Multi-Paradigm Generation of Tutoring Feedback in Robotic Arm Manipulation Training

Philippe Fournier-Viger[1], Roger Nkambou[2], André Mayers[3,]
Engelbert Mephu Nguifo[4,5], Usef Faghihi[2]

[1]Dept. of Computer Sciences, University of Moncton, Canada
[2]Dept. of Computer Sciences, University of Quebec in Montreal, Canada
[3] Dept. of Computer Sciences, University of Sherbrooke, Canada
[4] Clermont Université, Université Blaise Pascal, LIMOS, F-63000 Clermont-Ferrand
[5] CNRS, UMR 6158, LIMOS, F-63173 Aubière
philippe.fv@gmail.com, nkambou.roger@uqam.ca, andre.mayers@usherbrooke.ca, mephu@isima.fr, jfaghihi@yahoo.com

Abstract. Building an intelligent tutoring system requires to define an expertise model that can support appropriate tutoring services. This is usually done by adopting one of the following paradigms: building a cognitive model, specifying constraints, integrating an expert system and using data mining algorithms to learn domain knowledge. However, for some ill-defined domains, the use of a single paradigm could lead to a weak support of the user in terms of tutoring feedback. To address, this issue, we propose to use a multi-paradigm approach. We illustrate this idea in a tutoring system for robotic arm manipulation training. To support tutoring services in this ill-defined domain, we have developed a multi-paradigm model combining: (1) a data mining approach for automatically building a task model from user solutions, (2) a cognitive model to cover well-defined parts of the task and spatial reasoning, (3) and a 3D path-planner to cover all other aspects of the task. Experimental results indicate that the multi-paradigm approach allows providing assistance to learners that is much richer than what is offered with each single paradigm.

Keywords: tutoring services, expertise model, ill-defined domains

1 Introduction

To assists learners during problem-solving activities, an intelligent tutoring system (ITS) needs to be equipped with domain knowledge that can support appropriate tutoring services. However, modelling the domain knowledge can be quite time-consuming and difficult especially for *ill-defined domains* [1]. According to Lynch et al. [1], domains containing *ill-structured problems* are ill-defined. Simon [2] defines an ill-structured problem as one that is complex, with indefinite starting points, multiple and arguable solutions, or unclear strategies for finding solutions. To provide domain knowledge to an ITS, three popular paradigms have been widely used in the ITS community. The first one is cognitive task analysis, which consists of observing expert and novice users (e.g. [3]) to produce effective problem spaces or task models. However, cognitive task analysis is very time-consuming [3]. Furthermore, for ill-defined domains, it is not always possible to define a complete or partial task model by hand. The second paradigm is constraint-based modeling (CBM) [4]. It consists of specifying sets of constraints on a correct behavior instead of providing a complete task description. Though, this approach was shown to be effective for some ill-defined domains, it can be very challenging to design a complete set of constraints for some domains. The third paradigm consists of integrating an expert system into an ITS (e.g. [5, 6]). However, developing an expert system can be difficult and costly, especially for ill-defined domains, and expert systems sometimes do not generate explanations in a form that is appropriate for learning. Recently, a fourth paradigm [7, 8] used data mining algorithms to automatically extract partial task models from users interactions with an ITS. The partial task models can then be used to offer assistance to learners. Even though the approach was proven to be efficient in procedural ill-defined domains, the task models extracted are partial and are not useful for unseen situations.

We assume that a good integration of these different paradigms could help maximize the benefits associated with each of them in specific conditions. To validate this hypothesis, we have implemented the multi-paradigm model within CanadarmTutor, an ITS for training astronauts to the Canadarm2 robot manipulation in various situations. Our preliminary experiments have shown promising results.

This paper is organized as follows. Section 2 introduces CanadarmTutor and the three paradigms we have implemented into it for representing the domain expertise. Section 3 explains how we have combined them in a multi-paradigm expert model. Section 4 presents an experimental evaluation of CanadarmTutor equipped with the multi-paradigm model, followed by some concluding remarks in section 5.

2 CanadarmTutor

CanadarmTutor [9] (cf. Figure 1.a) is a simulation-based tutoring system for coaching astronauts how to operate Canadarm2 (cf. Figure 1.b), a 7 degrees of freedom robotic arm deployed on the International Space Station (ISS). The main learning activity in CanadarmTutor is to move the arm from a given configuration to a goal configuration. Such activity is usually done in various complex tasks including inspecting the ISS and moving payloads. The arm movements are performed by astronauts inside the ISS. Maneuvering Canadarm2 on the ISS is difficult since there is a limited view of the environment. The environment is rendered through 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. To move the arm, the operator must select at every moment the best cameras for viewing the scene of operation. Moreover, an operator has to select and perform appropriate joint rotations for moving the arm, while avoiding collisions and dangerous configurations. Operators also have to follow an extensive security protocol that comprises numerous steps because a single mistake, such as neglecting to lock the arm into position can lead to catastrophic and costly consequences. Operating Canadarm2 is an ill-defined task (according to the definition of Simon [2]) because there exist a huge number of ways to move the arm to a goal configuration and it is very difficult to formalize how to select the moves that a human would execute. The reason is that some arm movements are preferable to others depending on criteria that are hard to be formalized such as the view of the arm given by the cameras, the relative position of obstacles on the ISS to the arm and the familiarity of the user with certain manipulations. In practice, skills to operate the arm are mainly learned by practice. Because of this, it is hard to model the domain expertise in CanadarmTutor.

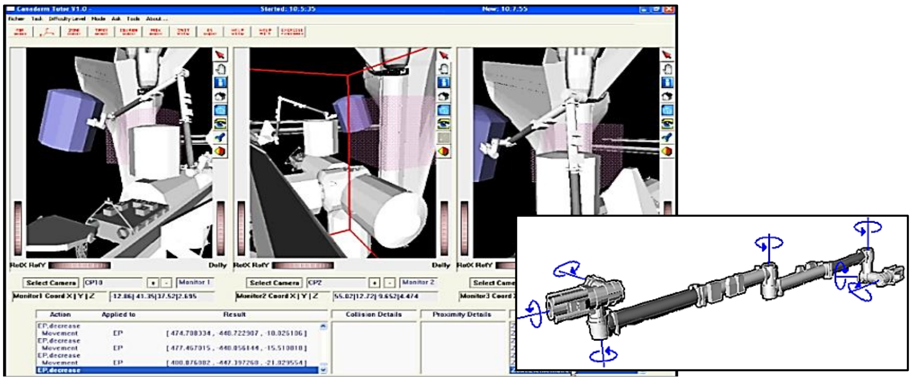

Fig. 1. (a) CanadarmTutor, (b) a 3D representation of Canadarm2

2.1 Integrating a Path-Planner for Automatic Path Generation

To implement the domain expertise in CanadarmTutor, we first based our work on the expert system approach. A custom path-planner named FADPRM was integrated into CanadarmTutor [9]. FADPRM is an efficient algorithm for robot path-planning in constrained based environments. It can calculate a trajectory (e.g. Figure 2.a) between any two robotic arm configurations while avoiding obstacles and considering constraints such as dangerous and desirable zones. Integrating FADPRM in CanadarmTutor provides the following benefits. First, in a training session, CanadarmTutor uses FADPRM to automatically produce demonstrations of correct arm maneuver on the ISS by generating a path between two arm configurations, while considering the obstacles (the ISS modules) and predefined constraints. Second, for a given task, CanadarmTutor automatically generates paths and estimates the distance with the learner solution to evaluate it. Although the path-planner can provide useful tutoring services, our experiments with learners show that the generated paths are not always realistic, as they are not based on human experience. Moreover, they do not cover some important aspects of the task such as selecting cameras and adjusting their parameters. Furthermore, given that the path-planner has no representation of knowledge and skills, it cannot support important tutoring services such as estimating learners' knowledge gaps.

2.2 Integrating a Cognitive Model to Assess Skills and Spatial Reasoning

Facing these problems, we applied the cognitive task analysis paradigm [3]. To understand how astronauts operate Canadarm2, we attended two-week training with astronauts at the Canadian Space Agency and also interviewed the training staff. To encode how users operates the robotic arm, we used a custom cognitive model [10], similar to the one used in CTAT [1], the reference model for building "model-tracing tutors". The main difference between CTAT and our model is that ours is designed to also evaluate spatial reasoning, a key issue for manipulating Canadarm2. To take into account the spatial dimension, our review of the literature on spatial cognition has shown that most researchers in psychology and neurosciences agree that spatial knowledge is declarative and is necessary for complex spatial reasoning ("allocentric representations") [11, 12, 13]. Furthermore, spatial knowledge could be represented by relations of the form "a r b", where "a" and "b" are symbols designating objects and "r", a spatial relationship between the objects [14].

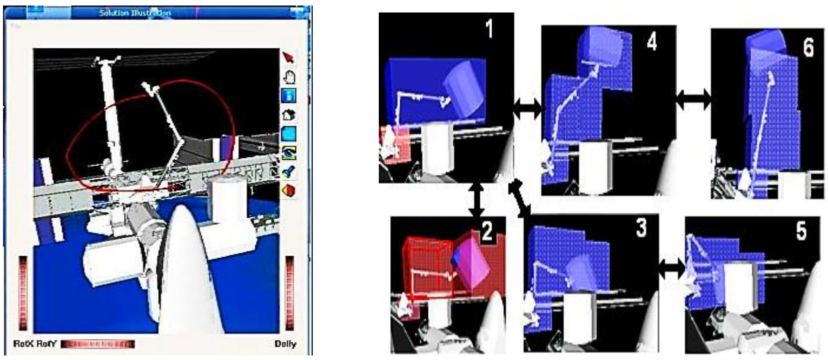

Fig. 2. (a) The FADPRM Path-Planner (b) Six Elementary Spaces

Based on these facts, to model the spatial knowledge in CanadarmTutor, we discretized the 3D space into 3D subspaces that we name elementary spaces (ESP). This allows us to represent the continuous space as discrete symbols. In Canadarm2 manipulation, it was determined that the most realistic types of ESP for mental processing are ESs configured with an arm shape. Figure 2b illustrates 6 of the 30 ESs that we defined. For example, one can move the arm from ESP 1 to ESP 2, ESP 3 and ESP 4. ESP 5 can be reached from ESP 3, and ES6 can be reached from ES4. Each ESP is represented by three cubes. Spatial knowledge was then encoded as four types of relationships such as (1) a camera can see an ESP or an ISS module, (2) an ESP contains an ISS module, (3) an ESP is next to another ESP and (4) a camera is attached to an ISS module. The procedural knowledge of how to move the arm to a goal configuration was modeled as a loop where the learner, before any arm movements, must recall a set of cameras for viewing the ESPs containing the arm, select the correct cameras, adjust their parameters, retrieve a sequence of ESPs to go from the current ESP to the goal, and then start moving the arm to the next ESP.

This task model allowed us to integrate six new tutoring services in CanadarmTutor. First, a learner can explore the task model to learn how to operate the arm and learn about properties of the ISS, the cameras and Canadarm2. Second, model-tracing capability allows the system to evaluate the learner knowledge during arm manipulation exercises. After a few exercises CanadarmTutor automatically builds a detailed learner profile that shows the strength and weakness of the learner in terms of mastered, missing and buggy knowledge. This is done by comparing the task model with a learner solution to see which knowledge is used by the learner. Third, CanadarmTutor uses the declarative knowledge linked to the task model to generate and provide the learner with direct questions such as "Which camera can be used to view the Node02 ISS module?". The fourth tutoring service is to assist the learners by providing useful hints and demonstrations during arm manipulation exercises. Suggesting the next step and generating demonstrations is done thanks to the model-tracing capability of this paradigm. The fifth tutoring service is to generate personalized exercises based on the student model. By using the student model, CanadarmTutor can generate exercises that involve knowledge not yet mastered by the learner. The sixth and last tutoring service is to offer proactive help to the learner. For instance, if Canadarm2 is moved without performing camera adjustment, CanadarmTutor warns the learner to check if cameras are well adjusted. This type of help which is also implemented based on model-tracing is particularly appreciated by beginners and intermediate learners. However, the cognitive model also has some limitations. Although it models the main steps of the manipulation task in detail, it does not go into details about how to select joint rotations for moving Canadarm2. The reason is that for a given arm movement problem, there is a huge number of possibilities and choosing one of them requires considering criteria that are hard to formalize such as the safety and ease of manoeuvres. It is thus not possible to define a complete and explicit task model for this task, making it an ill-defined task according to Simon's definition [2]. The path-planner could generate paths to provide help at the level of joint rotation. But they are sometimes too complex and difficult to be executed by users, as they are not based on human solutions.

2.3 Using Data Mining Techniques to Learn Partial Task Models

Given the aforesaid drawbacks with other paradigms, we applied the fourth paradigm, which is the automatic acquisition of partial task models [8]. It consists of applying data mining algorithms on user solutions to automatically extract a partial task model instead of defining it by hand. The goal is to provide tutoring services for parts of the task of operating the arm that are ill-defined and could not be represented easily with the cognitive model (e.g. how to select the joint rotations to move Canadarm2). An advantage of this approach over the path-planner is that it is based on real user data.

To apply this approach, we first recorded a set of user solutions for each exercise [8]. In CanadarmTutor, an exercise consists of moving the robotic arm from an initial configuration to a goal configuration. For each attempt, a *sequence of actions* is created in a database. We defined 112 actions that can be recorded including (1) applying a rotation value to one of the seven arm joints (2) selecting a camera and (3) performing an increase or decrease of the pan/tilt/zoom of a camera. An example of a partial action sequence recorded for a user in CanadarmTutor is <(0, *rotateSP*{2}), (1, *selectCP3*), (2, *panCP2*{4}), (3, *zoomCP2*{2})> which represents decreasing the rotation value of joint SP by two units, selecting camera CP3, increasing the pan of camera CP2 by four units and then its zoom by two units. Furthermore, we annotated sequences with contextual information called "dimensions". Table 1 shows an example of a toy database containing six solutions annotated with five dimensions. In this Table, *a*, *b*, *c*, and *d* denote actions. The dimension "*Solution state*" indicates if the learner solution was successful. Values for this dimension are assigned by CanadarmTutor. The four other dimensions are examples of dimensions that can be added manually. The dimension “Expertise” denotes the expertise level of the learner who performed a sequence. “Skill_1”, “Skill_2” and “Skill_3” indicate wether any of these three specific skills were demonstrated by the learner when solving the problem. This example illustrates five dimensions. However, any kind of learner information or contextual information can be encoded as dimensions. In CanadarmTutor, we used 10 skills that we selected to be the most important, and the “solution state” and “expertise level” dimensions to annotate sequences.

To generate a partial task model from the user solutions, we then applied a custom sequential pattern mining algorithm [8] on the database of user solutions. The algorithm takes as input a sequential database and a threshold named *minsup*. The algorithm then extracts subsequences of actions that are common to at least *minsup* learners. We have designed the custom algorithm specifically to accept dimensions and also different types of constraints useful in our context [8]. Table 2 shows some subsequences (also called patterns) found from the database shown in Table 1 with *minsup* = 2. Consider pattern P3. This pattern represents doing action *b* one time unit (immediately) after action *a*. The pattern P3 appears in sequences S1 and S3 of Table 1. It has thus a *support* of two. Moreover, the annotations for P3 tell us that this pattern was performed by experts who possess skills 1, 2 and 3 and that P3 was found in plan(s) that failed, as well as plan(s) that succeeded.

Table 1. An example toy database containing 6 user solutions

ID	Dimensions					Sequence of actions
	Solution state	**Expertise**	**Skill_1**	**Skill_2**	**Skill_3**	
S1	successful	Expert	yes	yes	yes	<(0,a),(1,bc)>
S2	successful	novice	no	yes	no	<(0,d) >
S3	buggy	expert	yes	yes	yes	<(0,a),(1,bc)>
S4	buggy	intermediate	no	yes	yes	<(0,a),(1,c), (2,d)>
S5	successful	expert	no	no	yes	<(0,d), (1,c)>
S6	successful	novice	no	no	yes	<(0,c), (1,d)

Table 2. Some frequent patterns extracted from the dataset of Table 1 with a *minsup* of 2

ID	Dimensions					Sequence of actions	Support
	Solution State	**Expertise**	**Skill_1**	**Skill_2**	**Skill_3**		
P1	*	expert	yes	yes	yes	<(0,a)>	2
P2	*	*	*	yes	yes	<(0,a)>	3
P3	*	expert	yes	yes	yes	<(0,a), (1,b)>	2
P4	successful	*	no	*	*	<(0,d)>	3

We have then implemented three tutoring services in CanadarmTutor that use the partial task models. First, CanadarmTutor can assess the profile of the learner (expertise level, skills, etc.) by looking at the applied patterns. If for example a learner applies patterns with the value "intermediate" for the dimension “expertise” 80 % of the time, then CanadarmTutor asserts that the learner expertise level is "intermediate". In the same way, CanadarmTutor can diagnose mastered and missing/buggy skills for users who demonstrated a pattern by looking at the “skills” dimensions of the applied patterns (e.g. “Skill_1” in Table 2).

The second tutoring service consists in determining the possible actions from the set of patterns and proposing one or more actions to the learner. In CanadarmTutor, this functionality is triggered when the student select "What should I do next?" in the interface menu. CanadarmTutor then checks the matching patterns to make a recommendation to the learner. For example, if the learner performed a rotation of the joint SP followed by a rotation of the joint EP and ask “What Should I do next?”, CanadarmTutor will look for patterns that match with SP, EP to suggest what next action the learner should do.

The third tutoring service is to let learners explore patterns by themselves to find out about ways to solve problems. CanadarmTutor provides an interface that lists the patterns and their annotations, and provides sorting and filtering functions.

The paradigm of learning partial task models from user solutions has several advantages. Unlike the path-planner, it allows us to provide tutoring services based on real users’ arm manipulations (multiple profile users). Moreover, it allows us to assist learners about how to choose a joint rotation –which was impossible to achieve with the cognitive model. However, an important limitation with the partial task model paradigm is that no help can be offered to learners for unexplored solution paths. Thus each of the three paradigms that we have separately tested into CanadarmTutor has its own advantages and limitations. Based on this observation, we decided to combine them to create a multi-paradigm expertise model.

3 Combining the Three Paradigms

The goal is to provide a model that can switch from one paradigm to another in order to take advantages of each one’s strength in situations where it is the best. The proposed multi-paradigm model works as follows.

During arm manipulation exercises, CanadarmTutor performs model-tracing to update the student model. The student model is a list of knowledge units from the cognitive model. Each unit is annotated with a probability that indicates if the knowledge is mastered by the learner. Moreover, the student model is also updated when a learner answers questions asked by CanadarmTutor (cf. section 2.2).

When an exercise is completed (fail or success), the solution is added to a sequence database of user solutions for that exercise (a database similar to the one shown in Table 1). The solution is then annotated with the dimension “Solution State” to indicate the success or failure. Moreover, the skills from the cognitive model are used to annotate sequences as dimensions (if the mastery level is higher than 0.8 in the student model, the skill is considered mastered). Thereafter, when a minimum of 10 sequences have been recorded for an exercise, the data mining algorithm is applied for extracting a partial task model for the exercise.

When CanadarmTutor detects that a learner follows a pattern during an exercise from the corresponding partial task model, dimensions of the pattern are used for updating the student model. For example, if a learner applies a pattern common to learners possessing “Skill_1”, the mastery level of “Skill_1” in the student model will be heightened by a small increment (we use 0.05 in CanadarmTutor). In this way, the partial task models are also used for updating the student model (the student model is shared by the cognitive model and the partial task model approach).

During a learning session, CanadarmTutor uses the student model for generating exercises that progressively involves new knowledge or knowledge that is judged not yet mastered by the learner (this is done as explained in section 2.2). The exercises that are generated are either questions about declarative knowledge of the cognitive model or robotic arm manipulation exercises.

During an arm manipulation exercise, when a learner asks for help about what should be done next, the system generates a solution using the three aforementioned approaches (cf. Figure 3). First, the cognitive model gives the general procedure that should be followed for moving the arm such as “You should select a camera and then adjusts its parameter for monitor 2” (cf. Figure 3.A). This help is generated by performing model-tracing with the cognitive model. Then, in the same window, the patterns from the partial task model that match the current user solution are displayed to the learner. For example, three patterns are presented in Figure 3.B. The learner can view a pattern as an animation by using the arrow buttons. Patterns give mainly the information about the joint rotations that should be performed for moving the arm. If no pattern matches the current learner solution, a demonstration is generated by the path-planner that demonstrates possible paths as solutions (cf. Figure 3.C).

Furthermore, CanadarmTutor can provide proactive help to learners such as assisting the learners to choose the best cameras thanks to the cognitive model (cf. section 2.2). CanadarmTutor can also let the learner explore patterns from the partial task models (cf. section 2.3) or the cognitive model (cf. section 2.1) to learn about different ways to solve problems or about the general procedure for moving the arm. The learner can also request demonstrations at any time from the path-planner (cf. section 2.1) or the cognitive model (cf. section 2.2).

Table 1 summarizes the different tutoring services supported by each paradigm and the multi-paradigm model is provided in Figure 3. It shows that the tutoring services supported by the multi-paradigm approach are much richer.

4 Experimental Evaluation

We performed an evaluation with ten users to evaluate the multi-paradigm version of CanadarmTutor. The goal of the evaluation was twofold: (1) to measure if the tutoring services help the learners to learn and (2) if, during an exercise, CanadarmTutor’s interventions are relevant to the current solution. To make sure that for each exercise some patterns are extracted by our data mining algorithms, we recorded at least 30 solutions for each robotic arm manipulation exercise.

Experimental procedure. We explained to each participant the procedure of the experiment and what kind of data will be collected. Then, we asked each participant to perform fifteen procedural exercises. Completing the exercises took about one hour for each participant. During this session, we allowed participants to use all tutoring services. We set CanadarmTutor to record all solutions so that they can be examined after the experiment. During the experiment, we observed the participant and took notes to evaluate (1) if the tutoring services gave relevant help when they were used and (2) whether the learners corrected their mistakes after using the tutoring services or they were more confused. Finally, we performed a five minute interview with each learner to see their opinion on the same two aspects, and also their general opinion about the tutoring services and how CanadarmTutor could be improved.

Experimental results. All participants completed the fifteen exercises. Most participants used all tutoring services. We found that participants relied more on the tutoring services for the most difficult exercises, which is what we expected. All participants mentioned that they found the tutoring services very useful and that the tutoring services helped them learn how to manipulate Canadarm2. Our observation was that learners using the tutoring services did not repeat their mistakes after receiving feed-back. Users also agreed that the set of tutoring services would be less interesting if some were removed, which confirm that the multi-paradigm model is superior to using each individual approach.

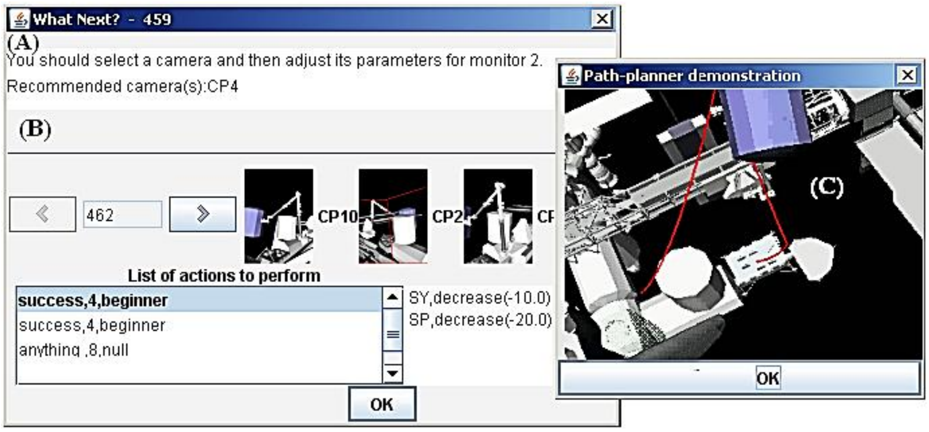

Fig. 3. A Hint Offered by the Multi-Paradigm Approach

Table 1. Tutoring services offered with each paradigm

	Path-planner	Cognitive model	Data mining approach	Multi-paradigm
Generate path demonstrations and evaluate the path followed by the learner	Yes			Yes
Free exploration of the knowledge, demonstrations, hints, proactive help, skill evaluation (for well-defined parts of the task)		Yes		Yes
Evaluate declarative knowledge with questions (including spatial knowledge)		Yes		Yes
Free exploration of the knowledge, hints, skill evaluation (for ill-defined parts of the task)			Yes	Yes
Integrated help covering all aspects of the task				Yes

5 Conclusion

In this paper, we have argued for the use of multi-paradigm approaches for supporting tutoring services in procedural and ill-defined domains. The motivation is that different approaches are sometimes better suited for different parts of the same ill-defined task. We have presented this idea using CanadarmTutor. We have first described how we have tested three different approaches to support tutoring services in CanadarmTutor. We then discussed their respective limitations and explained how the multi-paradigm approach combines the three approaches in the latest version of CanadarmTutor to overcome limitations of each paradigm. The result is tutoring services that greatly exceed what all previous versions of CanadarmTutor offered. An experimental evaluation confirmed that the multi-paradigm model allows us to provide relevant and helpful tutoring services that are appreciated by users.

References

[1] Lynch, C., Ashley, K. Aleven, V. and Pinkwart, N.: Defining Ill-Defined Domains; A literature survey. In: Proc. Ill-Defined Domains Workshop, pp. 1-10 (2006)

[2] Simon, H. A.: Information-processing theory of human problem solving. In: W. K. Estes (Ed.), Handbook of learning and cognitive processes: Vol. 5. Human information, 271-295, John Wiley & Sons, Inc. (1978)

[3] Aleven, V., McLaren, B. M., Sewall, J. and Koedinger, K.: The Cognitive Tutor Authoring Tools: Preliminary evaluation of efficiency gains. In: Proc. ITS 2006, pp. 61-70 (2006)

[4] Mitrovic, A., Mayo, M. , Suraweera, P. and Martin, B. Contraint-based tutors: a success story. In: Proc. IEA AIE 2001, pp. 931-940 (2001)

[5] Clancey, W.: Use of MYCIN's rules for tutoring. In: Buchanan, B., Shortliffe, E. H. (Eds.), Rule-Based Expert Systems., Addison-Wesley (1984)

[6] Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D. and Person, N.: Using Latent Semantic Analysis to evaluate the contributions of students in AutoTutor, Interactive Learning Environments, 8, 149-169 (2000)

[7] Barnes, T. and Stamper, J.: Toward Automatic Hint Generation for Logic Proof Tutoring Using Historical Student Data, In: Proc. ITS 2008, pp. 373-382 (2008)

[8] Fournier-Viger, P., Nkambou, R. and Mephu Nguifo, E. Learning Procedural Knowledge from User Solutions To Ill-Defined Tasks in a Simulated Robotic Manipulator. In: Romero et al. (Eds.). Handbook of Educational Data Mining, CRC Press, 451-465 (2010)

[9] Belghith, K., Kabanza, F., Hartman, L. and Nkambou, R.: Anytime Dynamic Path-planning with Flexible Probabilistic Roadmaps. In: Proc. ICRA 2006, pp. 2372-2377 (2006)

[10] Fournier-Viger, P., Nkambou, R. and Mayers, A. Evaluating Spatial Representations and Skills in a Simulator-Based Tutoring System. IEEE Trans. Learn. Tech., 1(1), 63-74 (2008)

[11] Burgess, N.: Spatial memory: how egocentric and allocentric combine, Trends in Cognitive Sciences, 10(12), 551-557 (2006)

[12] Nadel, L. and Hardt, O.: The Spatial Brain, Neuropsychology, 18(3), 473-476 (2004)

[13] Tversky, B., Cognitive Maps, Cognitive Collages, and Spatial Mental Models, In: Proc. Intern. Conference COSIT'93, pp. 14-24 (1993).

[14] Gunzelmann, G. and Lyon, R. D., Mechanisms of human spatial competence, In: Proc. Spatial Cognition V Conf. , pp. 288-307 (2006)

Acknowledgments. Our thanks go to the FQRNT and NSERC for their logistic and financial support. We also thanks all members of the GDAC/PLANIART involved in this project.