Pedagogy That Makes A Difference: Exploring Domain-Independent Principles across Instructional Management Research within the ITS Community

Proceedings of Workshop held during ITS-2014

The 12th Annual Conference on Intelligent Tutoring Systems,

Honolulu, HI

June 6, 2014

Workshop Co-Chairs:

Benjamin Goldberg & Robert Sottilare
Army Research Laboratory, Human Research & Engineering Directorate, Orlando, FL, USA

adfa, p. 1, 2011.

Preface

The purpose of this workshop is to examine current research within the ITS community focused on instructional management within educational technology, and to conceptualize its application in a domain-independent authoring environment, such as the Generalized Intelligent Framework for Tutoring (GIFT). The goal of this topic is to recognize the various factors associated with instructional management in ITSs, the types of strategies being applied in today's use cases, and current trends being researched by the field. This topic is of particular interest to the open-community of researchers currently involved in the development of GIFT, which provides a standardized environment to author and deliver adaptive functions in computer-based learning environments. For GIFT to be fully embraced by the ITS community, the architecture must be flexible enough to accommodate varying pedagogical strategies deemed useful by the field. This includes both individualized and collaborative team-based instruction. With that said, this is a critical time in GIFT's development, as standards and processes are still being defined. As such, this workshop provides a forum for the ITS community to influence future development of GIFT by defining functions and processes they would like to see supported.

This workshop closely aligns with the theme of ITS 2014, "Creating Fertile Soil for Learning Interactions." With the focus of the event on the research and application of instructional strategies across today's ITSs, the findings will help guide future development of GIFT modules and standards used to manage and regulate instructional practices. A key advantage of a generalized approach to ITS development (and GIFT in particular) is defined standards that warrant high potential for reuse across educational and training domains, thus creating the fertile soil necessary to produce effective learning experiences in a stream-lined and cost-effective manner.

The workshop is divided into three themes: (1) instructional management on a cognitive level; (2) instructional management on an affective level; and (3) instructional management in the context of team-based instruction. The themes include papers that address relevant pedagogical practices that are found to impact the effectiveness of ITS applications.

Program Committee

Co-Chair: Benjamin Goldberg, Army Research Laboratory, Orlando, FL, USA (benjamin.s.goldberg.civ@mail.mil)

Co-Chair: Robert Sottilare, Army Research Laboratory, Orlando, FL, USA (robert.sottilare.civ@mail.mil)

H. Chad Lane, *University of Southern California, Institute for Creative Technologies*

Randall Spain, *Army Research Institute*

Heather Holden, *Army Research Laboratory*

Keith Brawner, *Army Research Laboratory*

Tiffany Poeppelman, *Aptima*

Table of Contents

1. What Makes an Effective Pedagogical Model?
Benjamin Goldberg

10. Summarization during Tutoring Implications for Developing Micro-Adaptive Tutoring Systems
Sandra Katz, Patricia Albacete, and Pamela Jordan

18. Instructional Strategies in Diagram-based ITSs: Lessons Learned from Two Tutoring Systems
Enruo Guo and Stephen Gilbert

26. Pedagogy for Computer-Based Tutoring Systems Based on Experts Constraints in a Turn-Based Game
Robert Pokorny and Benjamin Goldberg

34. How Can the Affective Model be for Tutors on Moodle?
Júlia Marques Carvalho da Silva

41. The Unique Elements to Consider In Team Tutoring
Heather Holden, C. Shawn Burke, Christopher W. Wiese

49. A System Dynamics Approach to Building Team Trust Models: Exploring the Challenges
Mostafa Amin-Naseri and Stephen Gilbert

What Makes an Effective Pedagogical Model?

Benjamin Goldberg[1]

[1]United States Army Research Laboratory
Human Research & Engineering Directorate
Simulation & Training Technology Center, Orlando, FL 32826
{benjamin.s.goldberg.civ}@mail.mil

Abstract. In this paper we will provide a perspective overview on what makes an effective pedagogical model. The overview will be grounded in work being conducted by the U.S. Army Research Laboratory and their current GIFT (Generalized Intelligent Framework for Tutoring) architecture under development. An objective of GIFT is to capture best practices and lessons learned from instructional strategy research. These best practices will be encapsulated within GIFT's pedagogical module and will be used to inform authoring decisions during system development. This paper will be used to present the problem space and to identify variables that dictate instructional management decisions.

Keywords: Instructional Management, Strategy Recommendation, Pedagogical Modeling, Adaptation, Personalization

1 Introduction

Pedagogical modeling is associated with the application of learning theory and is based on variables empirically proven to influence outcomes [1]. It involves determining what to teach and how to teach it given a learner and desired end goal-state. According to Beal and Lee [2] the role of a pedagogical model is to balance the level of guidance and challenge during a learning event so as to influence performance and interaction, while maintaining student engagement and motivation. Conceptually speaking, a system should be designed to challenge a given learner just beyond their ability with guidance functions built within to support experienced impasses and misconceptions. Yet what does this mean from an operational standpoint? A pedagogical model, in a run-time instantiation, makes informed instructional decisions based on data known about the learner and functions supported by the learning environment.

The goals associated with recognizing components and processes that make up an effective pedagogical model are influenced by current work involving the GIFT (Generalized Intelligent Framework for Tutoring). GIFT is being developed as an open-source domain-independent architecture that provides the tools and methods to author, deliver, and assess adaptive instructional materials [3]. Another major goal associated with the architecture is extending its application outside of a laboratory setting and into the hands of educators and trainers. The intent is to enable course developers to apply intelligent tutoring practices into existing materials through

adfa, p. 1, 2011.

standardized tools and messaging protocols. This user base is expected to be well-versed in the knowledge and skills associated with a given domain, but lack many of the technical abilities necessary to develop a mature adaptive instructional program. With that said, GIFT requires baseline functionalities that are intended to guide authoring processes, both from a programming standpoint as well as from an instructional design standpoint. From this perspective, an ongoing effort linked with GIFT development is encapsulating best-practices into an over-arching domain-independent pedagogical model that will guide and influence authoring of run-time decisions. This model is called the engine for Macro-/Micro- Adaptive Pedagogy (eM2AP) [3].

The design and implementation of the eM2AP is still in early stages, and we are seeking critiques on proposed approaches and guidance on missing functionality. With respect to this workshop, we are presenting a conceptual framework of the various components an effective pedagogical model would manage. These include: (1) managing course flow; (2) managing knowledge assessment; (3) managing focused remediation; (4) managing practice opportunities; (5) managing guidance; (6) managing challenge; and (7) managing learner affect and motivation.

2 Adaptive Components of the engine for Macro-/Micro-Adaptive Pedagogy (eM^{2}AP)

From the traditional sense, a pedagogical model serves two primary functions. It manages both an outer- and inner-loop of instructional strategy selection, with each serving a distinctly different purpose during system run-time. In the case of outer-loop pedagogy (i.e., macro-adaptation), adaptive logic is applied to determine what is to be experienced by a learner and the sequence of its interaction. Depending on how an individual is performing and feeling during a lesson, the outer-loop will adjust the flow of instruction and practice based on metrics associated with competence and affect. In terms of inner-loop pedagogical functions (i.e., micro-adaptation), strategy selection is focused on real-time feedback intended to influence performance behaviors and scenario/problem adaptations to maintain appropriate challenge levels. From this perspective, a pedagogical model is designed to facilitate Vygotsky's Zone of Proximal Development (ZPD) [4]. These distinctions in pedagogical practice are managed by the components of instruction listed above. In this section, we define each component and how the eM^{2}AP is being built to support their processes.

The eM2AP is GIFT's first domain-independent contribution to the pedagogical module. Its design is based on sound instructional design principles and adaptive strategy selection logic informed by available empirical evidence in the literature. The overarching goal of the eM2AP is to provide a pedagogical framework that enables a course developer to easily establish adaptive course flow and guidance by enacting available techniques supported in the GIFT architecture. This multi-representation of what makes an effective pedagogical model serves as a guiding function in determining architectural design approaches to support each component. To this effect, the eM2AP is being built in multiple phases, with each providing a new functionality that improves the overall adaptive capability of GIFT.

2.1 Managing Course Flow

An effective pedagogical model is designed to provide personalized learning experiences across an array of course materials. In the instance of a complete distributed learning experience, a pedagogical model is responsible for presenting relevant information about a domain, assessing an individual's understanding of the concepts and relationships inherent to that domain, and structures and guides deliberate practice opportunities for skill development. One important outer-loop function is managing a learner's flow of interaction across these instructional events with content and guidance based on their current ability levels.

One well-received instructional technique that can guide outer-loop course flow management in an Intelligent Tutoring System (ITS) environment is mastery learning [5]. Mastery learning, from the ITS perspective, is based on a set of instructional procedures to assist a learner in mastering a set of course objectives. In most cases, this involves establishing an ontological representation of concepts for a domain. This representation highlights concept dependencies and prerequisites that ultimately guide content delivery. The notion is that a learner must show comprehension of a prerequisite concept before continuing on.

In the instance of GIFT's course management, a challenge is creating a pedagogical framework that supports the mastery learning technique. In reviewing the work surrounding instructional system design theory, David Merrill's work on the Component Display Theory (CDT) was recognized as a potential guiding benchmark for establishing a domain-independent structure of course elements [3, 6]. The CDT distinguishes course materials across a 2x2 matrix of varying content and presentation modalities. This simplified structure can assist in deconstructing a domain into its constituent parts that ultimately can be used to inform adaptive course flow across a set of learners. In essence, CDT describes a lesson structure around four fundamental piece parts. These include: (1) presenting facts and rules about a domain (rules quadrant), (2) presenting structured examples of the application of those facts and rules in a real-world context (examples quadrant), (3) assessing a learner's understanding of the facts and rules (recall quadrant), and (4) allowing a learner to apply those facts and rules in a practice environment for the purpose of skill development (Practice Quadrant; see Figure 1 for a visual representation of CDT in a GIFT lesson).

The CDT is currently being coded into the eM^{2}AP as a framework to build course interaction around. The four quadrants of Rules, Examples, Recall, and Practice establish a generalized course flow that is designed to operate on authored metadata descriptors. The metadata is used to define what quadrant specific instructional materials associate with, along with variables (i.e., difficulty ratings and individual differences) that will influence the type of learner those materials will be targeted towards. This provides the ability for an author to personalize learning experiences by designating certain materials and interactions for a particular type of individual, while maintaining a structure appropriate for mastery learning. This structure also influences how the other components of instruction identified above will be represented in the eM^{2}AP.

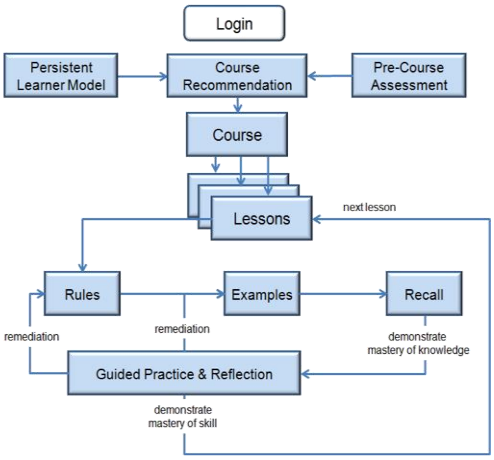


Fig. 1. Component Display Theory Represented in GIFT Lesson Flow

2.2 Managing Knowledge Assessment

An effective pedagogical model is designed to manage robust assessment practices so as to determine competency across a set of concepts linked to a domain. This approach to assessment is aimed at evaluating a learner's understanding of a domain as it relates to the dimensions of knowledge highlighted in Bloom's Taxonomy [7]. We attribute the management of knowledge assessment to outer-loop pedagogy, where a system will generate a set of assessment materials that will provide a granular snapshot of an individual's comprehension for a set of interrelated concepts. It is the pedagogical models job to determine the form of assessment along with the criteria for scoring responses. This management is important as it provides valuable information used to identify a concept that requires remediation, and the extent of understanding an individual has that will dictate the type of remediation to deliver.

Managing knowledge assessment in the eM^{2}AP is dependent on a linkage between the pedagogical module and GIFT's Survey Authoring System (SAS). The SAS is web-based authoring tool that allows a user to create surveys out of a bank of questions, and allows that user to define the context for which a survey will be used. With respect to managing knowledge assessment, both the SAS and the eM^{2}AP are being modified to support autonomous assessment creation. This capability enables a user to create a bank of domain relevant questions in the SAS, link each question to a concept that is being trained by GIFT along with an associated difficulty rating, and establishes logic for selecting questions and scoring responses. An example is when learner A enters the recall quadrant of the CDT for concepts X and Y, the eM^{2}AP requests a specified amount of questions for each concept and for each difficulty rating. Responses to a set of questions with varying ranges of complexity is used to gauge a learner's knowledge level for that set of concepts and guides remediation practices described below. Knowledge assessment can also be administered at other times during a learning event, and can serve different approaches to gauging comprehension, such as asking a learner to summarize a concept or to reflect on the interactions of a previous event.

2.3 Manage Practice Opportunities

An effective pedagogical model is designed to manage practice opportunities that optimize time on task and promote skill development through interaction. In comparing knowledge assessment to practice, the main distinction from our perspective is knowledge assessment deals with what we term as 'checks on learning'. This form of assessment is provided at designated times to check on an individual's comprehension and to assess cognitive understanding for conducting procedures and solving problems. It is the management of practice opportunities that allows a learner to apply the knowledge and skills covered in a lesson across novel problems and contexts. This management is an outer-loop function in that it determines the type of practice opportunity to deliver and directs subsequent problems and scenarios based on a learner's state information.

Practice opportunities in the eM^{2}AP are being designed around common use case examples, and map to the practice quadrant of the CDT. In the aim of providing adaptive training, there are two notional examples of interaction. The first involves blocked practice (i.e., drill and kill) that incorporates a large set of scenarios/problems of varying difficulty, where the eM^{2}AP manages problem selection based on gauged learner competency reported by the learner model. The goal is to adjust complexity of the practice in an outer-loop capacity by dictating what is presented to the learner next and designating when practice is complete. This conclusion is based on when a concept is recognized as needing remediation or a learner has shown mastery levels of performance. The other instance of practice involves a single scenario, most commonly involving game-based applications, that requires a learner to perform a set of task-related procedures that are modeled within a virtual environment. While the selection of a scenario is limited to a single event, many of these game-based environments enable scenario configurations that manipulate the difficulty and complexity of a problem by adjusting the presence and behaviors of available entities.

2.4 Manage Remediation Practices

An effective pedagogical model manages remediation practices by assisting learners in correcting misconceptions and bypassing impasses experienced during a learning event. An effective ITS indentifies persistent misconceptions across a domain, diagnoses a cause for each misconception and implements remedial methods to repair it [8, 9]. This either involves an outer-loop instance of remediation where a pedagogical model will manage the delivery of supplementary content intended to assist a learner in better understanding a topic, or it is an inner-loop instance of remediation in the form of a hint or instructional prompt aimed at providing instant remedial content. For the purpose of the paper, we designate the management of remediation practices to be associated with outer-loop adaptation, while the inner-loop remediation support will be described in the next section on managing tutorial guidance.

In terms of outer-loop remediation, the goal is to direct a learner to a source of information that will assist them in recognizing their error and correcting any mental model linked to that concept. The eM^2AP is being designed to support outer-loop remedial logic that enables an author to dictate how a learner transitions through a corpus of material based on student knowledge assessments and practice opportunity outcomes. This logic is based on the CDT and is focused on redirection to a rule or example quadrant for a concept when requested by the pedagogical model.

This outer-loop instance can take many forms. An initial approach is the idea of a system introducing a number of concepts to a learner before they are given the opportunity to practice their application. For this instance, an author may apply various checks on learning throughout the process to gauge a learner's cognitive understanding. When a misconception is identified, the system can adapt training by providing new material that covers the same concept, either by presenting a new representation of concept rules or by displaying a new example, with a hope that there is information in there to help correct the present error. The other approach is based off of performance information made available following a practice opportunity. If a training application can provide granular performance assessment so as to identify root causes of error, the eM^2AP can redirect a learner to the rule or example quadrants to review the concept identified as below expectation.

2.5 Manage Guidance

An effective pedagogical model manages tutorial guidance through system-initiated and student-initiated guidance requests. For both instances, this form of pedagogy is an inner-loop implementation in that learner-ITS communication takes place during a single learning event within a lesson. In terms of guidance, interventions are used to serve various components associated with the learning process (i.e., cognitive, metacognitive, and motivational), resulting in multiple functions for the purpose of regulating interaction. This includes guidance that provides a reinforcing function, an informing function, and/or a guiding or steering function [10]. Ideally, guidance on a cognitive level provides a learner with just-in-time information that assists them in attaining task goals as dictated by the type of learning event being experienced.

For student-initiated guidance, a learning environment requires resources a learner can use to seek additional information pertaining to a concept or to engage the tutor for assistance on how to proceed. Much of this interaction is based on a learner's metacognitive ability in recognizing resources available for assistance and knowing when best to use them. As for system-initiated guidance, the learner model is the input source for pedagogical decisions and provides real-time performance information on a concept by concept level. Dependent on the learner, the type of system-initiated guidance delivered should be adapted to better serve the ability levels of a given individual. This includes adapting the timing of feedback (i.e., what type of error constitutes the presentation of a guidance message) and the specificity of feedback (i.e., what type of content to provide in the guidance message). This is attributed to scaffolding techniques where the level of guidance is reduced as a learner exhibits more competence in a domain [11].

For the eM^{2}AP, guidance can be associated with each quadrant within the CDT. The goal of the eM^{2}AP is provide principles and heuristics for guidance delivery, and is intended to assist an author in establishing these practices in the context of their educational system. Student-initiated requests are most often attributed to help-seeking behaviors, where a learner recognizes their own impasse and actively seeks assistance. This can take place in any of the CDT quadrants. In terms of system-initiated feedback, the system is dependent on learner state information, which is only available in the recall and practice quadrants, unless it is informed by affect related information, which will be discussed below.

2.6 Manage Challenge

An effective pedagogical model is designed to manage the challenge level associated with an instructional event so as to influence learning and skill development by maintaining desirable difficulties during interaction. This type of pedagogical management is linked with both inner- and outer-loop adaptation. Managing challenge on a macro-adaptive capacity was highlighted in the managing practice opportunities, where as in this subsection, we discuss a model's ability to adapt challenge in real-time during a practice event. Specifically, this interpretation of challenge management is associated with practice environments that can have elements that can be manipulated in real-time by an ITS. An example is a game-based training environment in the military. Managing challenge levels could involve increasing or reducing the number of opposing forces, changing the weather conditions, or adding or removing distracters in the environment. The type of intervention that can be executed in this type of adaptation is dependent on options inherent to a selected training application.

In the eM^{2}AP, challenge is controlled by allowing an author to establish 'scenario adaptation' requests based on learner state information. These adaptations associate with both cognitive information (i.e., a practice scenario proving to be too easy or difficulty for a learner) and affective information (i.e., a learner is getting bored or frustrated during a practice scenario). In terms of establishing a set of heuristics that guide real-time challenge management, there is a lack of empirical evidence on how to handle this form of inner-loop pedagogy. Ideally, an ITS manages both guidance and challenge simultaneously through theoretical underpinnings linked to ZPD [4].

2.7 Manage Learner Affect and Motivation

An effective pedagogical model is designed to manage and influence a learner's affective state and motivation during an educational event. This form of pedagogy accounts for a learner's behavioral and physiological reaction to an event, and can dictate both inner- and outer-loop adaptations. These adaptations are dependent on the learning event being experienced and the state assessed by the ITS. In the instance of knowledge delivery, if a system can gauge a learner is getting bored while reading a passage, an ITS can intervene and provide new material on the same concept with the hope that it will re-engage the individual. In the example of a practice opportunity, there are many more options. If a learner is assessed as being bored during a scenario, the pedagogical model can increase the challenge. If a learner exhibits signs of frustration, the pedagogical model can provide guidance to assist the learner or it can reduce the challenge to better suit the individual's ability levels. The pedagogical model can also inform what to provide following practice based on affective states experienced during the interaction. There is an abundance of research on how to effectively model a learner's affective state, but there is little work on how best to use that information to inform pedagogical interventions.

To assess affective state that is communicated to the eM^2AP, GIFT includes a sensor module that takes in behavioral and physiological markers and outputs state determinations based on present classifiers. This state is combined with a learner's performance state in the learner model, which is then communicated to the eM^2AP. Currently, the eM^2AP can perform inner-loop adaptations based on affective information by executing a guidance function or adjusting challenge levels. Eventual work will be conducted so affect can influence outer-loop decisions on what will be presented next.

3 Conclusions and Future Work

In this paper, we present the components and processes being used to guide the development of GIFT's first domain-independent pedagogical model, the eM2AP. It is believed these components provide a framework to structure processes and functions that the architecture must be able to support. While much of the work is presented on a conceptual level, many of the functions described are currently available in the latest release of GIFT. While the functions themselves are available, how best to implement them is an open research question. The factors of interest include the types of pedagogical decisions a developer faces when creating a system (i.e., inner- and outer-loop adaptive strategies), the variables and modeling techniques used to trigger a defined strategy, and the requirements for its implementation within a specific learning context and environment [3, 12, 13]. In terms of future research, areas of exploration include curriculum sequencing, preventing system gaming behaviors, open student modeling techniques, using worked examples, learning through tutorial dialogs, and learning by teaching methods.

4 References

1. Mayes, T., Freitas, S., Review of e-learning frameworks, models and theories. JISC e-Learning Models Desk Study; Available from: http://www.jisc.ac.uk/epedagogy/ (2004).
2. Beal, C., Lee, H. Creating a pedagogical model that uses student self reports of motivation and mood to adapt ITS instruction. In: Workshop on Motivation and Affect in Educational Software, (2005).
3. Wang-Costello, J., Goldberg, B., Tarr, R., Cintron, L., Jiang, H. Creating an Advanced Pedagogical Model to Improve Intelligent Tutoring Technologies. In: The Interservice/Industry Training, Simulation & Education Conference. NTSA (2013).
4. Vygotsky, L., Zone of proximal development. Mind in society: The development of higher psychological processes: p. 52-91 (1987).

5. Kulik, C.-L.C., J.A. Kulik, Bangert-Drowns, R. Effectiveness of mastery learning programs: A meta-analysis. Review of Educational Research, 60(2): p. 265-299 (1990).
6. Merrill, M.D. The descriptive component display theory. Educational Technology Publications, Englewood Cliffs, NJ (1994).
7. Krathwohl, D.R. A revision of Bloom's taxonomy: An overview. Theory into practice, 41(4): p. 212-218 (2002).
8. Graesser, A.C., M.W. Conley, Olney, A. Intelligent tutoring systems. APA handbook of educational psychology. Washington, DC: American Psychological Association (2012).
9. Lesgold, A., Lajoie, S., Bunzo, M. Eggan, G. SHERLOCK: A coached practice environment for an electronics trouble-shooting job. Computer-Assisted Instruction and Intelligent Tutoring Systems: Shared Goals and Complementary Approaches. Hillsdale, NJ: Lawrence Erlbaum: p. 201-238 (1992).
10. Narciss, S. Feedback strategies for interactive learning tasks, in Handbook of Research on Educational Communications and Technology, J.M. Spector, et al., Editors. Lawrence Erlbaum Associates, Inc.: New York. p. 125-144 (2008).
11. Lepper, M.R., M.F. Drake, O'Donnell-Johnson, T. Scaffolding techniques of expert human tutors, in Scaffolding student learning: Instructional approaches and issues, K. Hogan and M. Pressley, Editors. Brookline: Cambridge, MA (1997).
12. Mathews, M.M., A Framework for Multiple Adaptable Pedagogical Strategies in Intelligent Tutoring Systems. In: International Conference on Computer Science & Software Engineering, University of Canterbury (2012).
13. Goldberg, B., Brawner, K. Sottilare, R., Tarr, R., Billings, D., Malone, N. Use of Evidence-based Strategies to Enhance the Extensibility of Adaptive Tutoring Technologies. In: Interservice/Industry Training, Simulation, and Education Conference. (2012).

Summarization during Tutoring: Implications for Developing Micro-Adaptive Tutoring Systems

Sandra Katz, Patricia Albacete, and Pamela Jordan

Learning Research and Development Center, University of Pittsburgh, Pittsburgh PA, USA
{katz, palbacet, pjordan}@pitt.edu

Abstract. Although research on human tutoring highlights the importance of a high degree of *interactivity* between the tutor and student, some instructional strategies that could be carried out interactively are implemented didactically in tutoring systems. This is especially true of summarization, a ubiquitous instructional strategy. We investigated summarization during human tutoring in order to determine how to refine decision rules that specify when and how summarization takes place in a tutoring system so that these rules can be made more interactive and adaptive. We describe the informational requirements for carrying out these rules and implications for developing an authoring framework that can provide this information to a dialogue management system.

Keywords: human tutoring, natural-language tutoring systems, instructional strategies, summarization, authoring frameworks, GIFT, TuTalk, Rimac

1 Introduction

Several researchers have proposed that the large effect sizes of human tutoring can be attributed to its interactive nature—that is, the high degree to which the student and tutor respond to and build upon each other's dialogue moves [1], [2]. However, an important line of research conducted to test this *interaction hypothesis* shows that it is neither how much interaction takes place during automated tutoring that is important, nor the granularity of interaction. Instead, what matters most is how *well* the interaction is carried out—for example, what content the tutoring system addresses, when (e.g., in the context of which activities?) and how (e.g., using which tutoring strategies? delivered through which types of media?) [3, 4]. In other words, interactivity during tutoring needs to be carefully managed.

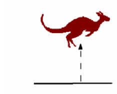
These important findings present several challenges to learning scientists and developers of natural-language tutoring systems. First, we need to determine which tutoring strategies are effective and for which types of learners. Second, because tutoring is essentially a linguistic process, we need to identify specific linguistic mechanisms that carry out these strategies. Several studies in the past decade have identified inctive dialogue patterns during tutoring whose frequency predicts positive learning outcomes (e.g., [5, 6]). Third, we need to specify decision rules that can guide a tutoring system in carrying out pedagogical strategies adaptively.

adfa, p. 10, 2011.

Table 1. Example of a post-problem reflective dialogue in Rimac

Problem: A kangaroo can jump about 2.5 m straight up. What is the magnitude of the take-off velocity?

Reflection Question: At what time during the jump (from the moment it leaves the ground to the moment it reaches the top of the jump) is the kangaroo moving fastest?



S = student; T = automated tutor (... = deleted dialogue)

S1: at the highest point *(incorrect response)*

Remedial dialogue implements Rule 3 in Table 2:

T2: Let's step through the reasoning. What is the magnitude of the kangaroo's velocity at take-off?

S3: .54m/s

T4: Good. Does the kangaroo's velocity change over the course of its upward jump or does the velocity stay the same?

S5: it gets smaller *(correct response)*

T6: what physics quantity represents change in velocity over time?

S7: acceleration *(correct response; triggers Rule 3)*

T8: Correct! Acceleration represents the change in velocity over time. In this case the acceleration is caused by gravity. This acceleration is pointing downward and it causes the kangaroo's upward velocity to get smaller…

End-of-tutoring summary

T9: To sum up, the kangaroo's velocity is greatest at take-off. It slows down as the kangaroo rises, until its velocity reaches 0 m/s at the top of the jump. Acceleration causes this change in velocity (velocity slows down to 0 m/s) and the acceleration is due to gravity.

Our research team has been addressing these issues in the process of developing Rimac, a natural-language tutoring system that scaffolds students in acquiring a deep understanding of physics concepts and principles, by engaging them in qualitative "reflective dialogues" after they solve problems and study worked examples [6]. (See Table 1.) Our goal is to specify tutoring decision rules that are empirically supported, domain independent, and more intuitive than those produced using automated approaches such as reinforcement learning (e.g., [3, 4]). Rimac's dialogues were developed using TuTalk, a dialogue development "toolkit" which has been used to build natural-language tutoring systems in various domains [7].

Our approach to deriving an initial set of decision rules to implement in Rimac can be summarized as follows. (See [6] for more detail.) We first identified patterns of collaborative dialogue exchanges in a large corpus of physics tutoring transcripts: 310 live tutoring sessions, in which one of seven tutors was paired with fifteen students. Interaction between the tutor and student was via teletype. We then conducted correlational analyses to identify relations whose frequency predicts positive learning outcomes and examined aptitude-treatment interactions. We described the context in which these potentially effective dialogue patterns typically occur and specified decision rules that capture relevant triggering conditions. We then implemented simplified versions of these rules within Rimac and are currently evaluating the system to determine whether these rules support learning—collectively, individually, and/or in groups defined by the tutorial strategy that they carry out. Table 2 shows a sample of these decision rules, expressed informally.

Table 2. Examples of tutoring decision rules to guide summarization in Rimac

End-of-tutoring "recap":

1. IF <*student response to problem or tutorial dialogue question* = correct>
   AND <*amount of steps or dialogue leading to response* = medium or high>
   → Recap line of reasoning that led to correct response

Summaries during tutoring:

2. IF <*solution step or student response to tutorial dialogue question* = correct>
   AND <*amount of tutor scaffolding moves leading to response* = 0>
   → Recap line of reasoning that led to correct response
3. IF <*current tutoring state* = remediation>
   AND <*student response to current tutorial dialogue question* = correct>
   → Recap line of reasoning that follows from student response
4. IF<*student response to problem or tutorial dialogue question* = incorrect or partially correct>
   → Summarize line of reasoning that leads to correct response
   OR scaffold student through main line of reasoning

These decision rules implement summarization—in particular, encapsulation of the line of reasoning that leads to a solution to a problem or answer to a question asked during tutoring, or that stems from a given problem-solving step. Our correlational analyses revealed that different types of line-of-reasoning summaries, as represented by these rules, predict learning [6]. For example, exchanges in which one dialogue partner (tutor or student) provides the steps in a line of reasoning that stem from, or lead to, a solution step expressed in his partner's turn predicted learning across ability levels (R=.65, p<.01), while tutor prompts for the student to summarize the reasoning that led to a correct answer to a problem or tutorial dialogue question predicted learning among high-knowledge students (R=.83, p<.05).

We take the view that language is purposeful action [8]. In essence, a summary abstracts the main points from a tutorial dialogue or any other instructional activity—for example, reading a text, watching a video, solving a problem, or playing an educational game. Summarization supports a wide range of instructional goals such as reinforcing facts and concepts, developing problem-solving scripts, and facilitating self-regulation of learning. It takes place dynamically, flexibly, and adaptively. As we illustrate presently, it can occur at the beginning of a tutoring session, at the end, and at various points in-between. It is typically didactic, but sometimes interactive.

In contrast, simulation of summarization in tutoring systems, including Rimac, is a far cry from capturing the level of flexibility and adaptability that we have observed during human tutoring. In response to this limitation and to the potential for summarization to support learning [9, 10], our goal is to make decision rules such as those shown in Table 2 more adaptive to students' cognitive and affective state. Towards this end, we are investigating summarization further in the tutoring literature and empirically. Although we also plan to do this with other tutoring strategies, this paper focuses on summarization—in particular, summarization of tutorial dialogue as opposed to other types of instructional activities. We first discuss what our analyses of summarization during naturalistic tutoring suggest about how summarization could be carried out more adaptively within tutoring systems. We outline the informational requirements of an adaptive tutorial dialogue system and describe how a domainneutral, interoperable ITS authoring framework such as the Generalized Intelligent Framework for Tutoring (GIFT, [11]) could be extended to support micro-adaptive implementation of summarization—that is, dynamic response to changes in students' cognitive and affective states.

2 Summarization During Human and Automated Tutoring

2.1 End-of-Tutoring "Recaps" of Main Points

Summarization happens in all forms of instruction—for example, end-of-chapter summaries in textbooks; recaps of classroom discussions or lectures, and of tutoring sessions with a human or automated tutor. In their extensive analyses of naturalistic tutoring sessions, Graesser and colleagues observed that summarization is one of the most frequent dialogue moves during Step 4 of their 5-step tutoring frame: scaffolding to improve an answer to a problem or question asked during tutoring [2], [12]. Frequent summarization is characteristic of both skilled and unskilled tutors [13, 14].

Natural-language tutoring systems, such as those developed within AutoTutor, typically simulate unskilled tutors' summarization practices, which Graesser et al. [14] describe as follows: "Unskilled tutors normally give a summary that recaps an answer to a question or solution to a problem. This summary serves the function of succinctly codifying a lengthy, multi-turn, collaborative exchange when a question is answered or a problem is solved" (p. 40). A tutoring system needs minimally adaptive decision rules to implement such end-of-tutoring recaps. The main parameter that determines whether a summary should be delivered is whether the topic under discussion has been adequately addressed during the dialogue, as reflected by the following AutoTutor decision rule [2]: "IF [quality of the cumulative collaborative exchange = completely correct] THEN [tutor supplies a summary or recap of the answer]" (p. 509).

Rule 1 in Table 2 similarly fires in Rimac when the student has arrived at a correct answer to a reflection question, after a series of questions that address the main ideas in the line of reasoning. See, for example, the summary at T9 in Table 1. Later versions of AutoTutor considered dialogue length, in addition to topic coverage, to determine whether to trigger a summary as the next dialogue move [12]: "IF [topic coverage = HIGH or number of turns = HIGH] THEN [select SUMMARY]" (p. 30). This rule ensures that the main points are extracted from lengthy dialogues and that short dialogues will not be summarized.

Typically, both skilled and unskilled tutors present end-of-tutoring summaries didactically. However, an alternative, which few tutors (skilled or unskilled) do, is prompt the student to generate a summary, perhaps with some degree of scaffolding from the tutor. Several tutoring system researchers have highlighted the potential benefits of doing so, in response to research which shows that generative activities such as summarization promote knowledge organization and retention [9,10]. To our knowledge, only one tutoring system, Guru, supports student summarization [13].

The choice between didactic and student-generated summarization provides a good example of how minimally adaptive decision rules such as those shown in this section and in Table 2 could be refined to support more adaptive summarization during tutoring. Under what conditions should the tutoring system choose each option? One relevant factor is the student's level of knowledge of the topic discussed. For example, if the student made few errors, or the dialogue was short, a brief didactic summary (or none) would be appropriate. Another important factor is dialogue history [15]; in particular, prior exposure to a didactic summary. If the tutor has already summarized the material covered at the end of a previous dialogue, the student might be ready to generate a summary, perhaps with scaffolding. In addition to refining summarization decision rules to select who generates a summary, these rules could be extended to select suitable presentation media: text, static graphics, and/or video. The main parameter to consider is type of content: does text suffice, or is visualization necessary? If the latter, is the material static or dynamic? Learner preferences and "styles" (e.g., is the student a more visual or verbal learner?) could also be considered.

Our analyses of summarization during human tutoring sessions revealed other types of summaries besides the didactic, minimally adaptive end-of-tutoring recaps of main points that pretty much define summarization in tutoring systems. We therefore suggest the need to broaden the view of what can be summarized to include anything that the tutor expects to be in the "world of discourse" that he or she shares with the student because it is, or will be, relevant to the current tutoring session. This can include the content of a lecture, lab, or textbook section; a conversation during a previous tutoring session—not just what was discussed during the current session. We present a sample of the types of summaries that we identified in the next section, in order to provide a "case study" of what more dynamic, micro-adaptive implementation of tutoring strategies would entail.

2.2 Summaries at the Start of Dialogue and Various Points Along the Way

Line-of-reasoning summaries throughout dialogues. Whereas Rule 1 in Table 1 fires only at the end of a tutorial dialogue or problem, after a correct solution has been reached (e.g., T9 in Table 2), Rules 2-4 represent summarization that can take place at various times. Like Rule 1, these rules are minimally adaptive because they mainly respond to the correctness of the student's answer to the tutor's current question. As we will illustrate presently, a higher level of adaptivity could be reached by taking other factors into account. Due to space limitations, we focus on Rule 3.

Rule 3 captures situations in which the tutor addresses an incorrect answer to a question asked during tutoring through a remedial sub-dialogue. At some point during remediation, the student answers a question posed by the tutor correctly. The tutor then completes the line of reasoning that would lead from the student's answer to a correct answer to the original question that triggered the remedial dialogue, as illustrated in Table 1. Here the student answers the Reflection Question incorrectly (S1). The tutor launches a remedial dialogue at T2. When the student answers correctly at S7, the tutor completes the line of reasoning at T8. The purpose of this summary appears to be expediency; the tutor doesn't want to spend so long in a remedial dialogue that the student loses track of the original question (the Reflection Question). The tutor returns to this question in the end-of-tutoring summary at T9.

In the human tutorial dialogues that we analyzed, these "tutor completion" lines of reasoning varied in their degree of interactivity. Sometimes the tutor would prompt the student for a few more steps in the line of reasoning that followed from the student's correct answer, instead of delivering all remaining steps. Once again, factors such as the student's confidence level and dialogue history should be considered, in order to refine this rule. If the student has difficulty getting through the dialogue, then fewer opportunities for failure (i.e., incorrect responses to the tutor's questions) might prevent the student from giving up. Also, if the dialogue history log indicates that the student has engaged in remedial dialogues about the same content in the past, then perhaps a summary would be appropriate in the current dialogue.

Summaries at the start of dialogue (or early on). Sometimes tutors state the most salient features of a problem or other instructional activity, or the main concepts and principles to apply, for example: "You need to think of these problems as the same in the sense that when you are dealing with problems where we have several forces acting the first equation that should enter your mind is Newton's 2nd law." Alternatively or in addition, the tutor might provide a sketch of the main steps to be taken to solve a problem. Cade et al. [16] refer to such "game plan" summaries as "highlighting" and note that they are characteristic of expert tutoring. Tutors might label the type of problem being addressed and their key features, compare the current problem with previous problems, and/or outline the main steps in order to help students do what domain experts do: classify a problem early on and invoke relevant solution schema [17]. Like end-of-tutoring summaries, these early-session summaries draw upon the tutor's expectations of a shared "world of discourse" with the student. Breakdowns in these expectations need to be repaired, as illustrated presently.

Tutors tend to offer highlighting summaries when a student has had limited exposure to a given type of problem or displayed difficulty solving similar types of problems in the past. Consistent with these knowledge state attributes, highlighting summaries are typically delivered didactically, although the tutor might guide the student in co-constructing a "game plan" at the start of a problem-solving session after the student has demonstrated increased skill in solving similar problems, for example:

T: Let's start from the beginning. Use Newton's Second Law. What does this law say?

S: Fnet = mass * acceleration

T: That is the equation you need to use. Now what are the forces acting on the object?...

Another type of summary that is presented early on in a tutoring session is a "mini lecture" about the domain content associated with a problem or other type of learning activity. Again, these summaries respond to the student's knowledge state. For example, we observed that physics tutors summarized a topic targeted by the current problem when they incorrectly assumed that the student had sufficient exposure to that topic. In automated tutoring, degree of exposure to domain content could be determined before a student starts to work on a tutoring system, through a questionnaire or pretest. This data could be used to macro-adaptively prime the tutoring system to provide "mini-lectures" about topics that the student has had limited exposure to.

Summaries to support self-regulation. Students sometimes exhibit poor learning habits during tutoring. Several physics tutors that we observed responded with corrective advice that might promote self-regulation—for example, reminders to memorize often-used equations; to express equations in terms of variables first and instantiate later. After the student solved the problem, tutors sometimes summarized this self-regulatory advice, for example: “OK that is what I got too but...next time, write the equations out symbolically first and then assign values before you do the actual calculation, so we can both follow what you are doing.”

3 Implications for Supporting Micro-Adaptive Summarization

The central aims of the preceding section were to show that summarization during human tutoring goes well beyond the didactic, end-of-tutoring “recaps” that are typically implemented in tutoring systems and to specify the types of information that would be needed by a dialogue system to simulate more flexible, dynamic and micro-adaptive summarization. This includes information about the learner’s knowledge state about domain content addressed during the dialogue; local and global dialogue history—for example, what topics have been covered during the current dialogue and during previous lessons? Has the content been summarized during a previous session and, if so, how (e.g., didactically or interactively; using which media?). Is the student ready to generate a summary, perhaps with scaffolding? Information about the student’s affective and metacognitive state is also important—for example, what types of self-regulatory advice should be recapped after a learning activity?

A modular, service-oriented, domain-independent framework such as the GIFT [11] will support management of this complex array of instructional information. The main “workhorses” are the Trainee, Pedagogical, and Domain Modules, which respectively model the learner’s cognitive and affective states, make decisions about what to teach and how to teach it (e.g., through which tutoring moves, using which strategies? etc.) and instantiate the preceding with domain content. As developers of the GIFT extend this framework to support micro-adaptive tutoring, the roles of these modules and inter-modular communication will need to be clarified—in particular, what types of information messages will each module send and provide? In addition, GIFT developers will need to consider how students’ responses and initiatives during tutorial dialogues can be used as input to the Trainee Module. As several tutoring researchers have noted, students’ dialogue contributions are one of the best resources for diagnosing the student’s knowledge about a topic [18]. Extensions to the GIFT such as these will greatly support the development of tutoring systems that interact with students as effectively as human tutors, perhaps more so.

Acknowledgement. This research was supported by the Institute of Education Sciences, U.S. Department of Education, through Grants R305A10063 and R305A130441 to the University of Pittsburgh. The opinions expressed are those of the authors and do not necessarily represent the views of the Institute or the U.S. Department of Education.

References

1. Chi, M.T.H., Siler, S., Jeong, H., Yamauchi, T., Hausmann, R.G.: Learning from Human Tutoring. Cognitive Sci., 25, 471-533 (2001)
2. Graesser, A.C., Person, N., Magliano, J.: Collaborative Dialog Patterns in Naturalistic One-on-one Tutoring. Appl. Cognitive Psych., 9, 495-522 (1995)
3. Chi, M., VanLehn, K., Litman, D., Jordan, P.: An Evaluation of Pedagogical Tutorial Tactics for a Natural Language Tutoring System: A Reinforcement Learning Approach. Int. J. Artif. Intell. Ed., 21, 83-113 (2011)
4. Chi, M., Jordan, P., VanLehn, K.: When is Tutorial Dialogue More Effective Than Step-based Tutoring? In: Trausen-Matu, S., Boyer, K. (eds.) ITS 2014. LNCS. Springer, Heidelberg, 210-219
5. Forbes-Riley, K., Litman, D., Purandare, A., Rotaru, M., Tetreault, J.: Comparing Linguistic Features for Modeling Learning in Computer Dialogue Tutoring. In: Lane, H.C., Kalina, Y., Mostow, J., Pavlik, P. (eds.). AIED 2007. LNCS. IOS Press, Amsterdam (2007)
6. Katz, S., Albacete, P.: A Tutoring System That Simulates the Highly Interactive Nature of Human Tutoring. J. Educ. Psychol., 105(4), 1126-1141 (2013)
7. Jordan, P.W., Hall, B., Ringenberg, M., Cui, Y., Rosé, C.P.: Tools for Authoring a Dialogue Agent That Participates in Learning Studies. In Luckin, R., Koedinger, K.R., Greer, J.E. (eds.). AIED 2007. IOS Press, Amsterdam (2007)
8. Searle, J.R.: What is a Speech Act? Perspectives in the Philosophy of Language: A Concise Anthology. In Black, M. (ed.) Philosophy in America, 615-628 (1965)
9. King, A. Comparison of Self-Questioning, Summarizing, and Notetaking-Review as Strategies for Learning From Lectures. Am. Educ. Res. J., 29(2), 303-323 (1992)
10. Ley, K., Young, D.B. Instructional Principles for Self-Regulation. Educ. Technol. R&D, 49(2), 93-103 (2001)
11. Goldberg, B.S., Brawner, K.W., Sottilare, R., Tarr, R., Billings, D.R., Malone, N.: Use of Evidenced-based Strategies to Enhance the Extensibility of Adaptive Tutoring Technologies. I/ITSEC. Vol. 2012. No. 1. National Training Systems Association (2012)
12. Person, N.K., Graesser, A.C., Kreuz, R.J., Pomeroy, V., TRG: Simulating Human Tutor Dialog Moves in AutoTutor. Int. J. Artif. Intell. Ed., 12, 23-39 (2001)
13. Olney, A.M., D'Mello, S., Person, N., Cade, W., Hays, P., Williams, C., Lehman, B., Graesser, A.: Guru: A Computer Tutor That Models Expert Human Tutors. In: Cerri, S.A., Clancey, W.J. (eds.) ITS 2012. LNCS. Springer, Heidelberg (2012)
14. Graesser, A.C., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R., TRG: AutoTutor: A Simulation of a Human Tutor. J. Cogn. Syst. Res. 1, 35-51 (1999)
15. Moore, J.D.: What Makes Human Explanations Effective? In Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society (1993)
16. Cade, W.L., Copeland, J.L., Person, N.K., & D'Mello, S.K. Dialogue Modes in Expert Tutoring. In: Schultz, R., Gentzoglanis, A. (eds.) ITS 2008. LNCS, pp. 470-479. Springer, Heidelberg (2008)
17. Larkin, J., McDermott, J., Simon, D.P., Simon, H.A. Expert and Novice Performance in Solving Physics Problems. Science, 208(4450), 1335-1342 (1980)
18. Person, N., Graesser, A.: Fourteen Facts About Human Tutoring: Food for Thought For ITS Developers. In: AI-ED 2003 Workshop Proceedings on Tutorial Dialogue Systems: With a View Toward the Classroom. 335-344 (2003)

Instructional Strategies in Diagram-based ITSs: Lessons Learned from Two Tutoring Systems

Enruo Guo[1], Stephen Gilbert[2]

[1] Department of Computer Science
[2] Department of Industrial & Manufacturing Systems Engineering
Iowa State University, Ames, IA 50011, USA
{enruoguo, gilbert}@iastate.edu

Abstract. Unlike text-based input to an intelligent tutoring system, a diagram is perceived as a whole state; the operation sequence is less important. Traditional step-wise coaching is not as appropriate in diagram-based intelligent tutoring systems (DITS). From two previous tutoring systems, StaticsTutor and Thermo Cycle Tutor, we propose cross-domain pedagogical guidelines for DITS. In particular, instruction needs to be mapped to a hierarchical understanding of the diagram, where each level focuses on different characteristics of the drawing. Also, instruction needs to address conceptual knowledge and procedure expertise separately. Some practical suggestions are described to achieve these goals, such as 1) different tolerance for error at different level of evaluation, 2) use of Q&A to resolve diagram ambiguity and 3) early loading of expertise that is important for avoiding difficult-to-fix diagrammatic states.

Keywords: intelligent tutoring system, diagram, instruction, pedagogy

1 Introduction

Given the comprehensive advantages of pictorial representations, diagrams play a big role in scientific cognition, e.g., free-body diagrams in physics, Temperature-volume (T-v) diagrams in thermodynamics, circuit diagrams in electrical engineering, and Unified Modeling Language (UML) diagrams in software engineering. Cognitive models of graphics comprehension [1] propose that graphics comprehension involves interaction between bottom-up perceptual processes of encoding information from the graphic as well as top-down processes of applying graph schemas and domain knowledge, which makes it a challenge to teach students how to use diagrams to represent information.

In this paper, we discuss lessons regarding pedagogy that we learned from two diagram-based tutoring systems, and provide cross-domain guidelines for the design of future diagram-based intelligent tutoring systems (DITS).

adfa, p. 18, 2011.

2 Background and previous work

We have designed and implemented two diagram-based intelligent tutoring systems in engineering statics and thermodynamics courses: StaticsTutor [2] for free-body diagrams, and Thermo Cycle Tutor (Guo et al., in preparation), for T-v diagrams of refrigeration cycles. Even though they focus on different domains, both feature pedagogy aimed at helping students' conceptual understanding and decision-making at the earliest stage of problem framing.

StaticsTutor was developed to analyze student-drawn free-body diagrams and recognize misconceptions without requiring numerical force values or the need to provide equilibrium equations. Preliminary results with 81 engineering undergraduates in fall 2013 showed that StaticsTutor could detect students' misconceptions that were categorized as “missing basics,” “hinge issue,” “rope issue,” and so on. A post-survey indicated an overall positive experience with the tutor with a mean usability score of 3.5 (SD 1.11).

The Thermo Cycle Tutor implemented a teaching pedagogy (Hagge et al., in preparation) based on decision-making, where class concepts are posed as a set of simple questions that can be answered for all problems in the thermodynamics course. In fall 2013, 42 undergraduate engineering students were given a pre-test on refrigeration cycles and then given the Thermo Cycle Tutor to complete a homework problem. They then took a post-test. Students' post-test scores improved from 70% to 89% on average. To test retention, they were given a second post-test after four weeks, and they scored an average of 81% better than the pre-test with no additional lectures on refrigeration cycles.

Both tutors faced the challenge of how to analyze the students' diagrams computationally, and how to give appropriate feedback. Pedagogical questions that arise include, "If there are multiple issues with the diagram, which issue should receive feedback first?", "Given an error in the diagram, what can I infer about the student's misconceptions, if any?", and "When should I evaluate the diagram, at each step of construction, or only at the end?"

2.1 Previous work on diagram interpretation and DITS

Koffman and Friedman [3] designed an early instructional tool for diagramming to assist beginning programmers in learning to make a computer-aided flow diagram. They emphasized the problem-framing aspects of diagram planning, and wanted students to use the diagrams to learn the program logic before implementing the code. Usually it is difficult to analyze a diagram at each step of its construction, because there are typically graphic elements that must be added one at a time in no particular order), and the diagram can frequently exist in non-well-formed states that cannot be fully anticipated by the tutor author. However, in Koffman and Friedman's case, the linear structure and the level of granularity of their diagram components helped this system avoid these open-ended ambiguities that usually occur during construction.

Constraint-based modeling (CBM) has been adopted in ITS community, where the domain knowledge is represented as a set of constraints. By focusing on violated constraints, CBM tutors are able to generate instructional actions even without having expert solutions. Instructional feedback is generated by focusing on one genuine misconception if more than one constraint is violated [4]; frequently one misconception will cause the violation of several related constraints. COLLECT-UML [5] is a CBM tutor to teach object-oriented design which supports both single user and multi-user for collaboration purpose. However, as Py. et al., [6] noted, instructional feedback directly generated from a violated constraint might not be a good solution from a pedagogical point of view. They separated the diagram diagnosis output from instructional feedback. However, they didn't have much emphasis on diagram structure and how to generalize it.

Futrelle [7] attempted to apply levels of abstraction to diagrams by offering a diagram constraint grammar and process for automatic computational diagram analysis loosely based on computer vision. His approach, however, was focused on analyzing the diagrams, rather than tutoring using diagrams. Tutoring through a diagram not only needs to analyze the diagram, but also to understand the student's knowledge and misconception within the abstraction in the diagram. Thus, a mapping between levels of abstraction in the diagram with domain-wide conceptual knowledge is highly desired. Here, we proposed a three-layer abstraction for diagrams used in engineering domain, where errors in lower layer need to be addressed first as it is more fundamental. Also, the three-layer abstraction follows a general process of knowledge acquisition: from superficial to deep, from rough to detailed.

It is worth mentioning the pedagogy in the Andes tutor [8] that allows students to pursue different correct solutions during problem solving instead of limiting them to a predefined optimal solution. A solution graph representation, which contains several types of nodes, is used to model all possible solution paths, upon which a Bayesian network is built. Then Bayesian inference is applied to designate student's current goal node and a rule-application node where the student is stuck for lack of knowledge. A hint is then generated to coach that knowledge accordingly. Even though Andes focuses on text-based inputs, this step-by-step coaching strategy also applies to diagram-based systems. However, there are some differences that make pedagogy in diagrams challenging: 1) A diagram should be perceived as an entire state, no matter when and how an element is added to the diagram. Step-by-step coaching needs to be redesigned appropriately. 2) Even though sequence is less important in a diagram, it does require a series of actions to be applied in order to meet a certain requirement in a given state. This means that the diagram must be properly defined as several sequential stages, where each stage represents certain conceptual understanding. Within a stage, the sequence of actions do not likely matter.

3 Instructional guidelines for DITS

Guideline 1: Instruction needs follow hierarchical diagram understanding.
Even though diagrams vary across domains, there are usually underlying concepts that drive core questions that should be answered during the assessment process. The core questions can be defined through an expert module, which might vary based on

the expert's instructional and pedagogical preferences. However, a general architecture that fits in a cross-domain evaluation system is highly desired, e.g., a version of the popular ontology editor *Protégé* customized for DITS authoring. For this purpose, we propose three levels of hierarchy for diagram evaluation. Before defining the levels theoretically, we offer an illustrative example from thermodynamics. Figure 1 shows an example with T-v diagrams. These diagrams are used to abstractly represent how pressures, temperatures, and volumes change within a mechanical refrigeration system, which may contain compressors, pumps, valves, etc. The Thermo Cycle Tutor basically asked six questions: 1) Is a vapor dome needed? 2) How many pressures are present in the cycle? 3) How is a pressure line drawn on a T-v diagram? 4) How should phase change P and T be labeled on the diagram? 5) What are the P, T, v relations for each component? 6) How can the problem information, and the decisions above uniquely identify each state?

While these questions are particular to refrigeration cycles, they have the following characteristic which applies across domains: some of them focus on the student's conceptual understanding (1, 2, 5), and some focus on the procedural skill of how to make a diagram appropriately (3, 4). Of course, these two aspects are tightly coupled, and some questions apply to both (6).

It is noteworthy that the six questions follow a hierarchical understanding of the diagram. At Level 1, nine straight line segments are recognized on a vapor dome (Figure 1a), where each three connected segments represent a pressure line. At this level, the message that the diagram conveys is simply that there are three pressures in this system. At Level 2 (Figure 1b), more details are shown: some text labels are attached to the pressure line segments at the right-hand side, and tick marks are added to show the phase change temperatures. These additions give the viewer more concrete information about the exact value of the pressures and phase change temperatures. Then, at Level 3 (Figure 1c), by adding some points with labels on the pressure line segments, the diagram brings in details on the state information and how it interacts with the pressure and phase change temperatures.

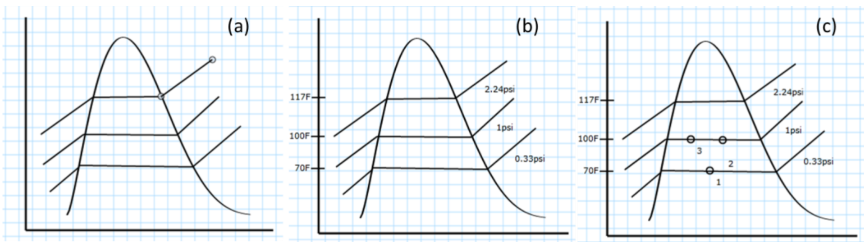


Fig. 2. Three levels in a refrigeration cycle T-v diagram. (a). A vapor dome with three pressures. (b). Labels of phase-change temperature and pressure values were added. (c). State information was anchored on the pressure line.

To generalize the levels just described, Level 1 focuses on basic graph-style structures and the spatial relations between each other. At Level 1, the tutor has a rough idea of what components are present and their connections. To give feedback at Level

1, the DITS needs to incorporate domain knowledge. Level 2 focuses on object identities and their object-type-specific relationships. At this level, attributes about an object will be identified through domain knowledge and some text labels. These include object name and possible values relate to object. Spatial relationships from Level 1 will be transformed to more specific numerical relationships. As is shown at Figure 1b, the numeric value has been explicitly shown in each pressure line, so it is easy to tell the second pressure is 0.67 (1 - 0.33) psi higher than the first pressure, whereas Figure 1a only tells the second pressure is above the first pressure. Level 3 focuses on properties or children of Level 2 objects. In this level, details on Level 2 objects will be revealed and examined. The details could comprise sub-objects that constitute a Level 2 object, or a sub-object that is attached to a Level 2 object but itself is not considered as a basic structure at Level 1. Instructional feedback can be composed based on the level of specificity. The lower level error should be tackled first, as it is more fundamental and serves as the basis of the higher level object. For instance, if a Level 1 object is missing, it doesn't make sense to correct a Level 2 object as by definition its structure is based on Level 1 object. We propose this "divide-and-conquer" strategy where each piece can be mapped to one or more states of student's understanding.

Guideline 2: Customized instruction from individual to individual.

A diagram embeds a student's conceptual understanding, while evaluation by the expert module is trying to infer it. Thus evaluation questions need to be somehow mapped to domain-wide concepts. In order to track student's knowledge on each concept, it is necessary to register them in student model. A complete set of evaluation steps will be applied to the student's diagram at the beginning, as the domain-wide concepts in her student model is not determined. As she finishes a problem, her student model will get updated, with some concepts being checked as passed. How to define a concept as mastered is not in the scope of this paper. The next time, the expert module should consult her concept inventory before initializing the tutoring process. For example, we have implemented six questions in the expert module in the Thermo-Cycle tutor. However, for the student who has understood phase change temperature, how to use the reference form to locate the value, and how it should appear in a T-v diagram, expert instruction would skip question 4, which checks the label of phase change temperature in the future T-v diagram evaluation.

Guideline 3: Separate conceptual knowledge from procedure expertise.

As the evaluation engine assesses a student's drawing based on the elements defined in the expert module and gives instructional feedback, there are some practical issues. How to handle these issues will affect the usefulness and quality of instructional feedback, student's engagement, and finally affect learning gains.

In most cases, when a student starts to frame a problem, she doesn't have a clear idea of what information needs to be drawn, and what might be a proper way to represent it. So a drawing with incomplete elements might be submitted to the tutoring system for help. In order to provide the most useful instructional feedback, the tutor is desired to "read" information from the drawing. The information includes what might be her intention, what knowledge she might have known or not known and what other knowledge needs to be further determined from the drawing.

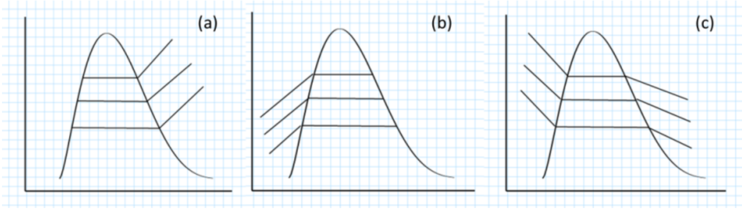


Fig. 3. Examples of wrong drawings on refrigeration cycle T-v diagram. (a) and (b): incomplete pressure line. (c). Wrong pressure line representation which uses a negative slope.

The student might not be able to represent her conceptual understanding correctly in the drawing at the beginning. However, when she gets familiar with the procedure or gains expertise on how to represent the knowledge, she can focus more on the conceptual part. So a tutoring system needs to set apart these two types of questions, and give instructional feedback separately. Figure 2 (a) and (b) show two examples of sloppy drawing where a vapor dome and three coupled lines were present. To be considered as a correct representation of a pressure line, three connected segments should be included. However, the incomplete drawing still implies that the author thought there were three pressure lines. Assume three pressure lines are the correct answer in expert solution. In this case, the tutor's instructional feedback should focus on how to help them to construct a pressure line, instead of correcting the number of pressures because zero "true" pressure lines were detected in the diagram.

Figure 2 (c) shows an incorrect representation of a pressure line since slope in the side lines should be positive. Many beginners tend to borrow the shape that they learned in P-v diagram, which is negative, and apply it to T-v diagram. Even though the tutor cannot detect pressure lines, it should be able to probe student's intention as three pressures in the system, and give her appropriate instruction such as "This is not a P-v diagram. Would you like help on drawing pressure lines in a T-v diagram?" To facilitate this strategy, we provide guidelines for a DITS evaluation engine.

Diagrams require different tolerances at different levels of evaluation.

As we mentioned earlier, instruction could be based on evaluation of a three-level hierarchical structure of the drawing. A different tolerance could be assigned to each level. Tolerance could be a concrete value applying to check functions such as 10%, or it could have a conceptual definition, i.e., a pressure line can have two or three connected segments. A larger tolerance should be used to detect whether a basic structure is present. For instance, to check the number of pressures, a pressure line can be recognized if one horizontal line in the middle and two positive sloped lines at sides are found and well connected. The tolerance for gaps between line segments could be a large value. After it passes the Level 1 check, and goes to the next level, which checks the representation correctness, this tolerance would decrease, and any big gap would need to be filled by moving line segments closer to their neighbor. Another two examples with incomplete pressure lines (Figure 2a and 2b) would pass the number of pressure check, given a higher tolerance on recognizing a pressure line. After a student passed the Level 1 check, the incomplete pressure line issue would be addressed due to a lower tolerance on how a pressure line should be drawn.

Diagrams' inherent ambiguity can be resolved with Q&A.

Due to the intrinsic complexity and ambiguity of a drawing, it is safer to confirm the information that is conveyed in a drawing with some text inputs. For example, if the drawing fails on a *number_of_pressures* check, a multiple choice question pops up and ask student to choose how many pressures are there in the system. If it is correct, it indicates that the student's conceptual understanding is correct, but some procedural issue caused the failure, e.g., she accidentally clicked the submit button without finishing the pressure line. Another example is shown in Figure 3. The student did a good job on drawing pressure lines, labeling pressure and phase change temperature, and anchoring points on pressure lines to show the state changes in each component. However, feedback from the tutor said "There appears to be some misconceptions about the specific volume change in a compressor." Then tutor the directed her to three multiple choice questions regarding pressure, temperature and specific volume change in a compressor. She answered all the questions correctly and was told to "modify state 3 and 4 to reflect this." These successful answers imply that the student understood knowledge in a compressor, but didn't incorporate it into the drawing.

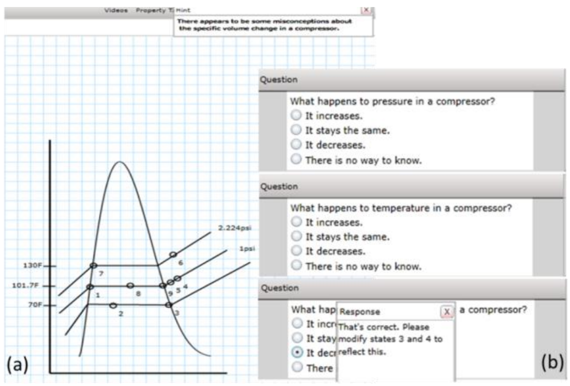


Fig. 4. (a). A refrigeration cycle T-v diagram. (b). Three windows that displayed questions about pressure, temperature and specific volume change in a compressor.

Conceptual and procedural performance in diagrams can be tightly coupled.

This problem is critical and stems from the fact that some aspects of constructing the drawing can make it difficult to edit elements later. This situation can frustrate a student if it occurs late in the problem solving process. As is shown in Figure 3, after the student realized state 3 should have a larger volume than state 4 (which means state 3 should appear on right side of state 4 in the T-v diagram), it is impossible for her change it in the diagram because there is no room. However, the student would not realize this issue until she reached this step if she didn't have much experience on solving this type of problem before. To alleviate this form of unnecessary frustration, when a particular problem is initialized by student, the evaluation engine should be able to load some practical expertise information about the base objects, e.g., the shape of the vapor dome should not be too thin and the distance between the horizontal lines should be greater than some percentage threshold.

4 Conclusion and future work

In this paper, we discussed cross-domain pedagogical strategies in diagram-based tutoring systems. In particular, instructional feedback needs to be mapped to a hierarchical understanding of the diagram. Personalized evaluation is desired which is based on student's current knowledge state. Also, it should be able to separate conceptual knowledge from procedure expertise. To achieve that, we proposed: 1) allow different tolerances at different level of evaluations, 2) use Q&A to reduce ambiguity, and 3) determine if conceptual knowledge can be applied by procedure expertise in the current drawing. In the future, we will design a general authoring tool for DITS to support the above pedagogical strategies, allowing instructors to define a) concepts in the knowledge base, b) objects and tolerances in each hierarchical level, c) evaluation pieces which link to one or more concepts and d) guidelines of procedural expertise.

References

1. Carpenter, P.A., Shah, P.: A model of the perceptual and conceptual processes in graph comprehension. *Journal of Experimental Psychology*: Applied, 4 , pp. 75–100 (1998)
2. Guo, E., Gilbert, S., Jackman, J., Starns, G., Hagge, M., Faidly, L., Amin-Naseri, M.: StaticsTutor: Free Body Diagram Tutor for Problem Framing. *In the Proceedings of the 12th International Conference on Intelligent Tutoring Systems*.(2014)
3. Koffman, E. B. and Friedman, F. L..: A computer-aided flow diagram teaching system. SIGCSE Bull. 8, 1, pp. 350-354 (1976)
4. Mitrovic, A.: SQL-Tutor: A preliminary report. Technical report TRCOSC 08/97. Computer science department. University of Canterbury (1977).
5. Baghaei, N., Mitrovic, A.: A Constraint-Based Collaborative Environment for Learning UML Class Diagrams. *In the Proc of the 8th conference on Intelligent Tutoring Systems (ITS'06)*, 176–186. Jhongli, Taiwan (2006).
6. Py, D., Alonso, M., Auxepaules, L., & Lemeunier, T.: Design of Pedagogical Feedbacks in a Learning Environment for Object-Oriented Modeling, *Proceedings of Educators Symposium at the 11th IEEE International Conference MODELS*, Toulouse, France pp.39–50 (2008)
7. Futrelle, R. P.: Strategies for Diagram Understanding: Generalized Equivalence, Object/Spatial Data Pyramids, and Animate Vision. *Proc. 10th ICPR, Int'l Conf. Pattern Recognition*, vol. 1, pp.403-408 (1990)
8. Conati, C., Gertner, A., VanLehn, K., & Druzdzel, M.: On-line student modeling for coached problem solving using Bayesian networks. In A. Jameson, C. Paris, & C. Tasso (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97*.

Pedagogy for Computer-Based Tutoring Systems Based on Experts Constraints in a Turn-Based Game

Robert Pokorny[1] and Benjamin Goldberg[2]

[1]Intelligent Automation, Inc., Rockville, MD 20855
[2]United States Army Research Laboratory, Orlando, FL 32826
{bpokorny}@i-a-i.com; {benjamin.s.goldberg.civ}@mail.mil

Abstract. Simulation-based instruction can be motivating for students. Content students learn can be directly applied to improve real-life performance. Simulation-based instruction will be more efficient if instruction adapts to each student's needs. Adapting instruction to students' needs requires (1) identifying topics that should be remediated and (2) guiding students through an instructional interaction which improves their ability at the identified topic. The process of developing an automated scoring system to specify instructional topics with open-ended simulations begins with experts scoring and then critiquing student performance. This data is transformed into scoring rules. The scoring rules can be automated and applied in real time to drive student remediation needs. The instructional content is based on domain knowledge specified in summaries of good practice. The instructional format follows current instructional theory, including situated cognition, scaffolding, interactions in which students create knowledge structures, and considerations of cognitive load. The application of these principles is illustrated within a turn-based instructional game.

Keywords: situated cognition; scaffolding, instructional interactions, cognitive load; assessment; critiques; cognitive task analysis

1 Introduction

Students can benefit by learning within simulations. Simulations can motivate students by providing an interesting context which captures students' attention; instruction in simulations enables students to easily transfer skills gained from the simulation directly to real-world conditions. While simulation-based instruction (SBI) has many advantages, instruction within simulation is historically more expensive and difficult to construct than more common, de-contextualized instruction. In addition to the costs and difficulties of creating simulations, further difficulties of SBI include (a) assessing student performance in real-time and using it to guide trainees to content they need to learn and (b) developing replicable, theoretically grounded approaches to instruction within simulations.

Widespread use of SBI that adapts to student needs face three major challenges: first, the assessment challenge is to identify skills and knowledge that are responsible

adfa, p. 26, 2011.

for students' performance weaknesses within the simulation. The second challenge is designing effective instructional interactions in which students gain knowledge while maintaining the flow of the scenario within the simulated environment. The third challenge is to design affordable SBI development processes so that organizations will choose adaptive SBI.

This report describes an approach to affordably develop SBI. The instructional approach collects scores and critiques from experts reviewing student performance, and transforms expert input into an automated scoring system. The instructional approach is similar to the Coached Practice Environment described by Lesgold and Nahemow [1]. The Coached Practice Environment places students within a simulation environment and encourages them to learn-by-doing within the simulation. As students progress through problems posed by the simulation, they may face obstacles that they cannot overcome on their own. The automated coach provides a series of hints to help them overcome performance difficulties. More in-depth reflective reviews of the challenge occur during post-problem reflection. The current paper reports an efficient approach to assess performance in complex simulations, and an instructional approach that provides short interactions that guide students in constructing knowledge that improves performance.

1.1 Approach to Assessment

Our assessment approach elicits and captures experts' knowledge as experts review student performance within a simulation. Experts who are knowledgeable about the domain and the simulated scenario are able to reliably estimate overall quality of student performance and provide rationale underlying their assessment. We then use expert reviews to develop automated assessment systems that take student performance input, apply scoring rules based on experts' scores and critiques, and produce assessments that (a) mimic experts' ability to rate overall quality of performance [2] and (b) identify patterns of performance that comply or violate good procedure [3].

The assessment approach uses a method to capture expert policies that we call Performance Evaluation through Expert Review (PEER). PEER begins by recording student performance within the simulation. These records are transformed into a presentation that is easy for experts to understand, enabling them to easily review and assess student performance. Experts' knowledge is elicited by asking experts to (a) rank order performance by overall quality; (b) assign scores that reflect overall quality; and (c) critique performance and justify assigned scores. The critiques are then classified by the analyst into semantically related categories and sub-categories. The focused comments enable analysts to summarize good practice. Performance indicators are created which identify compliance or violation with good practice. Rule sets that use the indicators of compliance or violation to good practice are created to yield scores that approximate experts' scores of overall quality. The assessment system is refined until the scores from the scoring system closely mimic experts' scores. The scoring system is validated by taking data from new students, and measuring the agreement between expert raters and the automated scoring system.

This approach differs from a more common ITS assessment approach in which actions are probabilistically linked to cognitive variable [4]. First, the PEER scoring system does not score each action that a student takes; rather, experts' comments identify patterns of activity and relations between concepts to assess performance. Second, the PEER scoring system does not use a probabilistic approach to assessing student performance. Uncertainty in performance is addressed by the triggers that experts use to identify compliance or violation with good practice. Third, the PEER-based automated scoring system focuses on performance characteristics that become targets of remediation rather than underlying cognitive variables. PEER identifies patterns of weakness that experts notice and used to initiate a training remediation. Thus, PEER-based remediations are often at a higher level of analysis than other ITS. This approach is consistent with a change in one view of instruction that has been well expressed by Sack, Soloway, and Weingrad [5]. They suggest that the concept of instruction in the early days of ITS assumed that learners skill and knowledge could be diagnosed to a fine granularity; remediation consisted of adjusting the details of students' knowledge and skill structure. Sack, et. al. suggest that this view of detailed knowledge review, which they refer to as a "learner as consumer" view, should be replaced by a view of "learner as constructor". According to this view, students improve performance not by targeted replacements of new thoughts, but by constructing a larger knowledge structure that incorporates the change of a specific fact and then integrates this change into a broader approach to understanding and solving problems.

1.2 Approach to Instruction

Our instructional approach is to design a learning environment which applies current instructional principles to SBI. These include situated cognition; instructional targets selected based on student need; interactions that help students construct skill and knowledge; and considerations of cognitive load. The principle of situated cognition is attained by embedding instruction within a simulation of the task environment [6]. Targeting skills and knowledge based on assessment of performance leads to focusing instruction on performance areas on which students should improve [7]. Instructional interactions that help students understand what they need to learn should in them in acquiring and applying the knowledge they need in the way they need [8]; and cognitive load must be factored in so as not to overwhelm and confuse students [9].

We will describe how to apply these principles, as the relationships between them are complex. While each principle makes sense and has demonstrated effects, the application requires design balance and pilot testing, as paying too much attention to one principle at the expense of another can lead to inefficient learning. An example of how these principles provide different guidance can be observed when trying to (a) address student weaknesses, while (b) considering restrictions due to cognitive load.

For example, targeting skills for remediation is somewhat at odds with managing cognitive load. At any point during a simulation, there may be many ways in which a student violates good practices and should be remediated. But students also have limited cognitive capacities, and presenting too many remediations will be counter-productive. Thus, the intent to remediate students on a wide variety of topics must be balanced with considerations of cognitive load. The approach we have taken in our instructional design is that during problem solution, only one topic, the one that is calculated as the worst, becomes the target of an instructional intervention. During a review of the entire problem, for example, during a post-problem reflection or during a review activity, all topics are addressed.

The design of the instructional interaction must take into account many considerations as well. Some of these factors include that the interaction should be (a) effective, (b) short and non-disruptive so it does not excessively interrupt the flow of the challenge presented by the simulation, and (c) the interaction should provide scaffolding so that the student performs as much as possible on one's own. Mechanisms underlying the view that students learn by interaction has been studied extensively by Chi [8] and is consistent with the view expressed earlier of considering students as constructors, not consumers. Combining the assessment that identifies the target with the largest violation of good practice with the interaction of instruction yields a remediation in which students are initially given a small clue regarding the focus of their largest violation. If they are able to use this to improve their performance, great; if they are not, they are given a more directed clue in the form of a question. If they do not improve their performance, they are then told what actions they should have taken. We will next put these theoretical pieces together by describing a use case.

2 Use Case

The targeted instructional system is a game that is used to teach Army leaders how to conduct counter-insurgency operations. This game, "UrbanSim", was sponsored by the Army's Research Development and Engineering Command (RDECOM) and built by the University of Southern California's Institute of Creative Technologies. This is a turn based game in which students are conducting counter-insurgency operations in a simulated city with a population of 300,000. In any turn, the student will assign 11 different resources to a specific mission. A mission can include repairing structures; communicating with civilians, civilian leaders, or military leaders; conducting military operations; or supporting host nation government activities such as training or recruiting for police or Army personnel. After the student selects the activities for each of the 11 resources, the student submits the order. The simulation executes the order and presents their effects. These include (a) changes in conditions of the simulated city, and (b) attacks on United States or government forces and (c) significant achievements. This game is a complex environment; Vogt, et. al. [10] reported $5*10^{27}$ possible paths through the game.

2.1 Applying PEER and Building an Assessment System

In applying PEER to the complex simulation environment of UrbanSim, we begin by collecting student traces (log files of simulation activity), and then transform them into a format that is easy for experts to interpret. An example of the presentation of student data that experts reviewed is shown in Figure 1. The student's actions for a turn are shown in the top half of the page, with actions shown on the city map on the left, and presented on the right side in text. The results of the actions are shown in the bottom half of the screen in terms of condition scores on the left side and listings of results on the right side (e.g., the percentage of the population supporting the host nation government is 26%).

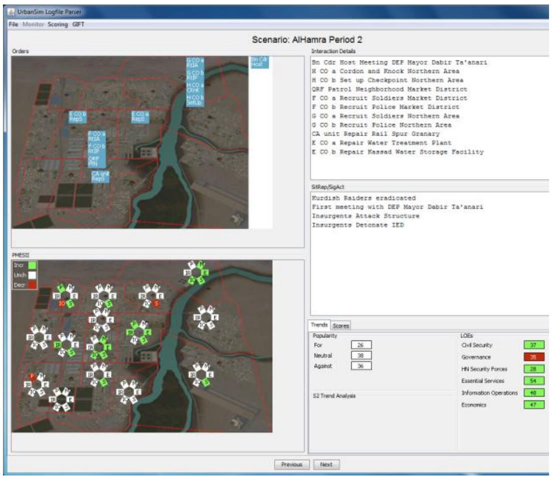


Fig. 5. Example Presentation of Student Data for Expert Review

Experts reviewed student actions, then analysts collected (a) experts' judgments of overall quality and (b) their critique of students' performance. Three senior leader experts critiqued and assigned scores reflecting overall quality to 15 student records. The correlations across experts ranged were .5, .72, and .74, all $p < .05$.

After seeing that experts' judgments did in fact correlate with each other, analysts categorized their critiques based on semantic similarity. For example, some critiques described conditions of security. Security became a high level category. The first two columns of Figure 3 show the process of collecting and categorizing the critiques.

Security actions were further divided into more refined categories. Some critiques from experts described security actions as being too harsh; others described security actions as too lax. Each of these became sub-categories under the security category. For each policy, there is a description of the policy and deviations from that policy. The development of policy is shown as the third column of Figure 2.

Next, analysts developed scoring rules. Compliance with good practice has positive scores; violations of good practice have negative scores. The weights of each rule are initially based on comments from experts about the importance of each factor considered; these values are revised so that the automated system yields scores that closely mimic the average of experts' scores of overall quality.

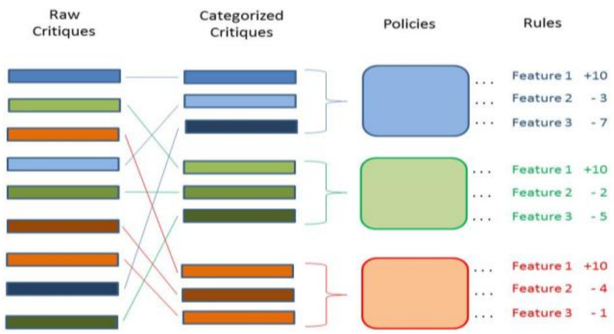


Fig. 6. Diagram of PEER Process

2.2 Application of Assessment System to Pedagogy

Scoring rules identify the category of performance that is weakest and most in need of remediation. By targeting the student's weakest sub-scores, the instruction focuses on the topic at which the student is weakest. Improving each student's poorest topic should efficiently raise their overall quality score. y (see figure 3).

In addition to responding to regular remediation scores, there are some egregious violations of good policy that require an immediate remediation. In UrbanSim, for example, a trainee might shoot all the civilians in the city. For each sub-score, we will determine a criterion that corresponds to a 'poison score'. If the trainee on any turn exceeds the poison score, the remediation related to the violation is immediately presented to trainees.

Security Sub-score	Ongoing score	Current score	Topic to be triggered
Too Lax	-1.5	-1.0	***
Too Strict	0	0	
Asad killed	0	0	
Insurgent groups persist	-1.25	-1.25	

Fig. 7. Example Selection of Worst Violation

2.3 Instructional Design

The most common form of remediation will assist students recognize performance weaknesses. For this type of remediation, the instructional design uses three levels of coaching. The pattern for instructional remediation is shown in Figure 4. The first level focuses on the high level policy that the student has misapplied. The student is given a question to direct the student to this policy; after the student answers the question by entering a short response in a text box, an expert's answer to the question is presented. The student takes another turn of the game, and enters another set of actions. If the student's actions related to the previous turn's hint now comply with good policy, the student's improvement is noted. If the student's interaction still contains a violation of that principle, a second level hint is given. This hint directs the student's attention to the expert's thought processes that lead the expert to the correct action. Again, the student answers the question, and is presented with the expert's response. The third level provides students with the answer to the question. The general format for this hint is, "Given factors YYY, what actions would you take in regards to ZZZ." The student answers the question, and then sees an experts' answer to this question.

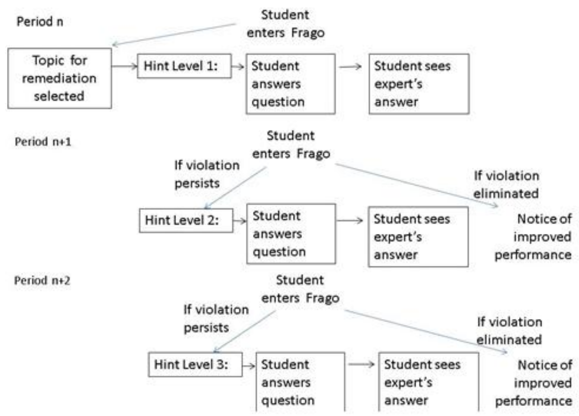


Fig. 8. Example Instructional Flow for Selected Remediation

Problem reflection.

A post problem reflection is presented to help students solidify what they learned from a scenario. This post-problem reflection is different from the ongoing training remediations, as the reflection is an intentional time to review and reflect on earlier actions, without the challenge presented by the scenario. When UrbanSim is used in an instructor-led class, the instructor holds a review of student activity at the middle of the game and at the end of the game. Our instruction will follow a similar policy, and conduct an "update briefing" with the students after the 8th turn, and an "after action review" after the last turn.

The Update Briefing and After Action Review will present student performance across all six performance categories. the major categories of policies that students use. For those concepts on which the student's performance is good (that is one of the top four categories of performance), the principle is cited, with an example in which the student followed the principle for each of the four categories, and references to Field Manuals that support the principles. For the two categories on which the student performed poorest, the principle would be cited, with examples of the student following the principle (if one can be found), and an example of the student violating the principle, with suggestions on how the trainee should have performed.

3 Results and Discussion

For UrbanSim, the PEER process resulted in six major categories of policies, and 18 sub-scores. We are implementing the instructional remediation with the Generalized Intelligent Framework for Tutoring (GIFT); when completed, we will conduct studies of this instructional design.

The innovation of our approach is the effort to build affordable ITS based on review by expert judgments of performance by real students. This approach produces two primary efficiency benefits for assessment: (1) We focus building scoring rules around the characteristics of performance taken by real students rather than taken by any possible student; while the scoring system also addresses theoretical aspects of poor performance, it spends the majority of effort investigating observed difficulties of students. And (2) we focus on performance rather than diagnosing underlying cognitive capabilities that are inferred from performance. We believe this is instructionally defensible. We will know more after instructional studies are taken.

The instructional benefit of this approach is that much of the content was built based on collecting critiques of student performance. This resulted in a first draft of the instruction. It was revised by experts, and references to official policy were added.

4 References

1. Lesgold, A., Nahemow, M. Tools to assist learning by doing: Achieving and assessing efficient technology for learning. Cognition and instruction: Twenty-five years of progress, 307-346 (2001).
2. Pokorny, B., Haynes, J., Gott, S. Performance Assessment in Complex Environments. In: Interservice/Interagency Training Simulation and Education Conference. Orlando, FL: NTSA (2010).
3. Pokorny, R., Hall, E., Gallaway, M. Analyzing Components of Work Samples to Evaluate Performance. Military Psychology, 161-177 (1996).
4. Woolf, B. Building Intelligent, Interactive Tutors" Student-Centered Strategies for Revolutionizing e-Learning. Burlington, MA: Morgan Kaufmann (2009).
5. Sack, W., Soloway, E., Weingrad, P. Re-Writing Cartesian Student Models. In J. &. Greer, Student Modeling:The Key to Individualizing Knowledge-Based Instruction (pp. 355-376). Berlin: Springer-Verlag (1994).
6. Collins, A., Greeno, J. A Situative View of Learning. In *International Encyclopedia of Education.* London: Elsevier (2010).
7. Wray, R., Woods, A. A Cognitive Systems Approach to Tailoring Learner Practice. *Advances in Cognitive Systems* (pp. 21-38). Baltimore, MD: Cognitive Systems Foundation (2013).
8. Chi, M. Active-Constructive-Interactive: A Conceptual Framework for Differentiating Learning Activities. Topics in Cognitive Science, 73-105 (2009).
9. De Jong, T. Cognitive Load Theory, Educational Research, and instructional Design: Some Food for Thought. Instructionmal Scneice, 105-134 (2009).
10. Vogt, B. Two Methodologies to Assess UrbanSim Scenarios. Proceedings of Intervice/Industry Teaching, Simulation, and Education Conference. Orlando, FL: National Training Systesm Association (2013).

How Can the Affective Model be for Tutors on Moodle?

Júlia Marques Carvalho da Silva

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
Câmpus Bento Gonçalves Av. Osvaldo Aranha, 540, Bento Gonçalves, RS, Brazil
julia.silva@bento.ifrs.edu.br

Abstract. Innovations in computer science have presented a lot of changes, especially in the field of education and more specifically in e-learning. Affective Computing has also emerged recently. Research is being done in this area, but it does not cover an important role in this process: the tutor. This research aims to study the tutor, more specifically the emotional factors that can be identified and how they can improve the teaching and influence in the learning process. We developed a module in the Moodle LMS, in order to help the tutor improve his/her actions and the relationship among the students.

Keywords: affective model, tutor, virtual learning environments, Moodle.

1 Introduction

Nowadays technology enhanced learning is becoming popular especially through Learning Management Systems (LMS). By LMS, students can access learning content and resolve activities. Teachers can consult student's performance and promote changes when necessary.

LMS are increasing in the last few years. They are becoming smarts and sensitive to feelings. They are able to adapt for learning needs considering cognitive and affective aspects of students [7, 9, 10]. However, there are few efforts to help teachers.

Intelligent Teaching Assistants systems (ITAs) are Intelligent Tutoring Systems (ITS), which aim to assist students and teachers. They provide assistants on teachers' tasks, although there are no systems that consider teacher affective model. Emotions are being related as essential in development to any activity. Students learn less if they are anxious, angry or depressive [4]; but how about teachers? They can better teach with positive affective states? Sutton e Wheatley [18] affirm that emotions can influence teacher's cognition and motivation and, consequently, students are aware to teachers' feelings.

Recent studies show that teacher's role in LMS and ITS seem to be forgotten [18]. But it is possible to find studies that describe the importance of teacher affective aspects [6, 18]. Cunha et al. [6] describe an empirical model based on ordinary teacher's behavior to identify affective states.

This paper demonstrates how Cunha [6] model can be applied into Moodle LMS. We verified each affective state and how to develop it in Moodle. Finally, we tested the model using controlled and real environment.

2 Related Work

Our study is based on the bridge among: Learning Management Systems (LMS), emotions and teacher profile. Therefore, we need to comprehend how these components are made and covered. So, we describe LMSs and their evolution stating form adding intelligent and affective components for both students and teachers.

LMS are information systems that offer learning objects, assignments and communication tools. Teachers and students use those resources in courses. When LMS is available through the internet, it is possible to offer online courses. Moodle [12] is an example of LMS, which is free and open source. Moodle also allows customizing according to developers or the school's needs.

Intelligent Tutoring Systems (ITS) are a type of LMS that include intelligent resources. Consequently, they allow students to expand their learning possibilities. ITS architecture includes student model, domain model and pedagogical model [21].

After, ITS presented some evolution, including easiness to teachers and affective model to students. Intelligent Teaching Assistants systems (ITAs) consist on ITS that add teacher's model and interface. This model has assistants to help (automating or guiding) teachers' tasks [21]. However, teachers remain in control of the activities and pedagogical decisions. When teachers are helped, students are benefited, because teacher can spend more time doing the mediation [6].

Recently, it is noted the need of considering cognitive and affective aspects of students to better provide personaled learning [9]. Affective Tutoring Systems (ATS) are ITS that include an affective student model [1]. This model can detect frustration or stress, simulate agents with affective states, monitor social interaction, diagnose motivation, and then to adapt the system for each student [14]. Therefore, ATS adapts as well as a human teacher does [2]. And, consequently, the student will feel more pleasant through the learning process [1].

It's also possible to find ITS that implements both ITA and ATS. Alice is one example. It considers affective aspects of students and provides intelligent assistants that help teachers to verify plagiarism and how correct an answer is [17].

However, despite these studies that describe how important is to consider emotions in learning process, there are only a few about emotion in the teaching process. About teacher, we know how important it is to have a lower workload, and one way is proving automatic tools. But, there are no studies about teachers and theirs emotions in order to improve teaching. This is also criticized by Tretiakov et al. [19], who affirm that ITS has failed to recognize the real role of teacher.

3 Motivation

According to Carvalho [5] and Tretiakov et al. [19], the teacher should be able to guide the learning, to motivate student, to know technological tools, to be aware to student's context, to select and to organize the content, to manage the curriculum, to observe learning progress, and to be open to judgements. However, affective skills influence more than cognitive ones [8]. Yacef [21] affirm that to help teachers to better teaching is an activity as important as teaching students. When teachers are able to recognize what they are feeling, they can better express themselves in classroom [3]. So, a teacher guided affectively (respecting the course content and class scenario) has better condition to develop the curriculum and get results more effective.

We believe that join Affective Computing with ITA architecture can improve affective tutor profile. ITA offers a set of functionalities designed to teachers. It allows to work better and quickly. Consequently, teachers take care more on how student is learning. This mediation needs affective skills to affect student positivity.

4 Our Proposal

The study goal is to describe how to include an affective model to teachers using Moodle. Although it is just a LMS, not a ITS, we choose it because Moodle allows customizing. Nowadays there are studies towards to expand Moodle as ITS. It allows creating pedagogical rules to provide personalized learning [9]. For the teachers, Moodle offers tools to help them in ordinary tasks, without smart or affective features.

The second definition was about the affective model oriented to teachers. There are many ways to identify emotions, using external clothes or equipment as sensors, or making assumptions through behavioral models [2]. We choose the same technology used by LMS that has affective model for students, which assumptions are made by their interactions [9].

We also choose the model proposed by Cunha et al. [6], whose we can predict six affective states from teachers' interaction. This is an empirical model based on previous studies made by the authors. This is not a surprise for us, as the most of studies are recent, which technologies and proposals are new [20].

4.1 Affective Model

Cunha *et al.* [6] model aims to identify affective states of teachers whose use LMS. Authors believe that teacher must present communicability and sociability skills to better talk and give attention for all students. Punctuality and commitment to tasks' deadlines. Meticulousness to be attend to all events, for example, a new post in forum. And, initiative, to provide new or alternative tasks and contents. They also describe how each affective state can be measured using variables present in most of LMS:

- Sociability: it is the teacher's capacity to communicate with all students in the same way (homogenous). It is calculated by the standard deviation of the number of messages sent to students. The higher the value found, the greater the chances of the teacher to be paying more attention to a student or to be paying less attention to a particular student than the other.
- Communicability: it is measured by the amount of message size per number of messages. It depend on the tool used. The chat messages are shorter and direct than forum posts. It is also evaluated messages sent as student's feedback in assignments and quizzes. Authors explain that too short or too long messages may mean

bad communicability. We must notice that is not measured messages' quality here, because it depends on each context.

- Punctuality: from what was agreed with the class, the teacher's point of care interactions of students, whether it be a question via discussion forum, sending a job, a general doubt about the course or feedback to an answer to an exercise. To determine the timeliness, we used the date of delivery of the task and the date of the teacher's response and the date of posting to a forum and the date of the student teacher's response to this forum. The difference of these dates is computed, so that later generate an average response time teacher.
- Commitment: it refers to the commitment of the teacher to meet the criteria previously established and agreement. The commitment is based on the difference of the final delivery date of the task and the date of assessment and teacher response.
- Meticulousness: it refers to the ability of the teacher to pay attention not only on the interactions of the students in the virtual environment, but also to maintain the perception and the solution of the consequences of these interactions. The meticulousness is calculated by the date and time of the last visit of the teacher to the forum. The difference of this date with the current date is the estimated time in which the teacher has not accessed the system.
- Initiative: it refers to the ability of the teacher to support the student in new actions in the virtual environment. We calculated how many weeks have passed since the beginning of the course and how many materials were placed by the teacher from the beginning of the course.

4.2 Development

We verified how Moodle organize each information in its database. For each affective state, we built SQL queries to get the values and then calculate. After, we established values to range: very good, good, regular, bad, very bad. Those values were gotten considering the university where the model was applied.

Moodle offers specific customization points, for example, reports, boxes, resources, questions types, etc. We choose box option, because it can be added as teacher wanting. Also, box is just visible to teachers, students are not able to see. Figure 1 presents the box (only in Portuguese).

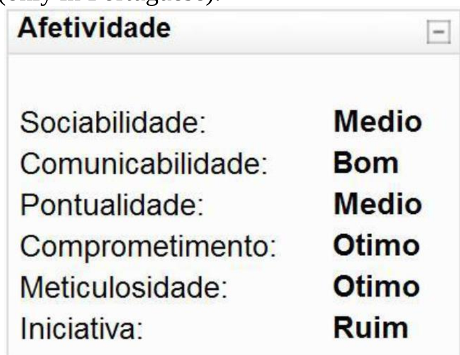


Fig. 9. Box implemented into Moodle.

Nowadays, the box shows each affective state and its value. Although, we know this information is not enough, because teacher may not comprehend the meaning of each one and how to improve it. So, it is important to give better messages, based on texts that guide teachers in their tasks. Those messages can be determined from affective states and can stimulates teachers' upgrade.

4.3 Tests

As a new and empirical model, there are no previous studies proofing its accuracy. We did some tests to check if teachers' actions really change affective states. We also double check if students have the same perception about teacher as model presents.

In the first test, we did some simulations using a controlled environment. We created teachers and did some actions to verify how each affective states oscillate. The results were satisfactory and consistent to model proposed.

The second test involved real data. We got data from 4 undergraduate courses that use Moodle LMS to support face-to-face and online activities. Students of each course received a survey with sentences like: "I receive activities feedback quickly". For each sentence, students have 5 answers options based on Likert: always, often, sometimes, rarely, never. We received 88 surveys from students.

The results show that communicability average indicated by students are above than model indicated. Commitment and initiative have the same values from students and model, no matter whether are positive of negative. Meticulousness was the affective state with worst accuracy, because model has indicated values too below than compared to students' opinion. Finally, we cannot measure sociability and punctuality because there were not information enough in Moodle LMS.

With these results, we can see a relationship between students' opinions and values presented in the equations. It is understood that full compatibility would not be achieved, since the surveys reflect the views of students, which is susceptible to some variables of the educational process.

5 Conclusions

Nowadays, with the news technologies, the possibilities of its use in education have been increasing. This situation enabled a revolution and new teaching methods. Several studies involving affectivity in educational processes have been developed in order to analyze the real impact of this aspect during this activity, but few of them work with the tutor itself, seeking to verify your affective aspect. Yacef [6] reinforces the importance that the teacher has on learning and propose new systems that make the teacher with a more important role in this process.

For this study, we chose to use an affective model already defined, requiring only the construction of equations that simulate the oscillations of the values according to the actions of the tutor. These equations were implemented in Moodle through SQL queries and then presented visually in the form of box. To validate the information, a survey was developed in which students informed their perceptions for each of the affective attributes.

The results are promising and open new studies: how can affective states benefit teacher as well as students? How to guide teachers to better work, without step in their pedagogical decisions? Can those orientations really benefit educational process? Is there another affective states that should be added to describe teachers?

This study limitation is concerned in frailty model. Especially if model is applied in a course with more than one teacher. We don't know how it will work.

We hope that this work will contribute the quality of teaching. We believe that with the inclusion of the box, the teacher can regulate themselves, discovering how to improve on their teaching process.

6 References

1. Alexander, S.; Sarrafzadeh, A.; Fan, C. Pay attention! The computer is watching: Affective tutoring systems. In: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education. (2003) p. 1463-1466.
2. Alexander, S.; Sarrafzadeh, A.; Hill, S. Easy with eve: A functional affective tutoring system. In: Workshop on Motivational and Affective Issues in ITS. 8th International Conference on ITS (2006) p. 5-12.
3. Brackett, M. A.; Katulak, N. A. Emotional intelligence in the classroom: Skill-based training for teachers and students. Applying emotional intelligence: A practitioner's guide, p. 1-27 (2006)
4. Burleson, W.; Picard, R. Affective agents: Sustaining motivation to learn through failure and a state of stuck. In: Workshop on Social and Emotional Intelligence in Learning Environments. (2004)
5. Carvalho, A. B. Os Múltiplos Papéis do Professor em Educação a Distância: Uma Abordagem Centrada na Aprendizagem. In: 18° Encontro de Pesquisa Educacional do Norte e Nordeste – EPENN. Maceió (2007).
6. Cunha, C. R.; Silva, J. M. C.; Bercht, M. Proposta de um Modelo de Atributos para o Aprimoramento da Comunicação Afetiva para Professores que atuam na Educação a Distância. In: Anais do Simpósio Brasileiro de Informática na Educação (2008) p. 573-582.
7. D'Mello, S. K et al. Integrating affect sensors in an intelligent tutoring system. In: Affective Interactions: The Computer in the Affective Loop Workshop at. (2005) p. 7-13.
8. Goleman, Daniel. Emotional intelligence. Random House LLC (2006)
9. Khan, F. A. et al. Implementation of affective states and learning styles tactics in web-based learning management systems. In: Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on. IEEE (2010) p. 734-735.
10. Kort, B.; Reilly, R.; Picard, R. An affective model of interplay between emotions and learning. In: Proceedings of IEEE International Conference on Advanced Learning Technologies (2004) p. 43-46.
11. Litman, Diane J.; Forbes-Riley, Kate. Recognizing student emotions and attitudes on the basis of utterances in spoken tutoring dialogues with both human and computer tutors. Speech communication, v. 48, n. 5, (2006) p. 559-590.

12. Longhi, M. T.; Behar, P. A.; Bercht, M. O Desafio de Reconhecer a Dimensão Afetiva em Ambientes Virtuais de Aprendizagem. In: Anais do Simpósio Brasileiro de Informática na Educação (2008) p. 471-480.
13. Moodle. http://www.moodle.org (2014)
14. Robison, J.; McQuiggan, S.; Lester, J. Evaluating the consequences of affective feedback in intelligent tutoring systems. In: Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on. IEEE (2009) p. 1-6.
15. Shen, Liping; Wang, Minjuan; Shen, Ruimin. Affective e-Learning: Using" Emotional" Data to Improve Learning in Pervasive Learning Environment.Journal of Educational Technology & Society, v. 12, n. 2 (2009)
16. Sidney, K. Dmello et al. Integrating affect sensors in an intelligent tutoring system. In: Affective Interactions: The Computer in the Affective Loop Workshop at (2005) p. 7-13.
17. Silva, J. M. C.; Raabe, A. L. A. Including Affective Student Model in ITS to Teaching Introductory Programming. In: Workshop Emotional and Cognitive Issues in ITS, Montreal. Workshop Emotional and Cognitive Issues in ITS (2008)
18. Sutton, Rosemary E.; Wheatley, Karl F. Teachers' emotions and teaching: A review of the literature and directions for future research. Educational Psychology Review, v. 15, n. 4, p. 327-358 (2003)
19. Tretiakov, Alexei et al. Human teacher in intelligent tutoring system: a forgotten entity. In: Advanced Learning Technologies. Proceedings. IEEE International Conference on. IEEE (2001) p. 227-230.
20. Woolf, Beverly et al. Affect-aware tutors: recognising and responding to student affect. International Journal of Learning Technology, v. 4, n. 3, p. 129-164 (2009)
21. Yacef, Kalina. Intelligent teaching assistant systems. In: Computers in Education, 2002. Proceedings. International Conference on. IEEE (2002) p. 136-140.

The Unique Elements to Consider In Team Tutoring

Heather Holden[1], C. Shawn Burke[2], Christopher W. Wiese[2]

[1]Army Research Laboratory, Orlando, Florida
Human Research and Engineering Directorate
Heather.k.holden@us.army.mil
[2]University of Central Florida, Orlando, Florida
Institute of Simulation and Training
{sburke, cwiese}@ist.ucf.edu

Abstract. The purpose of this paper is to expose the differences between the elements required for individual learner models to accurately assess an individual's learner state and the elements of team models to accurately assess a team's learner state. A literature review investigating the science behind teamwork and team performance pertaining to the principles of intelligent tutoring systems was conducted. The initial results are presented in this paper.

1 Introduction

Intelligent tutoring systems have historically focused on providing individualized adaptive learning for a single learner. Collaborative learning, on the other hand, has been found to produce great benefits such as, increasing social interaction and interpersonal relationships, improving students' time on task and motivation to learn, and increasing learners' expectations for personal success [1]. With this in mind, there has been a strong motivation to conduct work incorporating collaborative learning practices with intelligent tutoring systems (ITSs) to develop team tutoring systems. However, ITSs for individual learning already exhibit many challenges in terms of its learner modeling capabilities; therefore, the development of successful team ITSs will significantly increase the complexities of this challenge area.

The purpose of this paper is to expose the differences between the elements required for individual learner models to accurately assess an individual's learner state and the elements of team models to accurately assess a team's learner state. While previous team state models have been theorized [2], current work focuses on developing design architectures, inclusive of behavioral markers and metrics, for each of these models. The design architectures will be rooted in principles of intelligent tutoring and the science behind teamwork and team performance. The first step of the current work is to conduct a thorough literature review investigating the science behind teamwork and team performance as well as the principles of intelligent tutoring. This synthesis of the literature on teams includes both conceptual and empirical articles. Inclusion criteria is as follows: (1) search period 2003-2013, (2) sources include peer reviewed journals and conference proceeding, (3) databases searched include PsychInfo, DTIC, and ProQuest, (4) use of snowball approaches whereby the reference lists of identified articles such as meta-analyses, major reviews, other articles that are found in the initial search are also reviewed for additional sources, and (5) disciplines searched include psychology, healthcare, military, organizational behavior, and sports. Search terms included, but were not limited to: ‘teams and learning’, ‘teams and satisfaction’, ‘teams and viability’, ‘teams and performance’. This search yielded approximately 6,000 articles. After cross-referencing the articles from these search terms with the previous ones to avoid coding duplicates there are approximately 5,991 unique articles to code. Progression through this systematic review process will help to ensure that the design architectures to be developed are scientifically-rooted in the literature.

2 Elements of a Comprehensive Learner Model

The learner model, a core module of ITSs, is the representation of learner’s current state of knowledge at any given time [3]. A comprehensive model would include information on the learner’s individual difference characteristics, his/her past and current competency, performance, cognition, affect, behaviors, etc. The ITS uses such information to adapt and customize instruction accordingly based on the learn-er’s state.

The content within learner models, as shown in Table 1, is generally categorized in two parts: *domain-specific* or *domain-independent* information [i.e. learner-specific characteristics (individual differences)] [4, 5]. Domain-specific information represents a reflection of the learner’s state and level of knowledge or ability within a particular domain. Most learner models, particularly those of first generation ITSs, are concerned with modeling this type of information because this allows the model to be more generalized across multiple populations. While this information is useful, it alone is not sufficient for providing the highly adaptive individualized training. Domain-independent information consists of all relevant characteristics of an individual learner. These individual difference variables are significantly different between learners and, collectively, are not the same for any two learners.

Table 3. Learner Model Content

Domain-Specific Information	Domain-Independent Information
Represents a reflection of the learner's state and level of knowledge or ability within a specific domain.	Consists of all relevant characteristics of an individual learner.
Data Includes: • Historical Competency (i.e., domain knowledge and skills measured over time) • Misconceptions • Problem-Solving Strategies • …	***Data Includes:*** • Learning Goals • Cognitive Aptitudes • Measures of Motivational State • Learning Preferences (including styles and personality) • Interest • Demographics • Past Performance and Competency (domain-independent) • Behavioral/Psychological Measures • Cognitive and Affective Dimensions • Personal Control Beliefs • ….

2.1 Learner States As The Source of Adaptation

Current learner modeling research focuses on understanding the interrelationship between the domain-independent information and how it can best be used with the domain-specific information for accurately classify learner cognitive and affective states [6]. According the adaptive tutoring learning effect chain, as shown in Figure 1, cognitive and affective learner state models inform the selection of optimal instructional strategies to support higher learning gains [7]. There are also other learner states (i.e., motivational, behavioral, etc.) that may be important to monitor during this process, but cognitive and affective states are the most significant during learning.

Fig. 10. Adaptive Tutoring Learning Effect Chain for Individual Tutoring [7]

3 Elements to Consider for Modeling Teams

When it comes team ITS, there is an added workload of coordinating states of individual team members so a more comprehensive picture of the team state can be developed. More specifically, the ITS needs to understand the state (e.g., cognitive, affective, motivational, psychomotor/behavioral) of each team member, their individual performance, the communication and interactions of the team members, the contributions of each individual's performance and state and interactions to the collective performance of the team. Moreover, inputs to the "team state" might include the state of trust between individual team members, progress towards team goals, reassessment of team goals based on priorities, and the distribution of workload for each member [2]. Figure 2 presents a notional team tutoring learning effect chain.

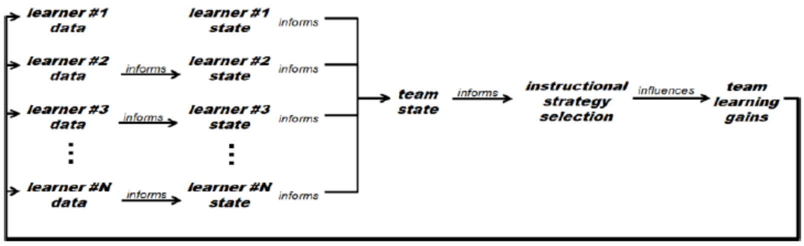


Fig. 11. Notional Adaptive Tutoring Learning Effect Chain for Team Tutoring [8]

Regardless of the number of individual learners to monitor on any given team, the team state is more complex than an aggregate collection of the learner's individual states. Therefore, accurately assessing the team state is the first main challenge of team tutoring. One of the first steps in ensuring the accurate assessment is to determine the team outcomes that one is interested in training as this will drive the content that should be incorporated into each of the team state models as well as provide insight into data aggregation methods.

3.1 Important Outcome Variables for Team Tutoring

In determining the content that should be included in the team state models one of the first steps is to decide upon the types of outcomes that the team tutor might choose to focus upon. Guided by conceptual models within the teams literature (for a review see [9]), we initially identified four team outcomes to focus upon: team performance, team learning, team satisfaction, and team viability. Team performance was characterized as a judgment of how well the results of teamwork meet some set of standards (objective or subjective). Team performance is the outcome that has received the most attention in the team's literature as it reflects how well teams are able to enact team processes and states to achieve a desired team goal. A review of the articles that examine team performance as an outcome include evidence for a wide variety of antecedents, some of the most common include: communication, coordination, mutual support, reflexivity, monitoring, conflict (task, relationship), leadership, interpersonal processes, conflict management, organizational citizenship behaviors, trust, collective efficacy, psychological safety, cohesion, team mental models, transactive memory systems, and situation awareness.

Team learning was also identified as an important outcome for the team tutor to focus on for several reasons. First, the ability to facilitate learning would seem to be at the heart of a tutor (team or otherwise). Second, it has been argued that team learn-ing is important in that a lack of team learning precludes a team's ability to be adap-tive – something that is fundamental for success in complex environments such as the military. In examining team learning as an outcome of interest we are referring to the acquisition or refinement of task-related knowledge or skills through interaction with one another [10]. While our meta-analytic and conceptual analysis of the literature revealed fewer articles that focused on this outcome a fair number were still uncovered. A review of these articles revealed several trends, including evidence for communication, collaborative learning, coaching/leadership, and psychological safety as being positively related to team learning. Meta-analytic evidence was found for communication, coordination, reflexivity, conflict, conflict management, trust, psychological safety, and cohesion being significantly related to team learning.

Team satisfaction is the third team outcome that is often identified within prominent team models and frameworks as being important to consider, especially if the team will work together for an extended time or under varying levels of stress. The degree to which members are satisfied with the team interaction can serve as a motivational driver for members. In this vein, team satisfaction can be defined as the degree to which members enjoyed being a member of the team. A review of articles examining team satisfaction as an outcome revealed several trends, including evidence for conflict, cohesion, team potency, and team trust in leadership, and trust in each other as antecedents of satisfaction. Meta-analytic evidence was also found for: communication, coordination, mutual support, reflexivity, conflict, transition processes, action processes, interpersonal processes, leadership, conflict management, trust, collective efficacy, psychological safety, cohesion, transactive memory systems. Another trend suggested by the literature is that team satisfaction is linked to overall team effectiveness.

Team viability, while appearing less often than team performance and team satisfaction, has also been argued to be an important team outcome, especially for those teams whose members are expected to have to work together in the future. Team viability has been defined as the desire to remain a part of the same team for future performance episodes [11]. A review articles focusing on this team outcome revealed several trends, including evidence for conflict, goal commitment, and team mental models as antecedents to viability. Meta-analytic evidence was also found for the following team constructs explaining significant variance in team viability: communication, coordination, mutual support, conflict, conflict management, collective efficacy, psychological safety, team cohesion, and transactive memory systems. Another trend suggested by the literature is that team viability is linked to overall team effectiveness.

3.2 Team State Taxonomy

Now that targeted team outcomes have been identified, we can begin to think about those constructs (e.g., attitudes, behaviors, cognitions) which serve to facilitate such outcomes. In this vein, [2] began to delineate a series of six team state models (e.g., team performance, team competency, team cognitive, team affective, team trust, team communication) which when incorporated into the GIFT framework would serve to guide the learner assessment (see Figure 3, left hand side). While these models provide an initial starting point the set needs to be verified through conceptual and empirical evidence as well as being expanded upon such that the subcomponents of each are apparent. It is these subcomponents that behavioral markers will be built around to facilitate the assessment and adaptive tutoring portions of GIFT as related to teams. Based on the results of the conceptual and empirical review, we compared our findings to the original categories proposed by [2]. The results of which appear on the right hand side in Figure 3.

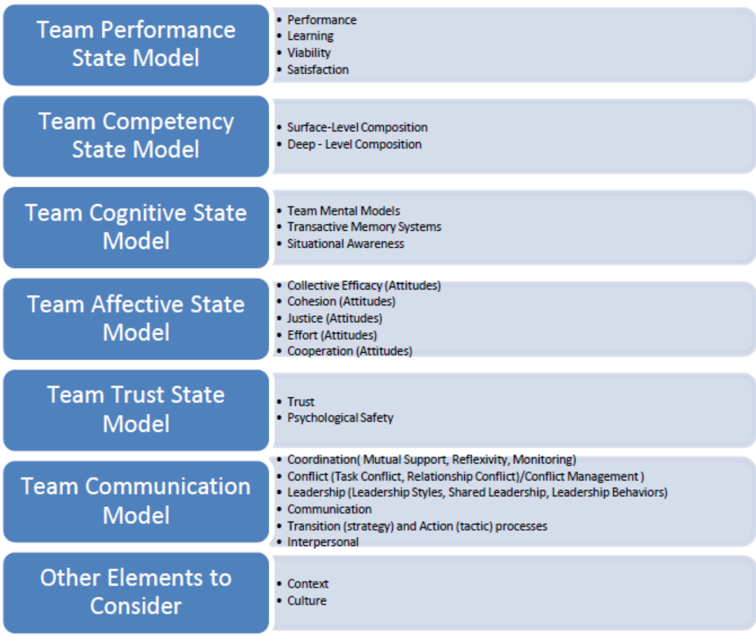


Fig. 12. Team State Models

Of note is that at an initial glance the high level categories seem to align with those team aspects which have been shown to be most predictive and focused upon within the teams' literature. In unpacking the team state models on the left, we expand the team performance state model to include the four team outcomes (performance, learning, viability, and satisfaction) to provide a fuller picture of what constitutes effective team outcomes. This expanded model reflects quantitative or qualitative evaluations of the status of team outcomes during any portion of the team's performance period. Current discussions are ongoing in terms of refining the name of this state model to more accurately reflect the content contained within.

With respect to the team competency state model here we include those individual team member characteristics which have the potential to impact or influence the accomplishment of team goals. These characteristics may include, but are not limited to knowledge, skills, ability, and experience. Some of the competencies reflect surface level characteristics (i.e., immediately observable) while others reflect those which are only apparent after interaction. The specific constructs which underlie each of these categories are currently being unpacked.

The third state model, team cognitive, reflects an evaluation of the shared cognitive state of all team members. This can include the degree to which members (a) structure knowledge in a similar manner, (b) understand the roles, expertise, and expectations of fellow members, and (c) have a shared impression of aspects of the team's status. Exemplar constructs here are shared mental models, transactive memory systems, and situation awareness.

The fourth state model, team affective, refers to those constructs which describe the general feelings that team members have towards one another during team interaction. This can include but is not limited to feelings regarding the team's ability to accomplish their goals and the team's emotional sentiments towards one another. Exemplar constructs here include, collective efficacy, cohesion, justice, effort, and cooperation.

The fifth state model, team trust, reflects the shared belief that all team members will fulfill their role responsibilities, perform delegated tasks, and not attack fellow team members for expressing their opinion. Exemplar constructs within this model include trust and psychological safety.

The final state model that was originally identified by [2] was the team communication model. This model reflects observable behaviors between group members which either directly impacts task progression/completion or indirectly facilitates the synchronization between team members. As such, it is broader than mere communication and might be more closely aligned with a 'team interaction model'. Exemplar constructs include: coordination (mutual support, reflexivity, monitoring), communication, conflict, conflict management, leadership, organizational citizenship behaviors, interpersonal processes, as well as the more recent work on action/transition processes.

Lastly, our literature search reveals two factors that could not be easily integrated into the state models presented in [2]. However, they are obviously important considering their prominence in the literature. Currently, work is in progress to determine whether these two factors could fold into one of the pre-existing state models or if they are distinct enough to require the creation of a new state model.

4 4 Conclusions and Future Work

This paper has only begun to highlight the complexities in moving from an individual learner model to a team-based learner model. Specifically, we have predominantly begun to highlight the differences in the type of content that may be included in each of these types of models. There are many other complexities in making this transition that will be covered in forthcoming papers. For example, the many of the identified constructs differ in terms of their nature – task generic, task specific, team generic, team specific [9, 12]. These distinctions speak to the degree of generalizability of their importance across tasks and teams. Another aspect of complexity lies in the assessment of such constructs. Within the teams literature the predominant form of assessment has been self-report Likert type measures. Within an intelligent tutoring framework the desire would be to identify behavioral markers of each construct that can be automatically captured by the system. Finally, there are also challenges in terms of how to aggregate the data to arrive at a team index [13]. All of this to highlight to fact that the transition from an tutor designed to train individual skills to one focusing on team skills is not an endeavor easily undertaken and one in which challenges are tackled in an iterative manner.

This paper has identified the primary elements that belong in the team state models. The next step of this effort is to develop the behavioral markers and metrics to assess these elements; however, how to measure and develop these markers and metrics is a great challenge. The need to have these behavioral markers and metrics assess team state in a domain-independent fashion produces even greater complexity. Ultimately, these models will be incorporated in the Generalized Intelligent Framework for Tutoring (GIFT), a domain-independent framework that can be used to generate ITSs and conduct ITS research.

5 References

1. Johnson, D. and R. Johnson, An Educational Psychology Success Story; Social Interdependence Theory and Cooperative Learning. Educational Researcher, 2009. **38**(5): p. 365-379.
2. Sottilare, R., et al., Challenges and Emerging Concepts in the Development of Adaptive, Computer-Based Tutoring Systems for Team Training, in International/Industry Training, Simulation, and Education Conference (I/ITSEC 2011)2011: Orlando, FL.
3. Kassim, A., S. Kazi, and S. Ranganath, *A Web-based Intelligent Learning Environment for Digital Systems.* International Journal of Engineering Education, 2004. **20**(1): p. 13-23.
4. Abdullah, S., *Student Modelling by Adaptive Testing - A Knowledge-based Approach*, 2003, The University of Kent at Canterbury (Dissertation).
5. Gonzalez, C., J. Burguillo, and M. Llamas, A Qualitative Comparison of Techniques for Student Modeling in Intelligent Tutoring Systems, in 36th ASEE/IEEE Frontiers in Education Conference2006, IEEE: San Diego, CA.
6. Holden, H., et al., Effective Learner Modeling for Computer-Based Tutoring of Cognitive and Affective Tasks, in Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)2012: Orlando, FL.
7. Sottilare, R., et al. Considerations in the development of an ontology for a Generalized Intelligent Framework for Tutoring. in International Defense & Homeland Security Simulation Workshop in Proceedings of the I3M Conference. 2012. Vienna, Austria.
8. Fletcher, D. and R. Sottilare, Shared Mental Models of Cognition for Intelligent Tutoring of Teams, in Design Recommendations for Intelliigent Tutoring Systems -- Learner Modeling, R. Sottilare, et al., Editors. 2013. p. 239-254.
9. Salas, E., et al., The Wisdom of Collectives in Organizations: An Update of Team Competencies, in Team Effectiveness in Complex Oranganizations2007. p. 39-79.
10. Argote, L., D. Gruenfeld, and C. Naquin, *Group Learning in Organizations*, in *Groups at work: Advances in theory and research*, M.E. Turner, Editor 1999, Erlbaum: New York.
11. Hackman, J.R., *The Design of Work Teams*, in *Handbook of Organization Behavior*, J.W. Lorsch, Editor 1987, Prentice Hall: Englewood Cliffs, NJ.
12. Cannon-Bowers, J.A. and E. Salas, *A Framework for Developing Team Performance Measures in Training*1997, Mahwah, NJ: Lawrence Erlbaum.
13. Klein, K.J. and S.W.J. Kozlowski, Multilevel Theory, Research, and Methods in Organizations: Foundations, Extensions, and New Directions2000, San Francisco, CA: Jossey-Bass.

A System Dynamics Approach to Building Team Trust Models: Exploring the Challenges

Mostafa Amin-Naseri, Stephen Gilbert

Department of Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, IA
{aminnas,gilbert}@ iastate.edu

Abstract. Learner models are one of the most important parts of any tutoring system. Due to the complexity of social systems, it gets more challenging to track personal data and to build a model of learner's state when dealing with teams. This research suggests leveraging the available literature on team dynamics to make a system dynamics model of teaming. This model will offer a more accurate representation of the complexity involved. An example system dynamics model of team trust is created based on a previous qualitative study of team trust [3]. Its benefits include a holistic understanding of trust structure in teams, the ability to evaluate and predict trust level in teams given current individual states, and providing a testbed to evaluate multiple remedies to team issues. The authors suggest that using this system dynamics (SD) modeling approach with GIFT as the individual learner model is a valuable initial approach to adding full team functionality to GIFT.

1 Introduction

Sottilare et al. emphasize the important role of learner models in understanding the learner's state in individual learning [1]. They expand this to team tutoring, where the input becomes relationships and states between individuals in teams. Fletcher and Sottilare also introduced the importance of team shared mental models and the difficulty of measuring traits like team trust, affect, or shared mental models [2]. Due to the complexity of individual differences and team member interactions, the team trust or shared mental model is not as simple as the sum of all individual states. For example, based on Wilson's study [3], if considering the team trust level as the sum of individual trust, the issue of having cliques or subgroups in the team is not considered. While subgroups are forming, certain individuals may build increasingly high trust and communication with each other, and be not much connected with others. In this case, the sum of the individual trust level may be increasing, but in fact, having subgroups in the team will reduce the team identity and affect the overall trust level. In this paper we will explore a different approach in modeling team trust level and its challenges.

2 Challenges of Making a Team Trust Model

The current trend in learner models is making inferences from data using machine learning techniques [4]. As suggested by VanLehn, this can happen through empirical techniques using learning curves or Bayesian knowledge tracing models. The problem in team learner models is that the learning environment has many changing variables that are not easy to record. Also, the same group of people can act differently based on the complexity of the task domain and the role assignment [1]. Considering the amount of variability in such systems, using pure machine learning techniques to elicit generalizable rules would require a very large amount of observations as a training set. As team tutoring systems are relatively new, there aren't many available sources for data to build these models.
On the other hand, the dynamics of teams have been studied in different disciplines.

Many of these studies assess and describe the dynamics of team characteristics development such as trust in teams [3]. Also psychologists have delved into more detail and evaluated the effect of personal characteristics and emotions in forming trust in teams [5,6].

Comprehensive literature reviews on team dynamics [7,8,9] offer a larger picture of what we know about dynamics of teams. This understanding can serve as a basis for forming dynamic team models. When the literature is reviewed, the amount and complexity of influencing factors can get overwhelming to analyze at once. This is due to the complexity of such systems arising from the amount of interrelations and feedback loops. Also, some elements of the system may change with a time delay. Due to our cognitive capacity and human's mental models, analyzing such complex systems is almost impossible. It is necessary to have a method of presenting all the information and the relationships dynamically.

3 Possible Approach

System dynamics (SD) is a method that provides a holistic view of complex systems. It has been widely used in different disciplines, mostly in business and policy making. SD is a helpful method in unraveling the unexpected behaviors of complex systems. System dynamics models are developed to mitigate our limitations in analyzing the four main sources of complexity that we can't easily comprehend: dynamic complexity (due to the rapid changing environment), feedback loops (interrelations of elements) in a system, time delay (in reactions) and the effect of stuck and flow (effect of accumulation and dispersal of resources) [10]. The authors believe the complexity of trust or shared mental models in teams is no less than any other complex social system.

An SD model can be helpful initially to give a holistic understanding of the dynamics of trust in teams. Once mathematical models are added to it, we have a simulation model that represents the overall trust state in a team. This model is now ready for validation. Once validated, given the current state of individuals, the model can evaluate the team trust level and simulate the future trend of trust under various scenarios.

This means that while the tutoring system is gathering data on the individual states and trust between individuals, the SD model can serve as a team trust model. In addition, in cases where we have a trust issue in the team, there might be several possible ways to address that. With the simulation we can examine the results and select the most effective one. In the following section an example is given to demonstrate how a system dynamics model of trust in teams can be made based on analytic studies.

3.1 Previous Work

Similar approaches have been taken to model team dynamics and have shown the potential capacity of this method. Kefan et al. have used SD to model an entrepreneurial team's risk-based decision-making [11]. The model considers many environmental aspects and also makes a number of assumptions about the logical way of decision making of team members. Their model provides a basis to analyze team decision making. However, its model is not based on specific literature on team communication.

Kim et al. have introduced a team performance model named team crystallization that simulates team performance at a nuclear installation [12]. The model uses as inputs the number of team communications, the state of the power plant, and different control strategies. Using sophisticated mathematical models and leveraging neural networks, the model is able to simulate the team performance under various conditions. This model, however, doesn't study any other team elements than communication, and doesn't use feedback loops.

4 System Dynamics Team Trust Model Example

As an example, a study on trust in distributed teams [3] is used to make a simulation model using Vensim software. According to Wilson there are three key factors that contribute to trust in such teams: group identity, relationship and familiarity. While Wilson's results are valuable, questions remain, such as, "If we changed multiple team factors at the same time, what will happen to trust?" and "What are the exact relationships between these factors? Does one contribute 60% to trust while the others contribute the remaining 40%?" Lastly, this research doesn't address feedback loops, e.g., "If trust falls, does that impact team identity?" An SD model answers these questions. To construct an SD model, one needs both results like Wilson's to define the structure of the model, as well as case studies or expert opinion to assign the numbers. Once the model is made, the critical step of validation is required. The challenges of the validation process are discussed in section 5 of this article. An initial SD model based on Wilson's results is represented in Figure 1.

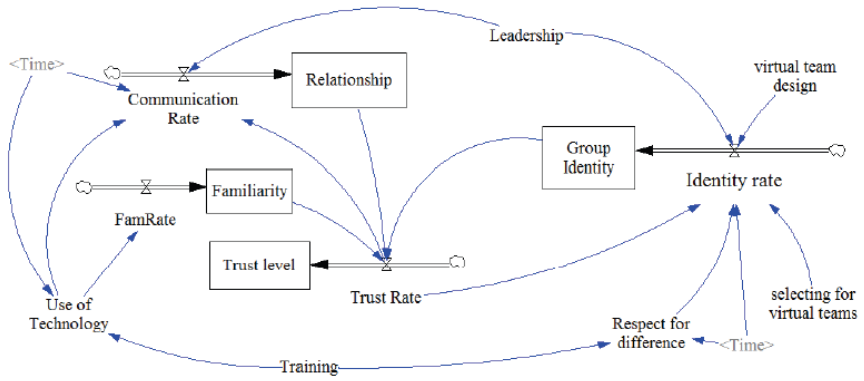


Fig. 13. SD Model of Trust in Teams, Based on Wilson, 2013

Each of Wilson's primary factors (shown as boxes) are affected by other variables that are represented in the model. The group identity, as explained by Wilson, is affected by the quality of team structure design, selection of team members and the level of respect the members give to differences. Also, role of strong leadership and training for diversity tolerance were mentioned as influencing elements in building team identity. The greatest threat to group identity was considered subgroups. The model represents all the mentioned relationships with the identity rate (arrow), which accumulates the group identity level over time, much like water filling a bucket. In this model low identity rate represents high subgrouping potential and vice versa. Feedback loops in social systems play an important role in explaining complex behaviors of a system. To include feedback loops in our model, we added the effect of trust rate in group identity rate. This means that once trust is falling, the group identity drops as well and vice versa. Also a feedback was added form trust rate to communication rate. Similarly the two other elements of the system, familiarity and relationship were modeled.

4.1 Simulating Team Scenarios

Assuming the model is validated, one of the benefits of an SD model is simulating different team scenarios. For example, what if the leader changed partway through a project? To show some outcomes of this particular model, we considered a scenario where the team design and member selection was not done carefully, and thus subgroups form. Also, the team leader is not successful in helping the team to build trust by enforcing communication or building team identity. The model is set to simulate a period of 100 time intervals which could represent 100 weeks. The trust level is a value between -5 and 5 starting at the initial state of 1. In this case the model shows the trend of team trust level. Imagine noticing this negative trend in week 10 and considering two options: either bring in a strong leader or have the members take a training intervention course. We test the effect of each and observe the results.
Figure 2 demonstrates the results under conditions of not doing anything, adding a strong leader in week 11, having the team take an effective training course in week

11, or doing both. In this case, doing both has the most influence. We can observe that in this model, the effect of a strong leader is higher than the effect of a training intervention. Having this SD model, many other scenarios can be tested as well.

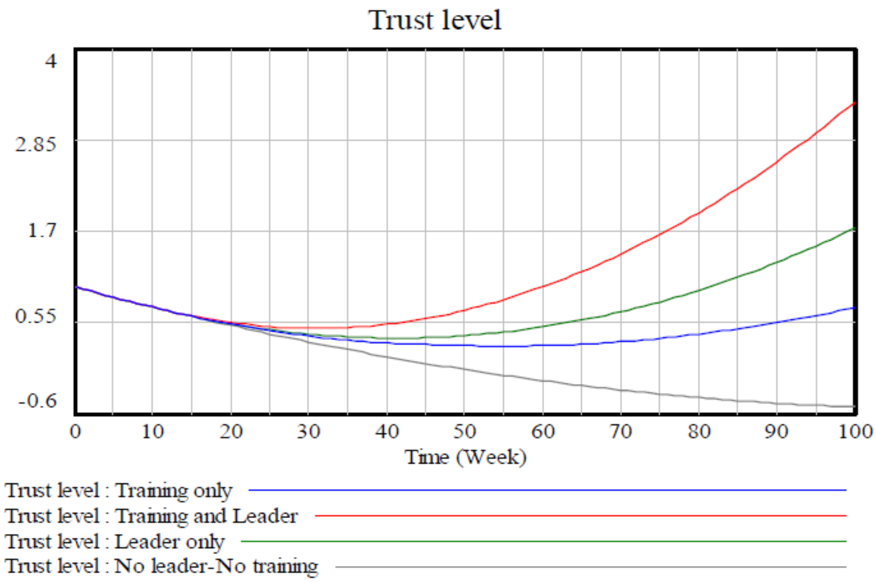


Fig. 14. Team Trust Level Under Different Scenarios

This model is solely based on one article, and thus may not be the best representative of trust in teams. Additionally, this model needs to be developed further to incorporate the dynamics of shaping subgroups based on individual behavior. However, the model illustrates the power of an SD approach.

When modeling abstract measures such as trust, we are more interested in the overall trends (rising or falling) in teams, rather than the actual values. By assigning some initial state numbers to the input values, we can test the model under various scenarios. Ideally the initial state numbers will come from the individual learner models. In order to validate the model, the weights need to be validated by case study data or in the early stages of model implementation.

5 Limitations and Challenges

Implementing a robust system dynamics model has some inherent challenges. First of all, every simulation model needs to be validated before the results can be considered dependable. The validation of the model requires actual data. Some research articles publish their collected data, and in this case, they can serve as a validation source. Otherwise, in the early stages of running the tutor, the team learner model needs to be refined and tuned until validated. In addition, in the literature, some research studies may not agree with each other. In that case we may need to have different models based on those ideas and validate them during the initial testing process.

6 GIFT Suggestions

The GIFT's sensor module has a very strong framework for inputting several streams of sensor data. Also, the built-in learner module of GIFT enables using the individual data to update the learner's affective state and learner model. However, for the team learner models, there needs to be a means to incorporate the SD model. Although these simulation models are easy to implement in simulation tools, it takes lots of effort to develop such a tool from scratch. Therefore, the authors suggest for the early stage of SD implementation, GIFT could facilitate an easy way to communicate with some existing SD software packages. However if using SD or other simulation models were proved to be helpful, then GIFT should consider incorporating such simulation as part of its learner model.

7 Conclusion

The structure of intangible and hard to assess features such as trust in teams is so complex that requires a holistic approach to understand and analyze. Using the analysis of teams in the literature and making a system dynamics model can first of all help the ITS team better in understanding the dynamics of the field. Secondly, the system dynamics simulation can construct a proper team learner model. Third, the model can serve as a laboratory to test several scenarios on teams and explore their behavior. Although validating such models may take some time, the validation process can happen in a shorter time than data-based models. Given GIFT's ability to collect all the sensor data through the sensor module, adding the ability to incorporate SD learner models or communicate with external SD sources will enhance team learner models.

8 References

1. Sottilare, R., Holden, H., Brawner, K., and B. Goldberg . (2011). Challenges and Emerging Concepts in the Development of Adaptive, Computer-Based Tutoring Systems for Team Training. Proceedings of the International/Industry Training, Simulation, and Education Conference (I/ITSEC) 2011, Orlando, FL, November 2011.
2. Fletcher, D. and R. Sottilare. "Shared Mental Models of Cognition for Intelligent Tutoring of Teams." Design Recommendations for Intelligent Tutoring Systems – Volume 1: Learner Modeling. (In Eds. Sottilare, Graesser, Hu, and Holden). Chapter 22, pp. 239-254.
3. J. Wilson, "Trust and conflict at a distance: How can I improve relational outcomes in distributed work groups?," in Developing and enhancing teamwork in organizations- evidence based best practice and guidelines, E. Salas, S. Tannenbaum, D. Cohen and G. Latham, Eds. San Francisco, CA: Jossey-Bass, 2013, pp. 268-297.

4. K. VanLehn, “The behavior of tutoring systems,” in International Journal of Artificial Intelligence in Education., 2006, pp. 227-265.
5. E. A. Linnenbrink, and P. R. Pintrich, “Motivation as an enabler for academic success,” School Psychology Review, vol. 31, pp. 313-327, 2002.
6. R.R., McCrae, and P.T. Costa, “The stability of personality: observations and evaluations,” Current Directions in Psychological Science, vol. 3, pp. 173–175, 1994.
7. Burke, Hughes, Marlow, Ogelsby, Savage, Sonesh, Stowers, Wiese, & Salas. “Towards a scientifically rooted design architecture of team process and performance modeling in adaptive, team-based intelligent tutoring systems,” In progress, Final Report (Year 1). USA, Orlando, FL: Army Research Laboratory.
8. S. Decuyper, F. Dochy, P. Van den Bossche, “Grasping the dynamic complexity of team learning: An integrative model for effective team learning in organizations,” Educational Research Review, Vol. 5, Issue 2, , Pages 111-133. 2010.
9. M. Buljac-Samardzic, C. M. Dekker-van Doorn, J. D.H. van Wijngaarden, K. P. van Wijk, “Interventions to improve team effectiveness: A systematic review,” Health Policy, Vol. 94, Issue 3, pp. 183-195, 2010.
10. J. D. Sterman, "System dynamics modeling: Tools for learning in a complex world," California management review, California, CA, 2001, pp. 8–25.
11. X. Kefan, C. Gang, D. D. Wu, C. Luo, W. Qian, “Entrepreneurial team's risk-based decision- making: A dynamic game analysis,” International Journal of Production Economics, Vol. 134, Issue 1, pp. 78-86, 2011.
12. S. C. Kim, S. H. Chang, G. Heo, “Team crystallization (SIO2): Dynamic model of team effectiveness evaluation under the dynamic and tactical environment at nuclear installation,” Safety Science, Vol. 44, Issue 8, pp. 701-721, 2006.