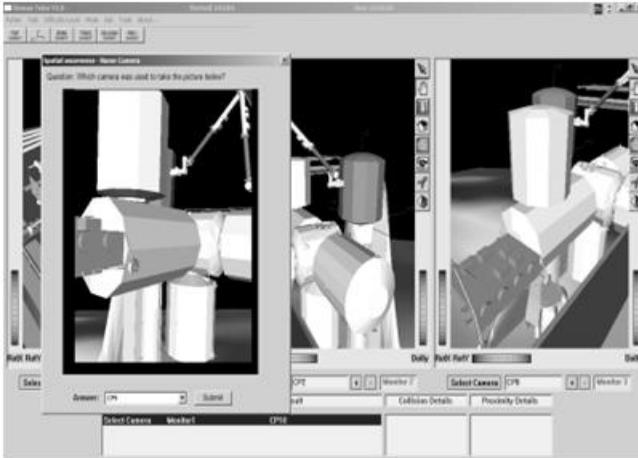


Fig.6. A camera identification exercise

#### 5.4 Providing Hints during Learning Activities and Generating Demonstrations

The third tutoring service is designed to generate demonstrations and provide customized hints during learning activities. In problem-solving activities, hints or demonstrations are offered upon learners' requests or when learners fail to respond within a predefined time limit. In that case, learners are considered to be ignorant of the correct procedure to reach the current goal, incapable of recognizing the relevant preconditions, or unable to recall certain semantic knowledge. In order to provide customized hints and demonstrations, RomanTutor analyzes the goal/sub-goal structures extracted by the model tracer from the current learner solution. From it, RomanTutor infers the appropriate correct procedures or knowledge to be recalled to achieve the goal. This permits a step-by-step demonstration that shows learners how to solve a problem, or providing hints on the subsequent correct steps. Textual hints are extracted from annotated procedures, goals, and general knowledge.

In order to generate a complete demonstration and illustrate a path leading to a goal, RomanTutor also relies on the path-planner (c.f., Figure 7). Given the initial arm configuration, the path-planner can calculate a path avoiding obstacles to reach the goal. The path generated provides an approximate possible solution.

RomanTutor also offers hints in the context of questions. For example, when learners request hints for multiple-choice questions, RomanTutor either shows a text hint that has been annotated to the valid general knowledge or removes one wrong choice if no hint is available.

### 5.5 Generating Personalized Exercises

The fourth tutoring service provides personalized exercises for learners. Once learners have performed a certain number of exercises, the system acquires a detailed profile of their strengths and weaknesses regarding their procedural and semantic knowledge (a set of probabilities). The system relies on such learner profiles to generate exercises, questions and demonstrations customized for the learner that will involve the requisite knowledge. If the system infers that a learner possesses erroneous knowledge, for example that camera "CP10" is the right camera to view the JEM module, it will likely generate direct questions about the corresponding valid knowledge or exercises to trigger its retrieval. Personalized exercises are derived from a curriculum, which consists of a set of learning objectives along with minimum mastery levels expressed as a value between 0 and 1 [14]. Different curricula are used to describe different levels of training. A learning objective consists of a performance description that learners must demonstrate once their training is completed [15]. We define a learning objective as a set of one or more goals. Such an objective is considered to be achieved when learners show that they master at least one procedure for each goal, according to the curriculum and the learners' profiles. The actual procedures used are of little importance, as several correct procedures can achieve the same goal. Learning objectives can also be stated in terms of one or more general concepts to be mastered. During a learning session, the system compares the curriculum requirements with the learner's profile estimates in order to generate exercises, questions and customized demonstrations for the learner, which involve the knowledge to be mastered [14]. In RomanTutor, a simple curriculum has been defined, where all the valid knowledge should be mastered at the same level. It will be part of our future work to evaluate and adjust this cur-

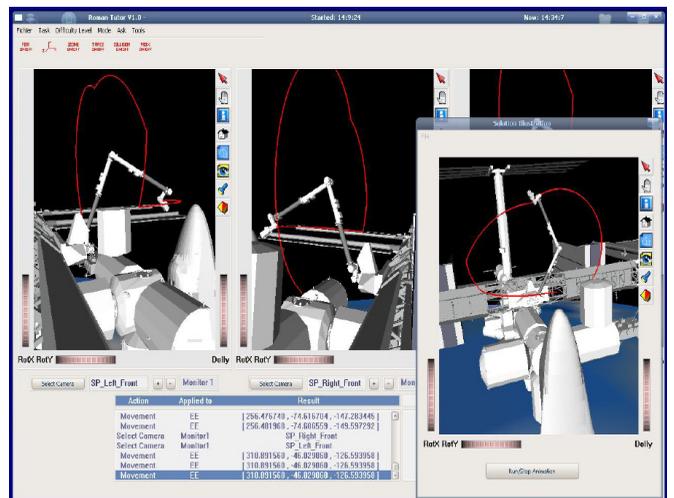


Fig.7. Path planning and task demonstration using FADPRM curriculum.

## 6 AN EVALUATION OF THE TUTORING SERVICES

An experimental evaluation was conducted to evaluate the effectiveness of the tutoring services in RomanTutor. We asked 8 students to use the new version of

RomanTutor for 30 minutes each. The same students were also asked to try the version relying solely on the path-planner. Globally, we found that the system behavior was significantly improved in the version integrating both the path-planner and the cognitive model. Students also preferred this version, because the tutoring services are more elaborated. Although the path-planner-only version provides hints during problem-solving exercises and path demonstrations, it cannot generate personalized problems and questions, and the errors that it can detect are limited.

Conversely, the cognitive model allowed detecting important procedural errors. One common type of mistake discovered was to not follow the strict security protocol. For instance, some learners forgot to adjust a camera's pan, tilt and zoom in that order. Others common procedural errors were to assign a camera giving a global view of the current ES to the first or the third monitor instead of the second one, or to move the arm without selecting and adjusting the cameras first. In all of these cases, the new version of RomanTutor intervened. As a result of this feedback, the rate of these errors decreased through learning sessions.

The evaluation of general semantic knowledge also showed it to be beneficial. After a few exercises, the system detected, for example, that learners did not know that certain cameras should be used for viewing certain ESs, although learners sometimes knew how to use the same camera to view other ESs. Also, it was observed that learners are more familiar with moving from certain ESs to certain other ESs, and that they tend to often use the same set of cameras. The new version of RomanTutor used that latter information to generate a tailored exercise that requires the learner to move to different ESs and use different cameras. This result in exercises that are more challenging and varied and that help gain a better cognitive map of the spatial environment. Learners generally appreciated the questions asked by RomanTutor. Although they found some of the questions to be difficult, most learners agreed that the questions helped them perform better in the manipulation task.

## 7 DISCUSSION AND RELATED WORK

This section first reports how the evaluation of knowledge is carried out in other tutoring systems. Then, it briefly describes limitations and complementary work in RomanTutor.

### 7.1 Evaluation of Semantic and Procedural Knowledge in Other Tutoring Systems

In tutoring systems, learners' semantic knowledge is usually evaluated through direct questions (for example, [27]). Another approach is composed of automatically scoring and comparing concepts maps drawn by learners with expert maps [20]. A concept map is basically a graph, where each node represents a concept or a concept instance, and each link specifies a relation. Although these approaches can be effective, they are best suited for non-procedural domains. They can, however, be used for

procedural domains where the procedural knowledge is taught as semantic knowledge (for example, as a text).

Assessing procedural knowledge in problem-solving activities is generally achieved by defining an effective problem space, where the correct and incorrect steps to solve a problem are specified. The most famous examples of such systems, which are based on a cognitive theory, are the Cognitive Tutors [3]. They assume that the mind can best be simulated with a symbolic production rule system [2]. In Cognitive Tutors, all actions conducted by learners are considered to be applications of rules that executes actions and may also create sub-goals. The process of comparing learners' actions (rules) with a task model to detect errors is called "Model Tracing". Recently, this concept has been embedded in a development kit called CTAT [1]. Although Cognitive Tutors are successful, they focus on teaching procedural knowledge in the context of problem-solving exercises. Anderson et al. [3] make this clear: "we have placed the emphasis on the procedural (...) because our view is that the acquisition of the declarative knowledge is relatively problem-free. (...) Declarative knowledge can be acquired by simply being told and our tutors always apply in a context where students receive such declarative instruction external to the tutors. (...) Production rules (...) are skills that are only acquired by doing."

Alternatively, it is possible to build tutoring systems that combine semantic and procedural knowledge evaluation by offering separate learning activities for learning procedural and semantic knowledge.

Unlike the aforementioned approaches, our proposal provides assessment tools for evaluating semantic knowledge not only with questions but also with procedural knowledge in problem-solving tasks. This matches educational researchers' work that emphasizes the importance of understanding how semantic and procedural knowledge are both expressed in procedural performance [36]. We claim that the view of the Cognitive Tutors is limited in two ways. First, it assumes that declarative knowledge can be taught in an explicit way effectively by human tutors. However, this is not always the case. Although one could teach spatial knowledge explicitly, handling correctly these representations during procedural tasks is best learned by doing. Second, Cognitive Tutors cannot evaluate semantic knowledge. They assume that learners acquire semantic knowledge before carrying out problem-solving exercises, that it will be available to them during the exercises or that they will recall it while performing the exercises, and that they will know when to use it. If a problem arises when learners possess erroneous semantic knowledge, when they cannot recall it or if they do not know when to use semantic knowledge, the Cognitive Tutors will incorrectly process the mistakes made by learners in terms of a missing procedure or the use of an erroneous procedure, possibly triggering inappropriate tutoring behavior. The lack of mechanisms for evaluating general semantic knowledge also means that a Cognitive Tutor would not be able to evaluate spatial representations.

One could ask, "why not simply represent the recalled

semantic knowledge as procedural knowledge?”. Although it is true that a procedure that requests semantic knowledge from semantic memory, and a procedure that uses it, can be replaced by one or many specialized procedures that perform the same function without recalling knowledge, this would require defining one or more procedures for each fact to be recalled. The advantage of our approach is that semantic knowledge is explicit. Thus, it can be recalled and used by procedures as a parameter. Hence, a task model can have fewer procedures (rules) which can be more general. In RomanTutor, this is important as it allowed us to explicitly define the spatial knowledge that can be recalled as semantic knowledge and to model its recall in a procedural task. Furthermore, the explicit representation of general knowledge through relations is advantageous as it allows for the evaluation of general knowledge in problem-solving learning activities, but also because it permits generating direct questions about it. For example, to test the semantic knowledge “CP10 canView Zone1”, the RomanTutor can generate the questions “? canView Zone1”, “CP10 canView ?” and “CP10 ? Zone1”. If such general knowledge is not explicit, answering these three questions would be viewed as using at least three different specialized procedures and a virtual tutor would not understand that they are related and that such knowledge can also be recalled in problem-solving exercises.

## 7.2 Limitations of the Model in RomanTutor and Related Works

The proposed model allows close tracking of students’ reasoning and it supports useful tutoring services in RomanTutor. It should be rather general as it is based on general cognitive theories such as ACT-R, which has provided inspiring results with over one hundred domain models, spanning backgammon games to driving simulations [47]. However, the proposed model also shows certain limitations in the Canadarm2 manipulation tasks, as it does not allow modeling finer details like how to determine the rotation values to apply to move the joints from an ES to another. The reason for this is that for the Canadarm2 manipulation tasks, it is impossible to set up a clear task model at this level of detail using a rule-based symbolic model. For this reason, RomanTutor also relies on two other techniques, characterized by complementary advantages.

The first technique refers to the aforementioned path-planner. Its advantage is that it can generate a path between any two arm configurations. However, the downside is that it generates paths which are not always realistic or easy to follow: it only applies to path-planning tasks, and it does not cover other aspects of the manipulation tasks such as selecting cameras and adjusting their parameters. In fact the path-planner is useful to provide hints and generate path demonstrations. But it does not support the generation of personalized exercises or the evaluation of a learner’s knowledge.

The second technique that we developed [31], [47] is to use knowledge discovery techniques to mine frequent action sequences and associations between these sequences in a set of recorded usage of the RomanTutor by

novices, intermediates and experts. This technique has the advantage of generating a problem-space to track learners’ actions and suggest hints at the joint rotation level based on real human interactions. However, the problem-space must be created for each exercise and the resulting problem-space remains incomplete. In fact, in certain situations, no frequent sequences are available to help learners and the assistance provided is sometimes inappropriate. We are currently developing algorithms to better integrate that latter approach with the cognitive model and the path-planner to provide more effective tutoring services in RomanTutor.

## 8 CONCLUSIONS AND FUTURE WORK

This paper highlights the importance of building simulator-based tutoring systems to acquire spatial skills and representations efficiently. The Canadarm2 manipulation task, which involves complex spatial reasoning, is described. Relevant findings from the field of spatial cognition are presented and the fact that current spatial cognition models are ill-adapted for tutoring systems is discussed. A cognitive model is then introduced, based on global theories of cognition used in previous research, and it is adapted to support the evaluation of spatial representations and skills. The extension is based on a representation of allocentric spatial knowledge as semantic knowledge. To model recalling spatial knowledge during a task, a semantic retrieval mechanism is added. Semantic retrieval allows for semantic knowledge assessment, not only with direct questions, but also indirectly by observing problem-solving tasks. We put forward tools that exploit the models and present four important tutoring services offered by RomanTutor. An initial experimental validation with RomanTutor shows promising results. As the solution cannot model all of the complexities of manipulation tasks, complementary work in RomanTutor is briefly presented. In addition, the related work section discussed evaluation of semantic and procedural knowledge in other tutoring systems, and presented limitations and complementary work in RomanTutor. Because the cognitive model is based on global cognitive theories, it should be fairly general, although it may need adaptation for particular domains.

In order to achieve more realistic simulations, new simulator functionalities are already being developed. This will permit the integration of new types of exercises such as using the arm to inspect the space station. The cognitive model and the task description will also be improved. After these improvements, we plan to perform a more comprehensive experimental evaluation of the cognitive model, which will allow measuring the learning benefits of the new tutoring services and their usefulness for real-life training situations. We also plan to take into account the challenges of working with three monitors that provide partial views of the same environment, by developing a visual perception model.

Moreover, the possibility of exploiting the spatial abilities and preferences of each individual will be investigated. Spatial abilities refer to general cognitive skills that each person displays, to a greater or lesser extent, such as

mentally rotating mental objects or imagining changes of perspective. They are generally evaluated by psychological tests [19], [21]. Considering the skill levels of individual learners, as well as their spatial abilities or preferences, would allow for further personalization. It has been shown, for example, that offering humans egocentric or allocentric instructions according to their preferences can improve their performance in navigation tasks [34]. Also, based on a questionnaire used to assess spatial abilities, Milik et al. [25] observe that it seems advantageous to adapt visual presentations of learning activities with text or images in the context of 2D exercises. Preliminary experiments by Morganti et al. [28] also show that assessing the "spatial orientation" ability with brain injury patients by observing them navigate in virtual 3D environments yields similar results as investigations designed to evaluate with paper and pencil psychology tests.

## ACKNOWLEDGMENTS

Our thanks go to the Canadian Space Agency, the Fonds Québécois de la Recherche sur la Nature et les Technologies (FQRNT) and the Natural Sciences and Engineering Research Council (NSERC) for their logistic and financial support. The authors also thank Khaled Belghith, Daniel Dubois, Usef Faghihi, Mohamed Gaha and other current and past members of the GDAC/PLANIART teams who participated in the development of RomanTutor.

## REFERENCES

- [1] M. Hegarty and D. Waller, "A dissociation between mental rotation and perspective-taking spatial abilities," *Intelligence*, vol. 32, pp. 175-191, 2004.
- [2] M. Kozhevnikov and M. Hegarty, "A dissociation between object manipulation spatial ability and spatial orientation ability," *Memory & Cognition*, vol. 29, pp. 754-755, 2001.
- [3] B. Tversky, "Cognitive Maps, Cognitive Collages, and Spatial Mental Models," *Proc. COSIT'93*, pp. 14-24, Berlin: Springer-Verlag, 1993.
- [4] J. N. Templeman and L. E. Sibert, "Immersive Simulation of Coordinated Motion in Virtual Environments: An application to Training Small Unit Military Tacti Techniques and Procedures," *Applied spatial cognition: from research to cognitive technology*, G. Allen, eds., Bahwah, NJ: Erlbaum, 2006.
- [5] E. Wenger, *Artificial Intelligence and Tutoring Systems*, Morgan Kaufmann, 1987.
- [6] F. Kabanza, R. Nkambou and K. Belghith, "Path-Planning for Autonomous Training on Robot Manipulators in Space," *Proc. Intern. Joint Conf. Artificial Intelligence*, 2005.
- [7] R. Nkambou, K. Belghith and F. Kabanza, "An Approach to Intelligent Training on a Robotic Simulator using an Innovative Path-Planner," *Proc. 8th Intern. Conf. Intelligent Tutoring Systems*, pp. 645-654, 2006.
- [8] V. Aleven, B. McLaren, J. Sewall, and K. Koedinger, "The Cognitive Tutor Authoring Tools (CTAT): Preliminary evaluation of efficiency gains," *Proc. Eighth Intern. Conf. Intelligent Tutoring Systems*, pp. 61-70, 2006.
- [9] J. R. Anderson, A. T. Corbett, K. R. Koedinger and R. Pelletier, "Cognitive Tutors: Lessons learned," *Learning Sciences*, vol. 4, no. 2, pp. 16-207, 1995.
- [10] E. Tollman, "Cognitive Maps in Rats and Men," *Psychological Review*, vol. 5, no. 4, pp. 189-208, 1948.
- [11] J. O'Keefe and L. Nadel, *The hippocampus as a cognitive map*, Oxford: Clarendon, 1978.
- [12] S. Etienne, R. Maurer and V. Séguinot, "Path integration in mammals and its interaction with visual landmarks," *Journal of experimental Biology*, vol. 199, pp. 201-209, 1996.
- [13] M. L. Mittelstaedt and H. Mittelstaedt, "Homing by path integration in a mammal," *Naturwissenschaften*, vol. 67, pp. 566-567, 1980.
- [14] T. P. McNamara, "How are the locations of objects in the environment represented in memory?," *Proc. Spatial Cognition III Conf.*, pp. 174-191, Berlin: Springer-Verlag, 2003.
- [15] D. Ekstrom and al., "Cellular networks underlying human spatial navigation," *Nature*, vol. 425, pp. 184-187, 2003.
- [16] J. S. Taube, R. U. Muller and J. B. Ranck, "Head-Direction Cells Recorded from the Postsubiculum in Freely Moving Rats. 1. Description and Quantitative Analysis," *The Journal of Neuroscience*, vol. 10, no. 2, pp. 420-435, 1990.
- [17] J. S. Taube, R. U. Muller and J. B. Ranck, "Head-Direction Cells Recorded from the Postsubiculum in Freely Moving Rats. 2. Effects of Environmental Manipulations," *The Journal of Neuroscience*, vol. 10, no. 2, pp. 436-447, 1990.
- [18] Terrazas and al., "Self-motion and the hippocampal spatial metric," *Journal of Neuroscience*, vol. 25, pp. 8085-8096, 2005.
- [19] L. McNaughton and al., "Path integration and the neural basis of the cognitive map," *Nature Reviews Neuroscience*, vol. 7, pp. 663-678, 2006.
- [20] V. D. Bohbot, G. Laria and M. Petrides, "Hippocampal function and spatial memory: Evidence from functional neuroimaging in healthy participants and performance of patients with medial temporal lobe resections," *Neuropsychology*, vol. 18, pp. 418-425, 2004.
- [21] J. D. Feigenbaum and R. G. Moris, "Allocentric Versus Egocentric Spatial Memory After Unilateral Temporal Lobectomy in Humans," *Neuropsychology*, vol. 18, no. 3, pp. 462-472, 2004.
- [22] L. Shelton and J. D. E. Gabrieli, "Neural correlates of encoding space from route and survey perspectives," *Neuropsychology*, vol. 18, pp. 442-449, 2004.
- [23] N. Burgess, "Spatial memory: how egocentric and allocentric combine," *Trends Cognitive Sciences*, vol. 10, no. 12, pp. 551-557, 2006.
- [24] L. Nadel and O. Hardt, "The Spatial Brain," *Neuropsychology*, vol. 18, no. 3, pp. 473-476, 2004.
- [25] D. Carruth and al., "Symbolic Model of Perception in Dynamic 3D Environments," *Proc. 25th Army Science Conf.*, 2006.
- [26] M. Harrison and C. D. Schunn, "ACT-R/S: Look Ma, no "cognitive map"!," *Proc. Fifth Intern. Conf. Cognitive Modeling*, pp. 129-134. Bamberg, Germany: Universitäts-Verlag Bamger, 2003.
- [27] R. M. J. Byrne and P. N. Johnson-Laird, "Spatial reasoning," *Journal of Memory and Language*, vol. 28, pp. 564-575.
- [28] G. Gunzelmann & R. D. Lyon, "Mechanisms of human spatial competence," *Proc. Spatial Cognition V Conf.* Berlin: Springer-Verlag, 2006.
- [29] D. Redish, A. N. Elga and D. S. Touretzky, "A coupled attractor model of the rodent head direction system," *Network: Computation in Neural Systems*, vol. 7, no. 4, pp. 671-685, 1996.
- [30] P. Fournier-Viger, M. Najjar, A. Mayers and R. Nkambou, "A Cognitive and Logic based Model for Building Glass-box Learning Objects," *Interdisciplinary Journal of Knowledge and Learning Objects*, vol. 2, pp. 77-94, 2006.
- [31] J. R. Anderson, *Rules of the mind*. Hillsdale, NJ: Erlbaum, 1993
- [32] Mayers, B. Lefebvre and C. Frasson, "Miace: A Human Cognitive Architecture," *Sigcuc outlook*, vol. 27, no. 2, pp. 61-77, 2001.
- [33] J. H. Neely, "Experimental dissociation and the episodic/semantic memory distinction," *Experimental Psychology: Human Learning and Memory*, vol. 6, pp. 441-466, 1989.

- [34] E. Tulving, "Episodic and semantic memory," *Organization of memory*, E. Tulving and W. Donaldson, eds., New York: Academic Press, pp. 381-403, 1972.
- [35] G. S. Halford, R. Baker, J. E. McCredden and J. D. Bain, "How many variables can humans process?," *Psychological Science*, vol. 16, no. 1, pp. 70-76, 2005.
- [36] J. R. Anderson and S. Douglass, "Tower of hanoi: Evidence for the cost of goal retrieval," *Journal of Experimental Psychology: Learning, Memory and Cognition*, vol. 27, no. 6, pp. 1331-1346, 2001.
- [37] R. M. Gagné, L. Z. Briggs and W. Wager, *Principles of Instructional Design (4th edition)*. New York: Holt, Rinehart & Winston, 1992.
- [38] R. Morales and A. S. Aguera, "Dynamic sequencing of learning objects," *Proc. Second Intern. Conf. Advanced Learning Technologies*, pp. 502-506, 2002.
- [39] E. M. Taricani and R. B. Clariana, "A technique for automatically scoring open-ended concept maps," *Educational Technology Research & Development*, vol. 54, no. 1, pp. 61-78, 2006.
- [40] J. R. Star, "On the Relationship between Knowing and Doing in Procedural Learning," *Proc. Fourth Intern. Conf. Learning Sciences*, pp. 80-86, 2000.
- [41] J. R. Anderson and al., "An integrated theory of the mind," *Psychological Review*, vol. 111, no. 4, pp. 1036-1060.
- [42] R. Nkambou, E. M. Nguifo and P. Fournier-Viger, "Using Knowledge Discovery Techniques to Support Tutoring in an Ill-Defined Domain," *Proc. Ninth Intern. Conf. Intelligent Tutoring Systems*, 2008.
- [43] P. Fournier-Viger, R. Nkambou & E. Mephu Nguifo. A Sequential Pattern Mining Algorithm for Extracting Partial Problem Spaces from Logged User Interactions. *Proc. 3rd Intern. Workshop on Intelligent Tutoring Systems in Ill-Defined Domain*, June 23-27, Montreal, Canada, pp. 46-55, 2008.
- [44] F. Pazzaglia and R. De Beni, "Strategies of processing spatial information in survey and landmark-centred individuals," *European Journal of Cognitive Psychology*, vol 13, no. 4, pp. 493-508, 2001.
- [45] N. Milik, A. Mitrovic and M. Grimley, "Fitting Spatial Ability into ITS Development," *Proc. Thirteenth Intern. Conf. Artificial Intelligence in Education*, pp. 617-619, 2007.
- [46] F. Morgranti, A. Gaggioli, L. Strambi, M. L. Rusconi and G. Riva, "A virtual reality extended neuropsychological assessment for topographi-

cal disorientation: a feasibility study," *Journal of NeuroEngineering and Rehabilitation*, vol. 4, no. 26.



spatial cognition and educational data mining.

**Philippe Fournier-Viger** received a B.Sc. (2003) and M.Sc. (2005) in Computer Science at the University of Sherbrooke and is currently pursuing a Ph.D. in Cognitive Computer Science at the University of Quebec in Montreal. He is a member of the GDAC research team and the IEEE, AACE and AIED societies. His work is funded by a NSERC Canadian Graduate Scholarship grant. His research interests include e-learning, intelligent tutoring systems, knowledge representation, cognitive modeling,



committee of the most important international conferences in Artificial Intelligence in Education.

**Roger Nkambou** (Ph.D.) is currently a Professor of Computer Science at the University of Quebec at Montreal, and Director of the GDAC (Knowledge Management Research) Laboratory (<http://gdac.dinfo.uqam.ca>). He received his Ph.D. (1996) in Computer Science from the University of Montreal. His research interests include knowledge representation, intelligent tutoring systems, intelligent software agents, ontology engineering, student modeling and affective computing. He also serves as member of the program



(<http://prospectus.usherbrooke.ca/>), a research group on data mining.

**André Mayers** (Ph. D.) is a Professor of Computer Science at the University of Sherbrooke. He founded ASTUS (<http://astus.usherbrooke.ca/>), a research group for Intelligent Tutoring Systems, which mainly focuses on knowledge representation structures that simultaneously make easier the acquisition of knowledge by students, the identification of their plans during problem solving activities, and the diagnosis of knowledge acquisition. Andre Mayers is also a member of PROSPECTUS