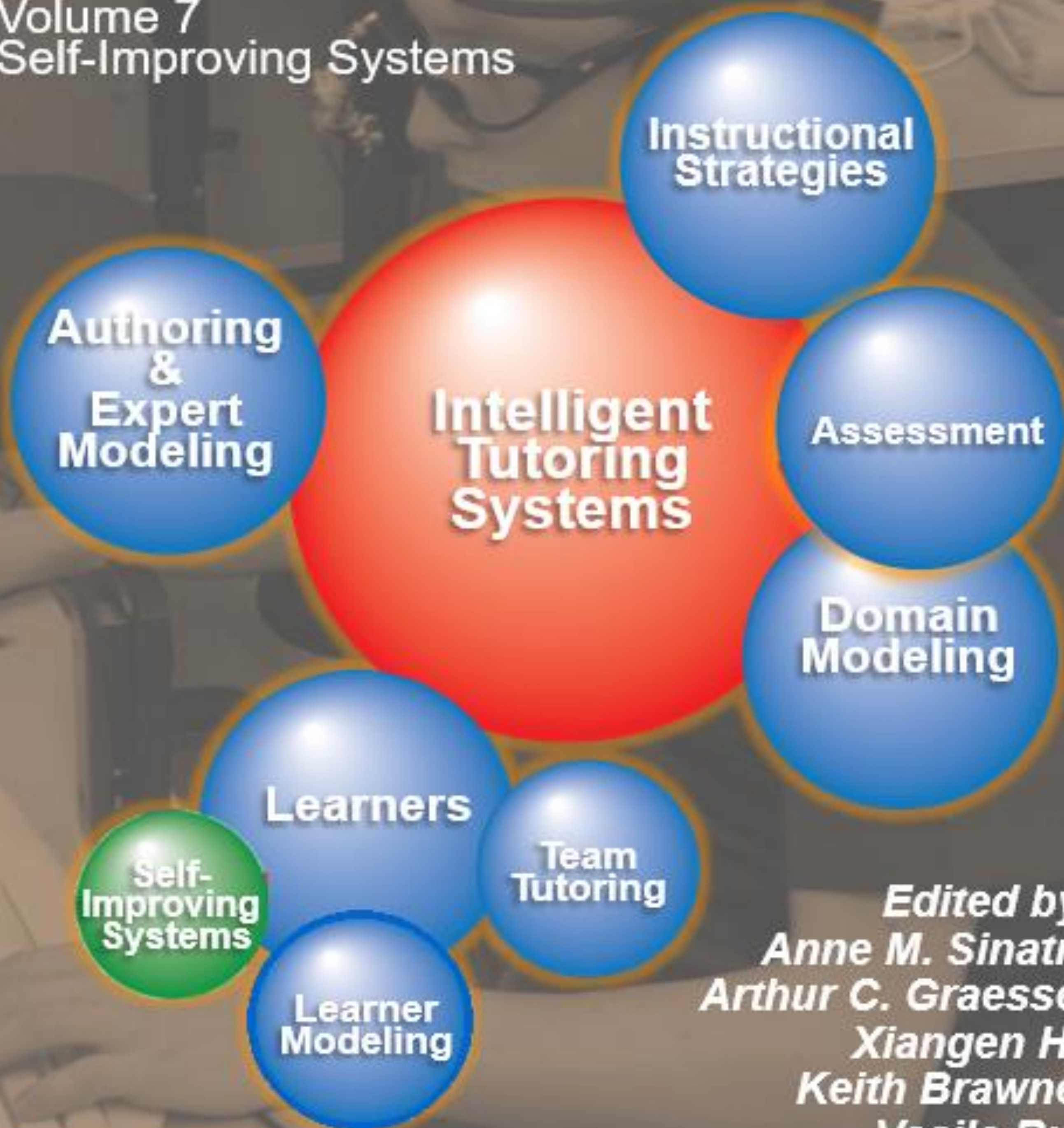
Design Recommendations for Intelligent Tutoring Systems

Volume 7
Self-Improving Systems



Edited by:
Anne M. Sinatra
Arthur C. Graesser
Xiangen Hu
Keith Brawner
Vasile Rus

A Book in the Adaptive Tutoring Series

Design Recommendations for Intelligent Tutoring Systems

Volume 7
Self-Improving Systems

Edited by:
Anne M. Sinatra
Arthur C. Graesser
Xiangen Hu
Keith Brawner
Vasile Rus

A Book in the Adaptive Tutoring Series

Printed in the United States of America
First Printing, October 2019

US Army Research Laboratory
Human Research & Engineering Directorate
Orlando, Florida

International Standard Book Number: 978-0-9977257-7-3

We wish to acknowledge the editing and formatting contributions of Carol Johnson, ARL and Elyse Burmester, Dignitas.

Special thanks to Jody Cockroft, University of Memphis, for her efforts in coordinating the workshop that led to this volume.

Dedicated to current and future scientists and developers of adaptive learning technologies

CONTENTS

INTRODUCTION TO SELF-IMPROVING SYSTEMS & GIFT

Anne M. Sinatra[1], Arthur C. Graesser[2], Xiangen Hu[2], Keith Brawner[1], and Vasile Rus[2] Eds.

[1]U.S. Army Combat Capabilities Development Command – Soldier Center – Simulation and Training Technology Center
[2]University of Memphis Institute for Intelligent Systems

This book on self-improving systems is the seventh in a planned series of books that examine key topics (e.g., learner modeling, instructional strategies, authoring, domain modeling, assessment, team tutoring, self-improving systems, and data visualization) in intelligent tutoring system (ITS) design. This book focuses on self-improving systems and the application of artificial intelligence in ITSs. The chapters within this book specifically examine topics in relationship to the Generalized Intelligent Framework for Tutoring (GIFT) (Sottilare, Brawner, Goldberg & Holden, 2012; Sottilare, Brawner, Sinatra, & Johnston, 2017). GIFT is an open-source, domain-independent, modular, service-oriented architecture for ITSs. The design of GIFT allows for reusability, reduction in authoring time, and reducing the skill level needed to create an ITS. GIFT provides functionality to create ITSs, distribute ITSs to learners through the Cloud, conduct research to evaluate ITSs, and to examine instructional outcomes.

Along with this volume, the first six books in this series, Learner Modeling (ISBN 978-0-9893923-0-3), Instructional Management (ISBN 978-0-9893923-2-7), Authoring Tools (ISBN 978-0-9893923-6-5), Domain Modeling (978-0-9893923-9-6), Assessment Methods (ISBN 978-0-9977257-2-8), Team Tutoring (ISBN 978-0-9977257-4-2) are freely available at www.GIFTtutoring.org.

We believe this book can be used as a design tool for self-improving ITSs. Before we discuss aspects of tutoring and ITSs, it is important to clarify what we and other stakeholders mean by self-improving systems.

Self-Improving Systems

ITSs are all different based on the goals and focus of the authors of the systems. The materials that are contained within the ITSs, the learner characteristics and actions that are to be tracked, and the types of adaptations that are provided by the ITSs vary depending on the specific system. While these adaptations and the desired feedback are generally static and determined ahead of time with an ITS (e.g., if the learner makes this specific mistake, then present feedback number 1), there may be an advantage to dynamically adapting strategies and content that have been found to be most effective with other learners in the ITS.

Utilizing artificial intelligence techniques to create self-improving systems could result in improved learning outcomes. For the purposes of the current book, we define self-improving systems in ITSs, as systems that continually improve by examining the positive and negative outcomes associated with tutoring, and adjusting the tutor to include the materials/paths with the most optimal outcomes. Within the current book different approaches to establishing self-improving systems, such as machine learning, are discussed, as well as specific considerations that should be taken into account when designing these systems.

GIFT and Expert Workshops

In 2012, Army Research Laboratory (ARL) with the University of Memphis developed expert workshops of senior tutoring system scientists from academia and government to influence the GIFT design goals moving forward. Expert workshops have been held each year since 2012 resulting in volumes in the ***Design Recommendations for Intelligent Tutoring Systems*** series the following year. In 2018, parts of ARL, including the GIFT team, were reorganized into another organization, Soldier Center. Research into applied adaptive tutoring and team tutoring have continued with Soldier Center. Additionally, the expert workshops and books have continued with topics in line with the relevant research gaps.

The *learner modeling* expert workshop was completed in September 2012 and Volume 1 followed in July 2013. An expert workshop on *instructional management* was completed in July 2013 and Volume 2 followed in June 2014. The *authoring tools* expert workshop was completed in June of 2014 and Volume 3 was published in June 2015. The *domain modeling* expert workshop was held in June 2015 and Volume 4 was published in July 2016. The *assessment* expert workshop was held in May 2016 and Volume 5 was published in June 2017. The *team tutoring* expert workshop was held in May 2017 and Volume 6 was published in August 2018. We recently conducted a workshop on *self-improving systems in intelligent tutoring systems* in May 2018, and Volume 7 is the current publications. Future expert workshops are planned for *data visualization*.

Design Goals and Anticipated Uses of GIFT

GIFT was designed with multiple functions in mind:

1. An architectural framework that is modular, and has components that can be replaced and customized by ITS authors for their specific tutor.

2. A set of authoring tools which allows subject matter experts, and those without a background in computer science to easily create customized ITSs.

3. A testbed for experimental research, which allows for the examination of research questions relevant to the continued development of ITSs.

The chapters within the book provide recommendations for how to implement the methods within the GIFT architecture with the above functions in mind.

How to Use This Book

This book is organized into four sections:

I. The Systems Perspective

II. Machine Learning

III. Content Authoring

IV. Social Perspectives and Human Factors

Section I, *The Systems Perspective*, explores different systemic views of self-improving systems for adaptive training. This section discusses different self-improving strategies, approaches, and mechanisms for intelligent tutoring systems. Using GIFT as a basis, this section discusses the different implementations of self-improving systems, and suggestions for future systems.

Section II, *Machine Learning*, highlights the different machine learning techniques that can be implemented in ITSs. The chapters in this section discuss a number of different machine learning techniques including sequence mining, discriminate subsequence analysis, multi-armed bandits, and reinforcement learning.

Section III, *Content Authoring*, discusses different approaches to creating content in self-improving ITSs, and considerations for creating content in these specific cases. The approaches that are discussed include crowdsourcing from learner data, refining content based on learner input, and algorithmically defining the content to be used in the ITS.

Section IV, *Social Perspectives and Human Factors*, focuses on considerations of the human in the loop. There is discussion about how social interactions between students can be leveraged to improve tutoring, and how tutors can be improved using design principles of human factors. There is additional discussion of designing tutors for ethics, and an overall perspective on the future of intelligent tutoring.

Chapter authors in each section were carefully selected for participation in this project based on their expertise in the field as ITS scientists, developers, and practitioners. *Design Recommendations for Intelligent Tutoring Systems: Volume 7 – Self-Improving Systems* is intended to be a design resource as well as a community research resource. We believe that Volume 7 can serve as an educational guide for developing ITS scientists and as a roadmap for ITS research opportunities. The authors of the chapters contained herein are experts in their area and the references provided (their own and those of others) compose a rich web of working professionals in the ITS field.

References

Sottilare, R.A., Brawner, K.W., Goldberg, B.S. & Holden, H.K. (2012). *The Generalized Intelligent Framework for Tutoring (GIFT)*. Orlando, FL: U.S. Army Research Laboratory Human Research & Engineering Directorate (ARL-HRED).

Sottilare, R., Brawner, K., Sinatra, A. & Johnston, J. (2017). An Updated Concept for a Generalized Intelligent Framework for Tutoring (GIFT). Orlando, FL: *US Army Research Laboratory*. May 2017. DOI: 10.13140/RG.2.2.12941.54244.

SECTION I – THE SYSTEMS PERSPECTIVE

Dr. Vasile Rus, Ed.

CHAPTER 1 – INTRODUCTION TO THE SYSTEMS PERSPECTIVE SECTION

Vasile Rus
University of Memphis

Core Ideas

The contributions in this section describe different systemic views of Adaptive Instructional Systems (AISs). These views include different strategies, mechanisms, and approaches to improve AISs. A key theme is that AISs are sophisticated systems that must adapt/tailor their operations to each individual learner. Furthermore, AISs interact with the learner and the wider environment in complex ways which should indicate considering a complex system perspective when modelling such systems. Based on different types of adaptivity (macro vs. micro) or various components of the AISs, they can be improved in a number of ways such as embedding learning mechanisms that will improve components based on experience, e.g., reinforcement learning methods, or simply intelligently selecting an external service for a given functionality (selecting a speech-to-text service in order to translate spoken learner input to written form).

Individual Chapters

The chapter by ***Hu, Tong, Cai, Cockroft, and Kim*** discusses a model of Self-Improvable Adaptive Systems (SIAIS) based on a "symmetric" structure in which "both *human learners* and *self-improvable learning resources* change/improve similarly and adapt to each other." They present their own AIS structure built around the following four components: the learners, self-improvable learning resources, learning environments, and learning processes. They then argue that all four components of the AIS are improvable.

The chapter by ***Tong, Rowe, and Goldberg*** distinguishes between the two different types of SIAIS: macro-level SIAIS and micro-level SIAIS. After presenting their own view of the generic architecture of an AIS, the authors focus on the main functionality of such AISs which is adaptivity to each individual learner. Following in the tradition of VanLehn and Rus who identified and depicted several levels of adaptivity, Tong, Rowe, and Goldberg discuss two major levels: macro-adaptivity (corresponding to the so-called outer-loop) and micro-adaptivity (inner-loop, which seems to include also the hint-level loop). The chapter then describes mechanisms to enable macro- and micro-level adaptivity in SIAIS such as reinforcement learning or genetic algorithms.

The chapter by ***Sottilare, Sinatra and DeFalco*** presents a new perspective on AISs as complex, self-improving systems. Their thesis is that the learner, AIS, and environment interact in complex ways and therefore should be regarded as a complex system. The chapter identifies areas for AIS development and integration of existing capabilities to enhance the complexity, granularity and sophistication of methods used by AISs to make effective instructional decisions.

The chapter by ***Nye, Thaker, Auerbach, and Brawner*** argues that one way for AISs to improve is to improve its components. Components could be improved, i.e., become more effective, either by directly improving "due to new data" or by intelligently selecting between different services that provide the same capabilities. For the latter approach, they detail their work on a Multi-Agent Architecture (MAA) framework for the Generalized Intelligent Framework for Tutoring (GIFT) whose purpose is to register and instantiate arbitrary services that are associated with a specific course. They note that the GIFT MAA agents framework offers a possible testbed for exploring self-improving systems.

Chapter 2 - Self-Improvable Adaptive Instructional Systems (SIAISs) – A proposed model

Xiangen Hu[1,2], Richard Tong[3], Zhiqiang Cai[1], Jody L. Cockroft[1], and Jong W. Kim[4,5]
University of Memphis[1], Central China Normal University[2],
Squirrel AI Learning[3],
U.S. Army Combat Capabilities Development Command – Soldier Center – Simulation and Training Technology Center[4], Oak Ridge Associated Universities[5]

Introduction

This chapter tries to consider the fact that adaptivity as a basic fact in instruction (or teaching & learning) environments can be dated as early as the times of Confucius and Socrates. Adaptive instruction is probably a natural pedagogy from the very beginning of teaching practice. About 2500 years ago, Confucius explained this when he was asked about why he gave very different answers to the same question asked by different students (Eno, n.d.). Adaptivity in learning and teaching are so important that psychologists have tried to systematically study it in instruction for many decades at various levels of detail. For example, Atkinson (1974) identified two major components for Adaptive Instructional Systems (AISs): *(1) the sequence of instructional actions taken by the program varies as a function of a given student's performance history, and (2) the program is organized to modify itself automatically as more students complete the course and their response records identify defects in instruction strategies. (page 336)*; Park and Lee (2003) provided a comprehensive view of AISs. More recently, a group of researchers have begun to move even further by establishing an IEEE standards group which focuses on the systematic study and standardization of AISs ("Adaptive Instructional Systems (C/LT/AIS) P2247.1," n.d.).

For the purpose of this chapter, we consider a model of Self-Improvable Adaptive Instructional Systems (SIAISs). A SIAIS is an extended, compared to Atkinson (1974), but still minimalist system that included explicitly four distinctive components: *human learners*, *learning environments*, *self-improvable learning resources*, and *learning processes*.

- *Human learners* are special components in SIAISs that are assumed to be constantly self-improving (i.e., learning).

- *Self-improvable learning resources* are human resources (trainers/teachers, for example) and some digital resources such as digital tutors. These are capable of changing (improving) constantly.

- *Learning environments* are the diverse physical or virtual locations, contexts, and cultures in which students learn.

- *Learning processes* are *the* instructional sequence for any given domain for a particular learner group (such as grades).

Both *learning environments* and *learning processes* are improvable and likely adaptive. These two components may not be able to improve by themselves. In this chapter, we focus on the components that are *self-improvable*. We will define this in later section. Behaviorally, this SIAIS can be described as

> *Human learners* interact with *self-improvable learning resources* in a given l*earning environment* following preset steps of *learning processes.*

This proposed model focuses on the relations between *human learners* and *self-improvable learning resources.* Specifically, this model has a "symmetric" structure in which both *human learners* and *self-improvable learning resources* change/improve similarly and adapt to each other. We first offer an observation from classic Item Response Theory (IRT).

Observed Symmetry in Item Response Theory (IRT)

If we assume that a learner learns during tests with feedback, then the following case fits the definition of an AIS:

An online test involved with M learners and N questions. Each learner is required to answer all N questions, and the result of each answer is dichotomous ("correct", "wrong"). After the learner answered a question feedback is always given (assume this is where the instruction happens). The presentation order of the problems is individualized based on the response history of each learner (this is why we call it "adaptive").

In this AIS, the four components are the learners (M learners), the learning resources (N problems), the learning environment (online), and the processes (individualized presentation order and feedback). The model of this AIS is the classic IRT. An IRT model for this AIS considers two sets of parameters: the item difficulties parameters for the N problems and the abilities parameters for M learners, Rasch model for dichotomous data links these two sets of parameters in the following way: the probability of learner i answering problem j correctly is $P_{ij} = \frac{1}{1+e^{-(\alpha_i-\beta_j)}}$. Equivalently, the probability of problem j being answered by learner i incorrectly is

$$Q_{ij} = 1 - P_{ij} = 1 - \frac{1}{1+e^{-(\alpha_i-\beta_j)}} = \frac{e^{-(\alpha_i-\beta_j)}}{1+e^{-(\alpha_i-\beta_j)}} = \frac{1}{1+e^{-(\beta_j-\alpha_i)}}$$

As it can been seen, in this simple IRT, both the *learners* and the *problems* are *mathematically symmetric*. It is also interesting to notice that if two sets of parameters, $(\alpha_1, \ldots, \alpha_M)$ and $(\beta_1, \ldots, \beta_N)$ are the estimates of the learner abilities and problem difficulties, then if $(\alpha_1 + c, \ldots, \alpha_M + c)$ is the new set of learners' abilities, then the difficulties of the problems must be $(\beta_1 + c, \ldots, \beta_N + c)$. This means, the estimate of one set of parameter values is only meaningful in the context of the other sets of parameters values, or the model makes one set of values adapt to another set of values. In this Rasch model for dichotomous data, the symmetry and mutual dependency of the learners and problems offers a general insight to AISs.

Self-improvable Learning Resources

Self-improvable learning resources are those learning resources that can update, retrieve, and utilize their associated memory of the learning activities. A human learner is obviously self-improvable and constantly works to self-improve. Human teachers/trainers, and human study mates are also self-improvable and constantly improving. Some specially designed digital resources can also be self-improvable. We consider two examples:

The first example is a human tutee and human tutor without computers. Assume the tutoring session is face-to-face and the tutor follows a typical expectation-misconception tailored (EMT) tutoring strategy (Graesser et al., 2001). It is obvious that both tutee and tutor are self-improvable. For tutee to self-improve, the tutee needs to remember and learn from prior mistakes and success when interacting with the tutor. For the tutor to self-improve, the tutor needs to have a memory of what have been tutored and whether the teaching was effective to the student. In this example, both self-improvable components (human tutee and human tutor) will need to update, retrieve, and utilize the stored activities during their interaction in the tutoring environment.

The second example is ElectronixTutor (ET) (Graesser et al., 2018). ET is created by the team from the University of Memphis. Behaviorally, ET interacts with a learner in the following steps:

1. Learner login
2. ET retrieve learner's relevant learning history data from the Learning Record Store (LRS)
3. ET recommends an assignable learning activity[1] to the learner based on the history data of the learner.
4. Learner interacts with ET with the recommended activity.
5. ET sends interaction data to the LRS.
6. ET combines the most recent learner's activity with the history data of the learner.
7. ET recommends next learning activity.
8. Learner interacts with ET with the recommended activity.
9. …. repeat beginning at step 5

The more relevant data collected for a given learner, the more efficient and effective the ET will be for the learner. The key for ET to be a self-improving learning resource is to update, retrieve, and utilize stored data to recommend the next learning activity.

In both examples, the learning environment (face-to-face or online) and the processes (the rigid steps of interactions) are relatively stable. After each learning activity, the human learner learns (improves), the memories of both the human and the self-improvable resource updates. Due to the update, retrieval, and utilization of memory, the self-improvable resources improve over time. Behaviorally, the two examples are the same in the sense that **two self-improvable components interact in a given learning *environment* following preset steps of learning *processes***. The keys for self-improving resources to self-improve are the existence of a memory (data) store and ability to update, retrieve, and utilize relevant memory (data) from previous learning activities.

We define SIAISs as AISs that include at least one self-improvable learning resource.

In this section, we purposefully equate the human tutor and ET as self-improvable counterparts of human learners in SIAISs. In the next section, we try to show that it is actually reasonable to equate the human learner and its self-improvable counterpart in a formal model.

A Proposed Model for SIAISs

For the purpose of the current chapter, we take a minimalistic, behavioristic view of AISs that contains only four components: the *learners*, the self-improvable learning *resources*, the learning *environments*, and the

[1] The assignable learning activity can be read a static document, interact with a digital tutor, or work on problems.

learning *processes*. In general, all four components of the AIS are improvable. For example, learning environments such as schools, classrooms, and laboratories improve each time when there are relevant theoretical and technological advancements. Through history, classroom conditions have improved from chalk and blackboard classrooms to a more modern classroom equipped with LCD projectors and networked computers. The improvements of learning environments, learning processes, and some learning resources are made from external effort and hence are not self-improvable. On the other hand, human *learners* are obviously self-improvable. Human resources (trainers/teachers) are self-improvable. They accumulate their instructional knowledge and skills by interaction with learners. Some specially designed digital resources such as computer based tutors are self-improvable.

We are now ready to propose a model for SIAISs. We keep the same minimalist and behavioristic four component model of AISs with the *Learners*, the *Resources*, the *Environments*, and the *Processes*. From the previous two sections, we observed that human learners, human resources, and some specially designed digital resources are self-improvable if they have the capability of performing real-time updates, as well as being able to retrieve, and process the associated memory (data). We also showed that human learning resources (such as trainers/teachers) can be behaviorally mimicked by specially designed digital resources such as intelligent tutoring systems. SIAISs should contain at least one *self-improvable resource*. To distinguish different types of learning resources, we consider those resources that are not self-improvable (therefore are static) part of the *Environments*. A simple demonstration of symmetry between the human learner and test items in the Rasch model for dichotomous data offered a key feature (#5) of this proposed model for SIAISs:

1. There are four types of components in SIAISs: The Human Learners, the Self-Improvable Learning Resources, the Learning Environments, and the Learning Processes;

2. Human learners and self-improvable learning resources interact in a given learning environment with a preset of processes.

3. Human learners and self-improvable learning resources update, retrieve, and process the associated memory (data) in real-time.

4. Learning environments and processes are relatively static compared to the instantaneous changes of human learners and self-improvable learning resources.

5. Human learners and self-improvable learning resources are *symmetric* in SIAISs when self-improvable resources are playing the roles of teachers. They are self-improvable components with their own properties.

The most important features of the proposed model are a) self-improvable learning resources and the real-time updating, retrieving, and processing of their associated memory of the SIAIS; b) self-improvable learning resources are behaviorally equivalent to the roles of human resources; c) human learners and those self-improvable resources that play the roles of teachers are symmetric in SIAISs.

From this definition, all our educational systems such as physical schools (with students, teachers, classrooms, and curriculum) are SIAISs, because they involve all four components of an AIS, and teachers are self-improvable learning resources. ET also satisfies minimal requirements of an SIAIS. The recommender in ET is a self-improvable component.

Recommendations and Future Research

We consider SIAISs superior to AISs because a SIAIS has at least one self-improvable learning resource. The Generalized Intelligent Framework for Tutoring (GIFT) is an *empirically-based, service-oriented framework of tools, methods and standards to make it easier to author computer-based tutoring systems (CBTS), manage instruction and assess the effect of CBTS, components and methodologies (“Overview - GIFT - GIFT Portal,” n.d.)*. We offer the following recommendations to GIFT, such that the CBTS authored by GIFT are self-improvable and GIFT enabled AISs are SIAISs.

This model emphasizes that the self-improvable learning resources and human learner are *symmetric* in an SIAIS. This means any of the self-improvable resources will likely be interacted with by multiple human learners and develop “personalities” like humans. If the environment of an SIAIS is not physical, then some of the self-improvable resources that are non-human may play roles like real human resources (teachers or study mates). It is important to start building them with care to avoid potential harm to human learners. For this reason, we should pay attention to all aspects of self-improvable resources. We need to understand all details when building, evaluating, and using them in SIAIS.

Behaviors of self-improvable resources should be recorded similarly to that of human learners. In SIAISs, one of the most important properties of self-improvable learning resources is real-time updating, retrieving, and processing of the associated memory of the behavior of the SIAIS. These SIAIS behaviors will include the behavior of all four components; existing standards such as xAPI are designed to only capture human learning behaviors. Within the proposed model, these data standards should be extended to include actors that are self-improvable resources. For example, a dialog-based tutor such as AutoTutor (Nye, Graesser, & Hu, 2014) is a typical self-improvable resource when it is used in an SIAIS such as ET. All dialog movies of AutoTutor should record human learners’ behaviors the same way. The structure of the statements should be the same (such as actor, verb, activity). It is important to note that human learners’ behaviors are observed, but behaviors of self-improvable resources are mostly programmed.

Self-improvable resources should be built based on known effective and efficient human learning principles. Self-improvable learning resources are behaviorally equivalent to the roles of human resources (such as teachers). Cognitive psychologists and education researchers have suggested effective and efficient learning principles (Graesser, Halpern, & Hakel, 2008; Lucariello et al., 2016; Pashler et al., 2007) for teaching and learning communities. They have been used in guiding the creation of educational institutions and training facilities. The building of self-improvable resources should be subject to the same principles (similar to teachers’ training). At least, any self-improvable resource should be well-documented with metadata during its creation.

Models for self-improvable resources should be similar to human learner models. There have been studies of learner modeling from the perspective of intelligent tutoring systems (Sottilare, Graesser, Hu, & Holden, 2013). Within the proposed model, we should study similarly self-improvable resource models from the human learner’s perspective. When we consider human learners from an intelligent tutoring system perspective, we consider the competency (such as knowledge skill, and abilities (KSAs)) of the human learner. Within this model, we should also consider competency of the self-improvable resources. For example, we should consider how a specific self-improvable resource is made, what the rules are, how it is trained, and how well it will get along with human learners and other self-improvable resources. The mathematical models for human learners and self-improvable resources should be the same. Some model parameters of human learners may only be obtained/estimated from the observation (data). Most model parameters of self-improvable resources may be directly obtained.

Conclusions

A minimalist model for SIAISs is presented. The key components of the model include the *human learners*, the *self-improvable learning resources*, the *learning environments*, and *the learning processes*. Self-improvable learning resources can be human resources (such as teachers, trainers, or study mates) or specially designed digital resources (such as digital tutors) that can update, retrieve, and utilize their associated memory (or data). The classification of SIAIS is quite general in that it is applicable to many traditional education agencies such as classes, schools, or school districts. In the context of this current volume, we consider training systems that include self-improvable digital resources such as intelligent tutoring systems. We use the "symmetry" between learners and questions in IRT to show the feasibility of similar symmetry between human learners and self-improvable resources in this model of SIAISs. With the proposed model for SIAISs, we recommend to 1) Create self-improvable learning resources with the guide of effective and efficient learning principles. It is likely the same resources will self-improve into human-like trainer/teachers and interact with thousands of students. We must take care to create them right to avoid potential harm. 2) Some of the research theories and methodologies that have been used to study human learners can be used to study self-improvable learning resources because they are behaviorally similar to humans in SIAISs.

References

Adaptive Instructional Systems (C/LT/AIS) P2247.1. (n.d.). Retrieved February 18, 2019, from http://sites.ieee.org/sagroups-2247-1/

Atkinson, R. C. (1974). Adaptive instructional systems: Some attempts to optimize the learning process. Retrieved from http://www.rca.ucsd.edu/selected_papers/Adaptive%20Instructional%20Systems_Some%20Attempts%20to%20Optimize%20the%20Learning%20Process.pdf

Eno, R. (n.d.). The Analects of Confucius -- An Online Teaching Translation. Retrieved from http://www.indiana.edu/~p374/Analects_of_Confucius_(Eno-2015).pdf

Graesser, A. C., Halpern, D. F., & Hakel, M. (2008). 25 principles of learning. Task Force on Lifelong Learning at Work and at Home Washington, DC.

Graesser, A. C., Hu, X., Nye, B. D., VanLehn, K., Kumar, R., Heffernan, C., … Baer, W. (2018). ElectronixTutor: An intelligent tutoring system with multiple learning resources for electronics. *International Journal of STEM Education*, *5*(1), 15.

Graesser, A. C., Hu, X., Susarla, S., Harter, D., Person, N. K., Louwerse, M., … Others. (2001). AutoTutor: An intelligent tutor and conversational tutoring scaffold. *10th ICAI in Education*, 47–49.

Lucariello, J. M., Nastasi, B. K., Dwyer, C., Skiba, R., DeMarie, D., & Anderman, E. M. (2016). Top 20 Psychological Principles for PK–12 Education. *Theory into Practice*, *55*(2), 86–93.

Nye, B. D., Graesser, A. C., & Hu, X. (2014). AutoTutor and Family: A Review of 17 Years of Natural Language Tutoring. *International Journal of Artificial Intelligence in Education*, *24*(4), 427–469.

Overview - GIFT - GIFT Portal. (n.d.). Retrieved February 21, 2019, from https://gifttutoring.org/projects/gift/wiki/Overview

Park, O.-C., & Lee, J. (2003). Adaptive instructional systems. *Educational Technology Research and Development: ETR & D*, *25*, 651–684.

Pashler, H., Bain, P., Bottge, B., Graesser, A., Koedinger, K., McDaniel, M., & Metcalf, J. (2007). Organizing instruction and study to improve student learning. Washington, DC: National Center for Education Research, Institute of Education Sciences, US Department of Education. *Google Scholar*.

Sottilare, R. A., Graesser, A. C., Hu, X., & Holden, H. (2013). *Design Recommendations for Intelligent Tutoring Systems: Volume 1 - Learner Modeling*. U.S. Army Research Laboratory.

Chapter 3 – Architecture Implications for Building Macro and Micro Level Self-Improving AISs

Richard Tong[1], Jonathan Rowe[2], and Benjamin Goldberg[3]
[1] Yixue Squirrel AI Learning, [2] North Carolina State University,
[3] U.S. Army Combat Capabilities Development Command - Soldier Center - Simulation and Training Technology Center

Introduction

In this chapter, we distinguish between two different types of self-improving adaptive instructional systems (SIAISs): macro-level SIAISs and micro-level SIAISs. Afterward, we explore architectural implications of achieving self-improvements and propose approaches for building each type of SIAIS.

The main design goal of a self-improving system is that it can inspect either a certain component of the system or the whole system itself, and then make an adjustment to improve the parts or the whole based on a set of performance criteria. In either case, self-improvement requires self-awareness through evaluation, and more importantly, a mechanism to self-correct. We decompose the architecture of adaptive instructional systems (AISs) and focus our attention on self-improvement mechanisms in practical implementations. We illustrate real-world AIS architectures and propose approaches such as using continuous inspection and correction, reinforcement learning, and evolutionary algorithms to design and construct both micro-level and macro-level SIAISs.

Definitions

Adaptive Instructional System (AIS)

AISs describe a class of software that includes intelligent tutoring systems (ITSs), adaptive learning technologies, interactive media, serious games, simulations and other learning tools or methods that are used to personalize and optimize instruction for a particular learner or teams. AISs have the common goal of enabling learning in a meaningful and effective manner by using a variety of computing technologies, especially computational intelligence.

In a typical non-adaptive instructional system, instruction is delivered to all students in the same way (i.e., waterfall method), often consisting of a fixed set and sequence of reading materials, videos, and/or exercises to be completed by all students. An AIS, on the other hand, may use individual variability in learning performance, learning pace, preferences, motivation, affective states, and other learner or team attributes together with instructional conditions to identify appropriate learning strategies and tutor actions. Due to recent advances in artificial intelligence, sensing technology, and data mining methods, modern AISs offer novel approaches to engaging students in open-ended tasks and drawing new insights about the learning process. From increasing integration of natural language processing and affective state monitoring, to applications of simulations and interactive media, AISs are increasingly able to capture, process, and fuse high-frequency interaction data and natural rich modalities of communication, such as speech, writing, and nonverbal interaction during real learning activities. This provides unprecedented insight into the moment-to-moment development of a number of learning experiences, especially those involving multiple dimensions of activity and social interaction, enabling researchers to get far more nuanced and complex understanding of student learning processes, something that we have only begun to study at scale.

Self-Improving AISs

The basic promise of a self-improving system is that it can learn from experience. Typically, the prerequisite for self-improvement is self-awareness. Typically, a self-improving system can inspect either a certain component of the system or the whole system itself and then *automatically* make adjustments to improve the parts or the whole based on a set of performance criteria. SIAISs are such systems.

Architecture

Basic AIS Components:

First, we re-examine the architecture of an AIS (Figure 1). In order to achieve basic adaptivity, AISs typically include the following improvable components:

1. Ontology
    a. Knowledge graph of KCs
    b. Knowledge graph of associated domain content
    c. Item organization
2. Content
    a. Instructional items
    b. Assessment items
    c. Intervention resources
    d. Practice or scenario-based resources
3. Learner Goal and Context
    a. Goals: To achieve adaptivity, there must be explicit learning goals so that adaptive decisions can be made. Goals are separated from the algorithms because they are not just static input to the adaptive models, but also could be optimized (improved) themselves.
    b. Context: This includes everything else that is relevant for the recommendation engine to consider in order to make a decision, such as the learner's state, traits, and current learning environment settings.
4. Adaptive Engine (i.e., algorithms and parameters)
    a. Learner progression measurement algorithms: The algorithm could be a knowledge-based heuristics or rule-based program, or it could be machine-learning based such as a probabilistic graphical model or artificial neural network.
    b. Task recommendation and planning algorithms
    c. Engine memory (data)
5. Sensors
    a. Assessment sensors
    b. Student learning context sensors
6. Interface
7. Computer system supporting environment and components. For the purpose of self-improvement examination, we will treat them as static, i.e., outside of the confine of self-improvement.

Key Processes of Adaptivity

There are typically several key steps in most modern AISs to provide adaptivity:

(1) Diagnose or assess learner's knowledge state of all or part of the target area of learning.

(2) Recommend actions to the learner, such as interactions with different content, depending on the knowledge assessment of the learner (as described above).

(3) Interact and teach the learner to understand what they have not learned to improve understanding and learning.

(4) Provide novel practice opportunities with assessments and scaffolding to re-diagnose learner's knowledge state post instructional interaction.

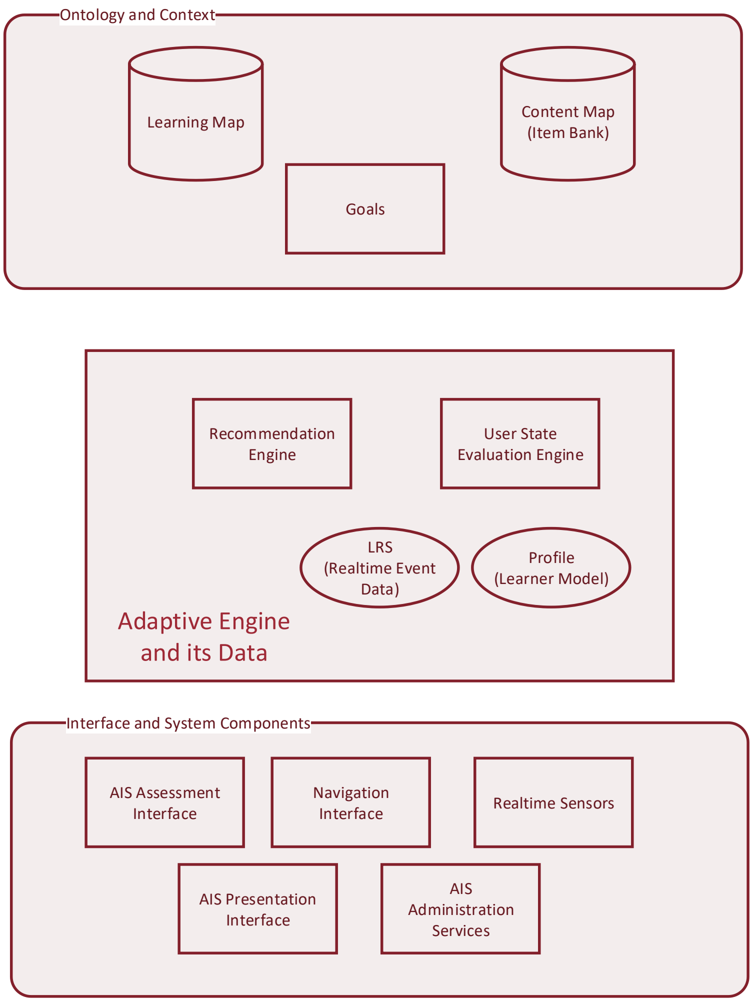


Figure 1. Component architecture for AISs

Note that on point (3) there is still wide variability in the degree to which these functionalities are implemented in AISs. Most AISs have at least some type of simple error-sensitive feedback, while others have intricate systems for analyzing and providing feedback during the student's learning process. We will explain more about the architecture and processes in the later sections of this paper.

Macro-Level and Micro-Level Adaptivity

Adaptivity normally comes at two levels: the macro-level (sometimes called the outer-loop) and micro-level (sometimes called the inner-loop). An illustration of this is shown in Figure 2. In a simple example of an AIS that involves math problems that can be broken down into multiple problem-solving steps, and in which the primary pedagogical mechanism is posing questions and providing feedback to learners, the outer loop tailors the task or problem set that a student sees, and the inner loop personalizes instruction at the level of individual problem-solving steps. The outer loop executes once for each task and iterates over the problems, giving feedback on the problem level (i.e. correct or incorrect) and selects the next problem that is appropriate for the student. The inner loop executes once for each problem-solving step and gives feedback or hints on each step. The inner loop assesses the student's performance and updates the learner model, which is used by the outer loop to select the next appropriate problem for that student (VanLehn, 2006). It does this by looking at the skills that the student has currently mastered, evaluating the student's knowledge state, and selecting the next optimal learning task(s).

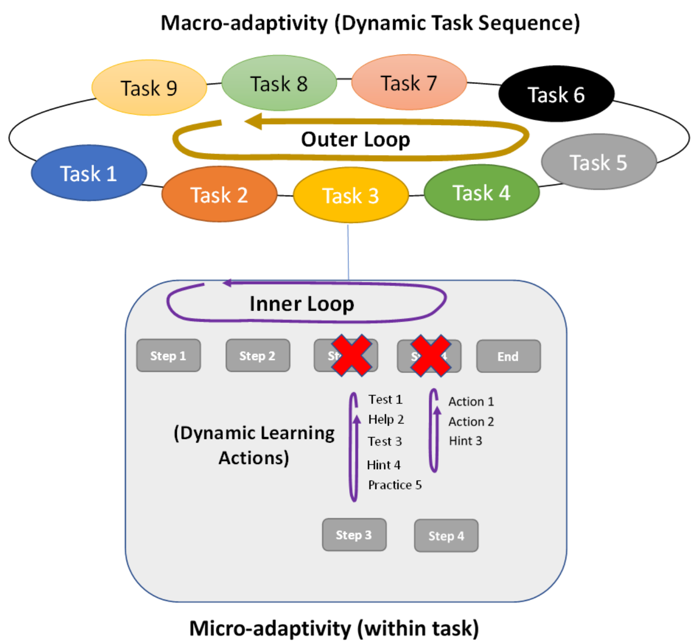


Figure 1. Dual loop behavior of AISs. **Source: Rus & Stefanescu, (2016)**

This simple two-loop model can become complex in a more sophisticated AIS that involves different types of learning tasks. An outer loop interaction may involve videos, interactive simulations, and writing or speaking prompts in which assessment is challenging. The inner loop interaction also depends on the task. For a task that evaluates a student's speaking skill, for example, the inner loop would need to evaluate the student's speaking pattern against an optimal expert. This dialog-based inner loop adaptivity may require a separate ontology from that of the outer loop or a subset of the outer-loop ontology.

The system would first present the learner with the most appropriate task (e.g., problem or action) for the student to complete. Once that recommendation has been accepted by the student, the system will initiate adaptive interaction with the student on the task (i.e., problem).

Compare and Combine the Architecture of Macro-Level and Micro-Level Adaptivity

Figure 3 shows the component architecture of an AIS with both Macro and Micro-level adaptivity.

Macro-level adaptivity

The top portion (the two boxes above the learner record store (LRS)) of the sample AIS architecture shown in Figure 3 performs what we define as the macro-level (i.e., outer-loop) adaptive actions and tasks. The lowest granularity of the macro-level tasks typically corresponds to the finest-grained Knowledge Components (KCs) or learning objectives (LOs) on the learning map (a special type of knowledge graph that captures the relationships among KCs or LOs).

Micro-level adaptivity

The bottom portion (the two boxes below the LRS) of the sample AIS architecture shown in Figure 3 illustrates what we define as the micro-level (i.e., inner-loop) adaptive learning actions. The lowest granularity of the micro-level action typically corresponds to the finest-grained learner actions that can be captured or recommended by the Adaptive Engine, such as a user click, a voice prompt, its corresponding response, or a step in the learner's attempt to complete the task.

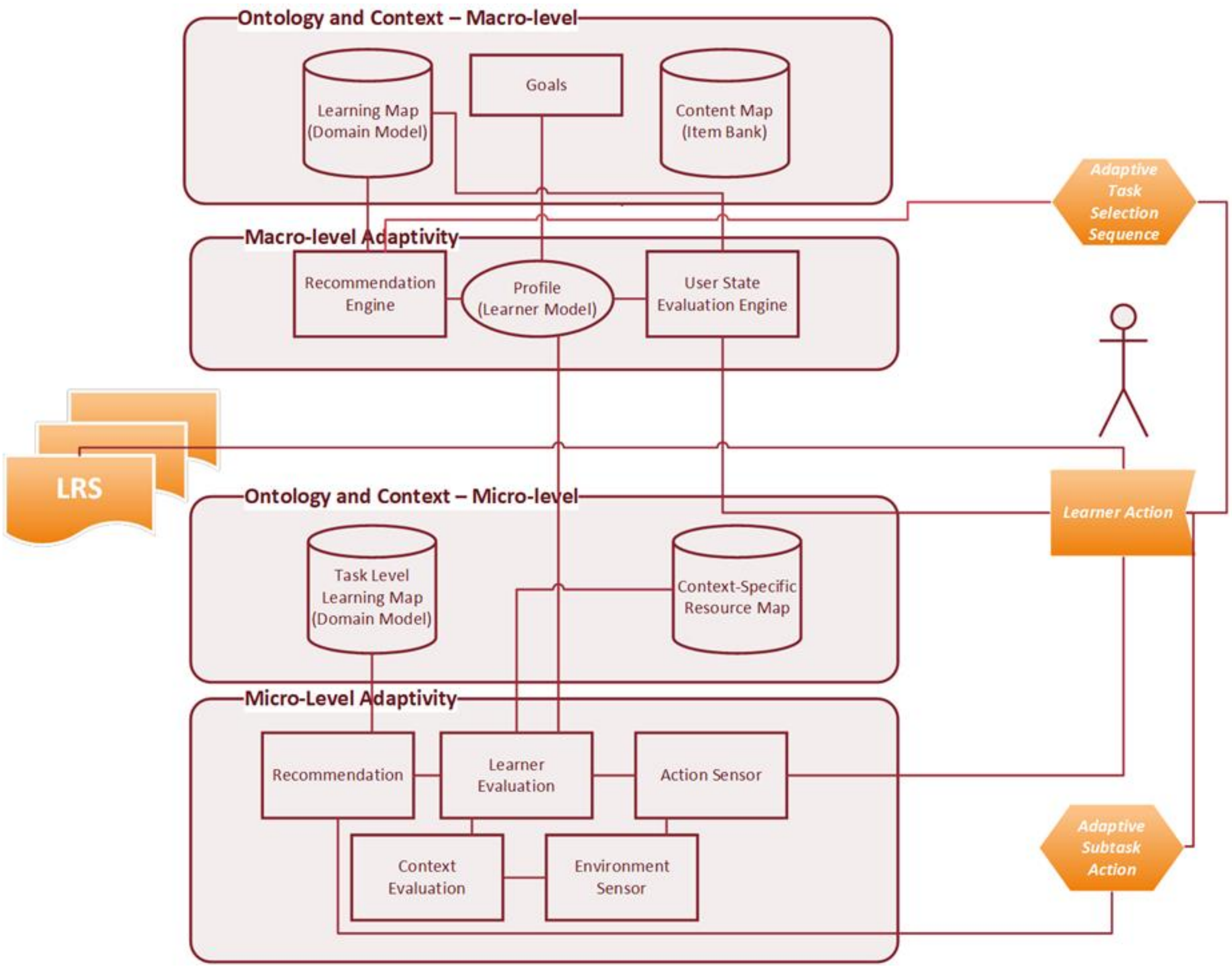


Figure 2: Architecture illustration for AIS with 2 levels (i.e., loops)

Macro-level tasks and resources

The boundary between the micro and macro level of the AIS is the macro-level task. Since each task typically corresponds to a KC or LO, we normally define four types of tasks at the macro-level: (1) assessment, (2) instruction, (3) practice, and (4) intervention. Practice has both assessment value and instructional value. Intervention sometimes does not have a direct impact on KCs or LOs. The static resources for tasks are typically called items. In the diagram, the item bank stores all the macro-level resources for the AIS.

Micro-level actions and resources

Below the macro-level tasks such as assessment, instruction or practice, the interaction between the tutoring agent and the human learner could be either static (e.g., watching a predefined instructional video or practicing a specific problem) or adaptive (e.g., conversing with a robotic tutor; engaging with dynamic challenges, questions, hints, explanations; interacting with game scenes or user menus).

Commonalities between the micro and macro levels

Conceptually, the adaptive engine at either the macro-level or the micro-level needs a minimum ontology such as the learning map, context-specific rules or policies (e.g., classifiers, conditions, heuristics, algorithms) and a pool of recommendable actions or resources so that an appropriate resource or action can be used to adapt to each individual learner's needs and actions dynamically and appropriately at the time of need. In either case, the actions or resources might be either static or dynamic (generated).

Components more specific to macro-level adaptivity

The components for macro-level adaptivity, as the name implies, are normally associated with the outer-loop functions and more focused on planning, task selection and overall learner evaluation. The time span for the learner interaction is typically longer. These components will also provide the learner specific macro-level information to the micro-level (inner-loop) components and collect summary data about the task managed by micro-level (inner-loop) components.

Components more specific to micro-level adaptivity

The components for micro-level adaptivity tend to be much more domain-specific and customized towards the domain model of the KCs or LOs within the context of a specific task. In this case, the interface and pedagogical model design is much more dependent on the domain ontology of the KC and the real-time inferred assessments derived from captured interaction/input data from the student.

Mechanisms for Self-Improvement

Since the early days of computer science, scientists and system designers alike anticipated the creation of a self-improving intelligent system, citing it as a pathway toward the creation of artificial general intelligence (AGI). As early as 1950, Alan Turing wrote:

"Instead of trying to produce a program to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain. Presumably the child-brain is something like a notebook as one buys from the stationers. Rather little mechanism, and lots of blank sheets... Our hope is that there is so little mechanism in the child-brain that something like it can be easily programmed. The amount of work in the education we can assume, as a first approximation, to be much the same as for the human child" (p. 456).

Our goal here for SIAISs is not as grandiose as Turing's. Nevertheless, we would like to examine the basic mechanisms and architecture approach for achieving self-improvement within AISs.

Different Self-Improvement Mechanisms

Common approaches to building self-improving systems can be categorized into the following types: (1) Automated continuous modification, (2) Self-modification through self-learning, and (3) Evolutionary approaches.

Type 1: Automated Continuous Modification

In previous industrial systems, this approach is the most commonly used and the most practical for modern AISs and even non-adaptive instructional systems. The basic mechanism is to have an agent within the system use predefined evaluation criteria to monitor the performance of system components that can be continuously improved. This "auto-improvement" agent will collect data about system component(s) to identify those that need modification and routinely update the system accordingly.

An example routine in a SIAIS might automatically prune content and update content metadata. The process for each iteration of the agent would look like the following:

```
Step 1: Compute item parameters (Item_correlation_factor_to_KC_proficiency,
Item_difficulty, Item_discrimination)

Step 2: Run automatic bad item identification; Notify (Content_editors); Re-
move (bad_items) from (Active_item_bank)

Step 3: If Item_parameter reaches update threshold, Update (item_parameter)
```

This approach works well for content, ontology, model parameters, validity of assessment items, etc.

Type 2: Self-Modification through Self-Learning

Many AISs utilize models induced using machine learning to drive adaptive support for students. A key promise of data-driven modeling techniques is their capacity for self-improvement. For example, reinforcement learning (RL) provides a mechanism to enable models to self-improve through iterative accumulation of experience and reward. An example of a self-improvement mechanism using RL is the optimization of an adaptive recommendation policy, where the parameters for the current model become the target for continuous improvement.

Another self-improvement mechanism is apprenticeship learning, which is a form of learning from demonstration (Abbeel & Ng, 2004). For example, we can use human teachers to instruct an adaptive agent throughout the adaptive learning process while the model improves by leveraging additional input from human experts.

Type 3: Evolutionary Approaches

For certain components of an AIS, especially models and algorithms, evolutionary algorithms such as genetic algorithms or genetic programming, which optimize system parameters with respect to some well understood fitness function through an iterative process, show promise. However, this approach will probably require a simulated environment to iteratively generate a large number of adaptive sequences.

Approaches for Different Target Components for Self-Improvement

In the AIS context, each of the components could become a target for the agent's continuous improvement. For micro-level and macro-level AISs, there are differences in the role and type of these components. Since the different self-improvement mechanisms usually apply to specific types of components, we would probably take different approaches to designing and implementing SIAISs. The following tables compare the macro and micro-level SIAISs and suggest appropriate mechanisms.

Macro-Level SIAIS

The focus for the macro-level AIS is to optimize the learner's task selection and learning sequence to accelerate knowledge acquisition and support retention for future application. Therefore, the key is to improve the model, ontology, and content. The table below maps the different components to the SI approach.

Macro-Level Components	Type	Improvement Approach Type
1. The adaptive engine (algorithm and parameters)		
a. Learner progression measurement algorithm	Model	2,3
b. Task item recommendation and planning algorithms	Model	2
c. The software code	Program	
d. The rules and heuristics	Program	1
2. The sensors		
a. The student learning context sensors	Program	N/A
3. The ontology		
a. The knowledge graph of KCs	Graph	1
b. The knowledge graph of associated contents	Graph	1
c. The item organization	Data	1
4. The contents		
a. The instructional items	Content	1
b. The assessment items	Content	1
c. The intervention resources	Resource	1
5. The learning goals		
a. Static Goals	Content	1
b. Dynamic Goals	Model	2,3
6. The interface		
a. Menu and Navigations	Program	N/A

Micro-Level SIAIS

The focus for the micro-level AIS is to optimize the learning within the task. Therefore, the higher priority work is to improve the model for content generation and interface generation. The table below maps the different micro-level SIAIS components to the SI approach.

Micro-Level Components	Type	Improvement Approach
1. The adaptive engine (algorithm and parameters)		
a. Learner progression measurement algorithm	Model	2,3
c. Sub-Task steps recommendation and planning algorithms	Model	2,3
d. The software code	Program	
e. The rules and heuristics	Program	1
f. Engine Memory – Data	Model	1
2. The sensors		
a. The assessment sensors	Program	N/A
b. The student learning context sensors	Program	N/A
3. The ontology		
b. The knowledge graph of associated contents	Graph	1
d. The knowledge graph under KCs	Graph	1
4. The contents		
a. The instructional items	Content	1
b. The assessment items	Content	1
c. The intervention resources	Resource	1
5. The learning goals		
a. Static Goals	Content	1
6. The interface		
a. Dialog Interface	Program	2

Recommendations and Future Research

In the previous sections, we have used component architecture analysis to help map different implementation approaches to two types of SIAISs. Future research directions include the following:

1. Implement one to two pilot projects for both macro-level SIAISs and micro-level SIAISs by applying the discussed techniques to confirm or refute the proposed approach.
2. Refine both the theoretical foundation and practical implementation of the models and algorithms in the field of SIAISs.
3. Our initial next-step focus would be approach 2 for the adaptive engine (including models, ontology and goals) using RL, Generative Adversarial Networks (GANs), apprentice learning, and simulated students.

The Main Challenges for Different Self-Improvement Approaches

For any of these self-improvement approaches, there are several common challenges.

1. The amount and quality of the data needed by any self-improvement algorithm is substantial. In general, AISs require substantial scale of operation in order to have adequate data to run such algorithms. This issue is specifically acute for deep learning algorithms.
2. The objectivity and quality of performance measurement indicators is also critical. Whether it is to make rule-based decisions or implement self-learning or evolutionary processes, it is critical to have high-quality, objective measurements of existing system components in terms of effectiveness

and results. For example, if we use learning gain as the main criterion for measuring the impact of content, or the effectiveness of a machine learning algorithm, any bias or inaccuracy in such measurements will cause ambiguity in whether self-improvement can be consistently obtained.
3. It can be unclear how to determine the frequency of system changes, or what criteria to use for guiding system changes. Currently, this is more of an art than a science in real-world systems.

There are also specific challenges for the three approaches discussed above:

4. For automated continuous modification, a primary challenge is the lack of historical interaction data for new or untested content items. Without longitudinal comparisons, decisions about self-improvement prove difficult.
5. For self-learning and evolutionary approaches, it can be difficult to obtain an adequate number of users to run RL with data from real students. This is true even in large-scale deployments, such as those typified by popular MOOC courses. Therefore, it is imperative to use simulated learners to generate sufficient data for training and optimization. However, the design of simulated learners raises a host of challenges that are beyond the scope of this chapter.

Conclusion

SIAISs show significant promise for the next generation of advanced learning technologies. In this chapter, we use an architecture-analysis approach to identify self-improvement opportunities around core AIS components at both the macro-level and micro-level. The three suggested mechanisms—automated continuous modification, self-modification through self-learning, and evolutionary approaches—provide a starting point for the design and development of practical SIAISs.

Initial work on domain-general approaches to developing SIAIS functionalities are beginning to emerge. For example, Rowe et al. (2018) have integrated a mechanism for macro-level self-improvement within the Generalized Intelligent Framework for Tutoring (GIFT). Specifically, they have developed an adaptive courseflow object within GIFT that supports data-driven models of instructional feedback based upon Chi's ICAP model of active learning (Chi & Wylie, 2014). Their work uses RL to automatically induce and refine tutorial policies that control pedagogical decisions about ICAP-based instructional remediation within adaptive training courses. This is illustrative of emerging efforts to develop domain-general SIAIS capabilities, and it points toward future directions for the development of AI-enhanced instructional systems that automatically self-improve as learners use them. Extending SIAIS capabilities to other components of GIFT, such as learner modeling and task selection, and extending these capabilities to team-based training, will be important to progress toward realizing the vision of practical, domain-general SIAISs. Furthermore, it will be critical to conduct rigorous evaluation studies with human learners to validate the effectiveness of SIAIS capabilities in run-time instructional environments.

References

Abbeel, P., & Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. Proceedings of the Twenty-First International Conference on Machine Learning (p. 1). New York, NY: ACM.

Chi, M. T., & Wylie, R. (2014). The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational Psychologist, 49*(4), 219-243.

Rowe, J., Spain, R., Pokorny, B., Mott, B., Goldberg, B., and Lester, J. (2018). Design and Development of an Adaptive Hypermedia-Based Course for Counterinsurgency Training in GIFT: Opportunities and Lessons Learned. In Proceedings of the Sixth Annual GIFT User Symposium (GIFTSym6), pp. 229-239, Orlando, FL.

Rus, V., & Stefanescu, D. (2016). Non-intrusive assessment of learners' prior knowledge in dialogue-based intelligent tutoring systems. *Journal of Smart Learning Environments*, *3*(2), 1-18.

Turing, A. (1950). Computing Machinery and Intelligence. *Mind*, *59*(236), 433-460.

CHAPTER 4 – CONSIDERATIONS IN MODELING ADAPTIVE INSTRUCTION AS A COMPLEX SELF-IMPROVING SYSTEM

Robert A. Sottilare[1], Anne M. Sinatra[2], and Jeanine A. DeFalco[2,3]
[1]Soar Technology, Inc., [2]US Army Combat Capabilities Development Command – Soldier Center – Simulation and Training Technology Center, [3]Oak Ridge Associated Universities

Introduction

This chapter examines the modeling of adaptive instructional systems (AISs) as complex, self-improving systems. According to Sottilare and Brawner (2018), AISs are "computer-based systems that guide learning experiences by tailoring instruction and recommendations based on the goals, needs, and preferences of each individual learner or team in the context of domain learning objectives" (p. 25). We contend that the interaction among the learner(s), the artificially-intelligent AIS that guides the learning experience, and the environment (e.g., simulation, media, problem set) which represents the content which the learner(s) interact(s) with should be considered a complex system and specifically a complex adaptive system (CAS). It is not our intent to design, develop and validate a complex model of adaptive instruction in this chapter. Our goal is to identify areas for AIS development and integration of existing capabilities to enhance the complexity, granularity and sophistication of methods used by AISs to make effective instructional decisions. Toward this goal, we also provide recommendations for further development of the Generalized Intelligent Framework for Tutoring (GIFT; Sottilare, Brawner, Goldberg & Holden, 2012; Sottilare, Brawner, Sinatra, & Johnston, 2017).

Relating Complexity Theory to AISs

Complexity theory focuses on uncertainty and non-linearity in systems and suggests that elements of a system act together to influence the system's collective behavior and interaction with its environment (Bar-Yam, 2002). In complex systems, perfect understanding of the individual elements or processes in a system do not automatically translate into a perfect understanding of the whole system's behavior (Miller & Page, 2007). In other words, the ability to model all system components or understand its processes is not a guarantee of the ability to accurately predict its behaviors or outcomes. While complex systems may seem chaotic, close examination can identify order, structure, and trends in their processes.

If we expect a complex system to grow and learn over time, we identify the system as a CAS. A CAS possesses the intelligence to learn and adapt to changing conditions in its environment to optimize outcomes. In considering the elements of a complex adaptive instructional system (CAIS), we must also consider learning outcomes of interest:

- Learning – the acquisition of knowledge or skill through experiences that are intended to change the long term behavior of the learner

- Performance – the application of knowledge and/or skill to completing a task or function

- Retention – the ability to recall stored knowledge or skill over time

- Transfer – the ability to apply knowledge or skill acquired during instruction in a work environment

By targeting the outcomes in a CAIS, we can more strategically analyze the factors that contribute to these critical outcomes, and thereby enhance opportunities to identify/model its order, structure, and trends. The seeds of a CAIS have already been sown in the learning effect model (LEM) for individual learners and teams (Sottilare, 2012; Sottilare, Burke, Salas, Sinatra, Johnston & Gilbert, 2018). Next, we introduce the LEM and its associated concepts as the foundation to begin the construction of a model for CAISs.

Introducing the Learning Effect Model (LEM)

The major elements of the LEM include the learner, the environment, and the adaptive tutor. Functionally, these elements interact to support the learning experience and to reinforce or improve the decisions made by the adaptive tutor by improving its policies. While the LEM is and has been the basis for the design of GIFT, the implementation of GIFT's design to date has been primarily based on static decision making methods (e.g., rules and decision trees) that are not associated with self-improving systems. A recent version of GIFT has implemented Markov Decision Processes to support GIFT recommendations, but a true agent-based approach across all AIS decision making processes has yet to be implemented.

By examining the LEM, we hope to demonstrate the advantages of agent-based approaches (compared to rule-based approaches) to self-improving systems. In Figure 1, agents in the adaptive tutor observe both the learner(s) and the environment, assess their states, and make decisions about learning strategies and tactics to optimize both the learning experience (near term) and learner proficiency (long term).

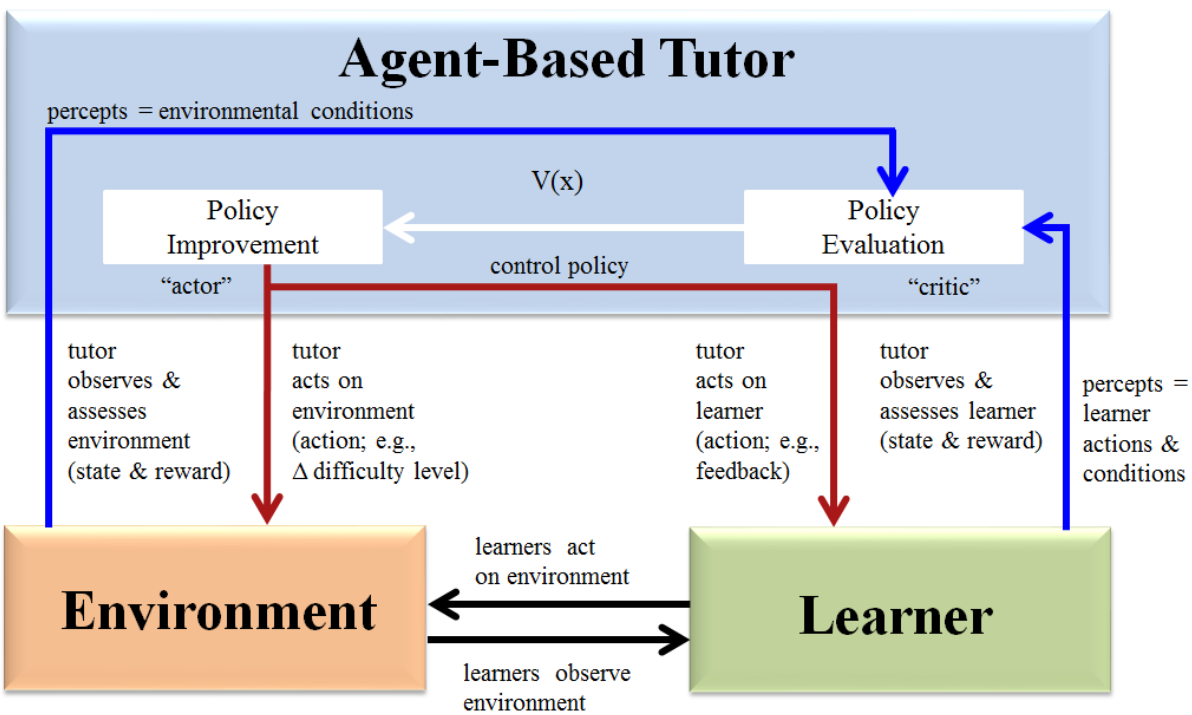


Figure 1. Simple model of learning effect in AISs

Why is it important to represent adaptive instruction as a complex system? Current theories of learning attempt to provide a simple model of instruction which might be simple to implement, but lack the complexity to fully capture the learning process for either individual learners or teams. We contend that in order to fully represent the adaptive instructional process, we must forego simple models and embrace more complex systems that account for all the variables in the system and learner interaction with the environment. Models of learning or instruction provide enticing strategies to be implemented in AISs, but these models are only part of a more comprehensive picture. To truly adapt strategies to individual learners and tailor tactics (actions) to optimize their learning outcomes, we need a complex system that represents all of the antecedents that influence learning and track decision outcomes to reinforce future decisions.

The implementation (authoring process for AISs) should remain simple, but the underlying mechanisms are in truth much more complex and need to be modeled as a complex system. It is only through a CAS model that we see a realistic opportunity to accurately represent the adaptive instructional process, to predict future states of learners, and to understand the effectiveness of AIS decisions and actions in terms of learning, performance, retention, and transfer. "Research on learning and transfer has uncovered important principles for structuring learning experiences that enable people to use what they have learned in new settings" (National Research Council, 2000, p. 4) and it is important that instructional designers apply these important principles, but it may be even more important to understand how to apply them and under what specific circumstances. The circumstances we refer to are the conditions of the learner and the environment which are the basis for triggering and selecting tailored strategies and tactics in AISs.

Now that we have introduced complexity theory, CASs, and the functional elements of AISs, the next step is to dissect AIS decision-making processes into essential steps as they relate to the primary LEM interactions with learners, environments, and tutoring agents. We begin by examining the LEM as it exists today and by identifying its critical decision points in the adaptive instructional process as a basis for a more complex and effective AIS.

Interaction and Decision Making within the LEM

The LEM was originally developed as a method of identifying the interactions and data flow between three essential elements: 1) the learner, 2) the environment, and 3) the adaptive tutor (Sottilare, 2012). Originally called the learning effect chain, the LEM evolved over time to include processes that acquire learner data, derive learner states from that data, select strategies (plans for action) based on learner and environmental conditions, and finally select tactics (tutor actions) with the goal of optimizing learning outcomes as shown in the latest version of the LEM in Figure 2 (Sottilare, Burke, et al., 2018).

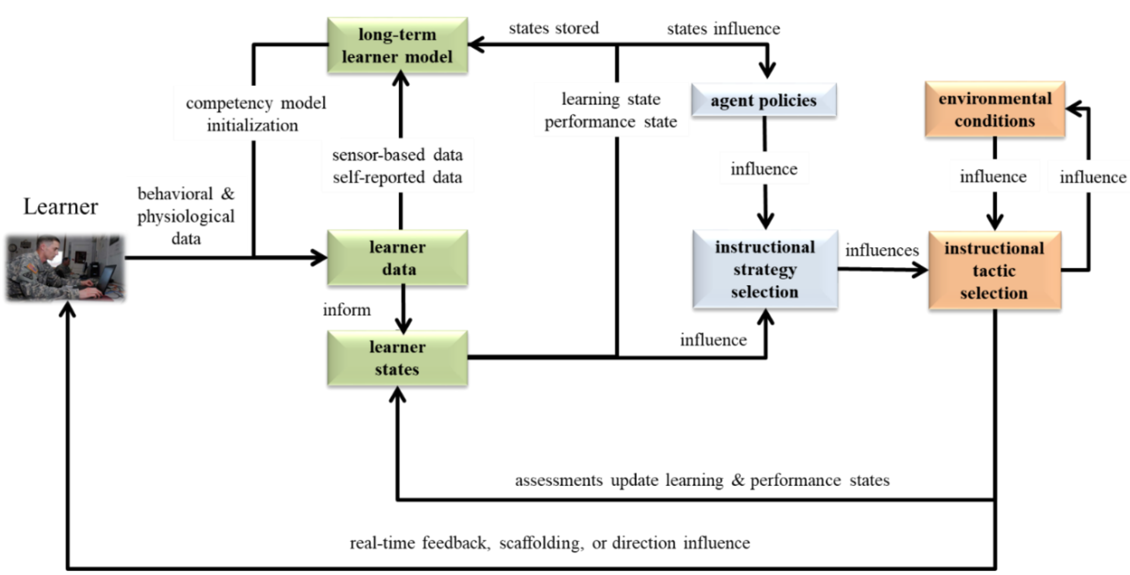


Figure 2. Learning Effect Model for Adaptive Instruction

If we dissect the processes shown in Figure 2, the colored boxes represent more granular elements of the learner (green), the environment (orange) and the tutor (light blue) than shown in Figure 1. In the LEM we make a clear distinction between data and states. Learner data may be behavioral (data associated with actions) or physiological (data associated with physical processes in the body). Data sources include sensors, direct observers or historical records. States (e.g., emotions or motivation) are typically derived from data (facial markers, voice inflection or surveys).

Behavioral data in the form of learner inputs (e.g., surveys, self-reported data or control selections), verbal and non-verbal actions, and other physical acts are captured by sensors for later use. Physiological data (e.g., heart or breathing rates) may also be captured, filtered, and processed to inform/classify derived learner states (real-time or stored long-term). Learner states influence the selection of strategies. Contextual cues from the environment influence tactics along with the selected strategy. Once a decision has been made about an instructional tactic, it has the potential to influence the learner, the environment or both.

While the LEM details decisions like learner state classification, strategy selection, and tactics selection, it does not dictate the process or method for their selection. The mechanics of decision-making in AISs is left largely to the author of the system, but should be influenced by best practices in learning science. Much like a hammer is great for driving nails, but less effective in sawing logs, the effect of best instructional practices is not universally known among all the possible combinations and permutations of learner and environmental conditions. In the next three sections we explore learner models, instructional decisions, and environmental conditions to understand their roles in CAIS decision making.

Understanding Learners in CAISs

Learners by themselves are complex systems in that they are composed of many components which interact with each other resulting in probabilistic learning outcomes. The modeling of learners and processes like perception and learning is therefore also complex. While the LEM concerns itself primarily with learning outcomes, there are many other elements and processes embedded in the learner model that influence those learning outcomes. During instruction there are many states and traits that could be used to describe the learner and their readiness to learn. There are overlapping or concurrent states, uncertainty associated with classification methods, and other context-dependent factors.

The interaction of learners with AISs is even more complex. A goal of this chapter is to more fully define the interactions, states, and processes within the learner that influence learning outcomes (e.g., skill acquisition, retention, and transfer of skills), but to also extend known models of adaptive instruction such as the LEM (Sottilare, 2012; Sottilare, Burke, et al., 2018) to encompass the probabilistic nature of learner states.

Our objective is to reflect the complexity of interaction between the learner(s), the AIS, and the environment (e.g., virtual simulation or serious game) as a system, and to understand the system design and processing required to reinforce system decisions (e.g., recommendations, learning strategies, and computer-based instructional actions) and improve their effectiveness over time. Next, we discuss a few variables of interest and their potential to influence learning outcomes. We begin with perception and learning.

Perception and Learning

> "*Perception yields models of the world that lies outside us,*" (Johnson-Laird, 2012, p. 134).

In order to be able to process information to be learned it first has to be perceived and attended (acted upon). Once the content has the attention of the individual they need to interpret, understand it, and then store it in memory if they choose to. Information that is provided in a tutoring context is often either in the auditory form or the visual form. When provided in the visual form it traditionally includes a linguistic component such as reading directions or words that are associated with it. There are a number of different types of memory that are involved with the interpretation of information by an individual learner.

Perceptual Systems, Memory, and Learning

The modal model of memory (Atkinson & Shiffrin, 1968) is the traditional model that is used to describe the process of bringing in information and remembering it. In the modal model of memory, there are three main types of memory: sensory, short-term and long-term memory (Atkinson & Shiffrin, 1968). Sensory memory relies heavily on the perceptual system and has two different types: echoic and iconic. Echoic memory is auditory based and iconic memory is visual based. In order for information to get processed by an individual they have to pay attention or attend to it. Unattended iconic memory lasts about 1 second, and echoic memory approximately 30 seconds (Atkinson & Shiffrin, 1968). If the information is attended to in that time it can then be moved to short-term memory.

The length of retention in short-term memory varies based on the literature, however, it is generally agreed that once it is in short-term memory the information has to be used, processed or repeated to ensure that makes it to long-term memory. Not every piece of information that is paid attention to will ultimately make it to long-term memory. Rehearsal, or repeating the information helps to retain it in short-term memory, and then ultimately can move it to long-term memory (Atkinson & Shiffrin, 1968). Long-term memory storage is one of the ultimate goals of tutoring.

When designing tutoring you want to ensure that the learner pays attention to it, works with it and re-hearses/interacts with reinforcing instances of it through feedback, interventions and questions so that it gets to long-term memory for later processing and retrieval. Additionally, it has often been theorized that there is a working memory store, which acts as an active short-term memory (Baddeley, 1992). The working memory can be overloaded which can then result in less information being processed and moving on to long-term memory.

The working memory processes both linguistic and visual information. Linguistic information includes the phonological loop which is responsible for processing auditory and verbal information including language and music. Visual information includes the visuo-spatial sketchpad which is responsible for processing visual and spatial information including image retrieval from long term memory. Ideally the amount and type of information that is coming in can be processed through short-term/working memory without over-loading either of these resources, and ultimately reach long-term memory.

Models of Perception and Learning

In education, there is an orientation towards supporting learning experiences that facilitate pragmatic ends. This pragmatism can be understood both as the notion of transfer of skills or more simply as problem solving.

There are two research traditions that address problem solving: research on problem representation (Gestalt legacy) that looks at how people understand problems, and research on how people generate solutions to those problems (Bassok & Novick, 2012). Gestalt psychologists demonstrated how the organizing princi-ples of visual perception (e.g., proximity, good continuation, closure, grouping) in combination with a solver's prior knowledge, effect how people understand and generate problem solutions (Bassok & Novick, 2012).

Problem representation is a model of a problem constructed to summarize the essential nature of that prob-lem. This model might be rendered as an internal representation (mental model) (Lakoff & Johnson, 1980) or external (e.g., diagram) (Card, Mackinlay, & Schneiderman, 1999), both of which can be manipulated to aid in searching for solutions. Hayes and Simon (1977) demonstrated in their famous Tower of Hanoi problem that people's search for solutions depended on various perceptual and conceptual inferences that were drawn from specific representations of a problem's structure. This principle implicates the importance of background knowledge in order to render an initial representation. In the search for a solution, these initial representations may undergo a change as new information informs the model. In this way, problem representations may evolve out of necessity to integrate this new information into the model in order to more effectively find the solution end state or goal.

In essence, the relevant principles of problem-solving are rooted in notions of perception: both in how problems are represented, which determines or constrains generating a solution, and how these constraints shape the problem space to influence the cognitive activities employed in achieving a goal solution state. According to Dunker (1945), these cognitive activities can include: (a) categorizing objects and drawing inferences based on category membership; (b) making inductive inferences from multiple sources: (c) analogical reasoning; (d) identification of causes of events; (e) deductive reasoning; (f) devising judgments; (g) using evidence to make diagnoses.

The import of perception, then, in AISs includes devising platforms that provide the learner with enough relevant background knowledge so the learner can devise a flexible representation of a problem (either internally as a mental model or represented externally through AIS tools), that can be manipulated through cognitive activities in order to reach a solution end state.

AISs include technologies such as Intelligent Tutoring Systems (ITSs), recommender systems, and intelligent media and often have the ability to adjust the materials, strategies, or content that are provided to a learner based on the individual characteristics that are displayed during an interaction. Often times these adjustments can occur based on real-time performance as well as based on trait or state measures that occur during the interaction with the system. An ITS traditionally includes a learner model that is specifically designed for the domain-specific content of the tutor, and focuses on aspects of the learner that the ITS designer feels are relevant to the improvement of performance during the tutoring interaction. Learner models can be used to compare and contrast individual differences, previous learning experiences, prior domain knowledge, current performance, and states such as emotions, motivation or personality traits. While all learner models focus on the characteristics of the learner, they can be defined in very different ways depending on the context and tutor technology – for instance, an algebra tutor's learner model would be expected to be very different than the learner model of an individual engaging in a team learning task in an external computer-based game environment.

Recommendations for Enhancements to the GIFT Learner Model

In terms of recommendations for the design and further development of GIFT, learner models should be purposely built from a historical record or pool of data about the particular learner who is about to experience adaptive instruction. A model of perception, working memory, and the data in the model should be relevant to the domain of instruction and semantically related to prior experiences of the learner to assess their level of proficiency in that domain. This will require assessment of the learner's historical records by a machine learning method that can decipher the relationships between key learner attributes and experiences to determine their domain proficiency with a high level of certainty. High certainty is desirable since subsequent instructional decisions are dependent on the accuracy of assessed learner states. Only in this way can the learner model represent the true complexity of the learner and tailor their instructional experience effectively.

Learner models usually include both static data (e.g., name, gender) and dynamic data usually related to the domain of instruction. We contend that an important piece is missing that directly relates to learning – the representation of the cycle defined by Jung (1932) while exploring psychological types: perception, judgment, and action by the learner. The information being perceived by the learner varies, and there may be a difference between what has been perceived and the ground truth of what was presented to the learner. Working memory might also vary between learners and affect their ability to retain or recall information required to demonstrate learning. This can often lead to learner errors or misconceptions. Perception and memory are not generally represented in learner models today, but are well represented in cognitive models (e.g., Soar, Adaptive Control of Thought—Rational (ACT-R), Sigma). As we explore instructional decision making processes in the next section, we make a case for including cognitive models not only to represent the decision making process, but also to represent the capabilities/limitations of the learner.

Understanding Instructional Decisions & Actions in CAISs

There are several methods used by AISs to drive their actions. Actions range from simple recommendations to more complex interactions (e.g., interactive dialogues, reflective prompts). The goal of these actions is to optimize learning outcomes. In some cases, (e.g., the learning of fundamentals), short term learning may be prioritized over long term learning goals. In most cases, long term learning goals are prioritized above short term learning.

There are also many methods available to drive AIS decision-making. Some are rule-based and others are implemented as decision trees, but rules and decision trees assume a full understanding of the existing conditions and the best options available. In many domains, uncertainty exists about the conditions of the learner and the environment, and the best options available may be sub-optimal. All the possible conditions may not be known and those that are known may not have obvious relationships to desired learning outcomes.

A better option may be an intelligent agent that observes the conditions of the learner and the environment and responds using its *policies - knowledge of the domain* (e.g., expectations, standards, rules, decision trees) to optimize learning. The difference between an agent and more static methods like rules and decision trees is that the agent can learn, modify its policies, and make better decisions as it experiences more learners and environments (Figure 1). Inexperienced agents can bootstrap (initialize) their policies based upon best practices in the literature – specifically *instructional theories*. "*Agent architectures* emphasize the integration of multiple techniques as well as programmability and flexibility" (Laird, 2012, p.11). In order to better understand how learning theories might help initialize agent knowledge, and how agent architectures might help us produce more effective agents, the next two sections touch on both of these topics.

Applying Instructional Theories to CAISs

Instructional theories are design-focused, goal-focused, and probabilistic. They are often described in terms of the likelihood that they will influence learning. Instructional theories are often confused with learning theories which are not prescriptive or probabilistic, but focus on how people learn (National Research Council, 2000; Eaton, 2012). Learning theories are generally grouped into three genres: 1) behaviorism – a long term change in behaviors in response to stimuli; 2) cognitivism – learning resulting from the internal processing of information; and 3) constructivism – mental model building based on experiences. While these are important in explaining how people learn, these models are not critical to design decisions for AISs. Next, we will sample two instructional theories used in GIFT as we examine methods to improve the complexity, granularity, and sophistication of decision making processes in AISs.

Merrill's Component Display Theory (CDT)

David Merrill's Component Display Theory (CDT) (1983) is composed of four quadrants that represent four phases of instruction (rules, example, recall, and practice) that have been generalized across various instructional domains and are a basis of instruction in the AIS architecture GIFT. In the rules phase, the learner is exposed to content that includes facts and tenets about the domain under instruction. For example, if we were teaching a student about baseball, they would need to understand concepts related to batting (e.g., batter's box, stance, grip, swing away, and bunt), fielding (e.g., outfield, ground balls, bases, pop flys, and outs), and pitching (e.g., the mound, the rubber, home plate, strike zone, balls and strikes).

In the example quadrant, the concepts are modeled for the learner. Again, using our baseball example, an instructor might model how to stand in the batter's box, how to field a ground ball, or throw a curveball. The examples presented in this phase are dependent upon facts and tenets learned in the rules quadrant. Once the instructor has presented rules and examples the next step is to assess the ability of the learner to recall the facts, tenets, and examples previously presented. This is critical to the last phase which is practice, where the learner applies knowledge to a practical application of skill. If the learner is unable to recall the in information in the rules or examples quadrant, they will not be able to apply them in practice, Merrill's fourth quadrant.

Interactive, Constructive, Active, and Passive (ICAP) Framework

In 2018, the Engine for Management of Adaptive Pedagogy (EMAP), GIFT's recommender engine was extended to go beyond the CDT instructional model to include the Interactive, Constructive, Active, and Passive (ICAP) framework (Chi, 2009). The ICAP framework defines engagement behaviors as occurring in one of four modes: Interactive, Constructive, Active, and Passive. The ICAP hypothesis predicts that learners become more engaged with content as they transition from passive to active to constructive to interactive and that their learning will also increase. Instead of being limited to passive delivery of new content in the CDT instructional model, the ICAP framework now supports a configurable phase where the AIS author can select content and feedback strategies for remediation.

Recommendations for Enhancements to the GIFT Instructional Model

The enhanced EMAP affords GIFT capabilities to select and sequence content in a logical fashion, and considers engagement behaviors through the ICAP framework. The framework extends across domains, but the configurations are specific to the AIS under development. Much of the decision making logic is based upon a metadata tagging schema that associates content with learner states or pedagogical configurations specified by the AIS author. While this is simpler than developing strategies from scratch for every AIS developed, it does contribute to the workload of the author and is only usable for the AIS under development. Alternatively, more complex learner models could provide learner states that can be used to drive automated selection of instructional sequencing and adaptations (e.g., scaffolding).

Tutorial planning includes macro-level adaptations (e.g., decisions on next pieces of content to present) and micro-level (e.g., delivering tailored hints/prompts about current problems/scenarios). A computational model that governs the frequency, type, and mode of interaction with learners would go a long way to reducing the authoring workload. Rowe et al. (2015) noted several drawbacks associated with tutorial planner development: 1) labor intensive knowledge engineering processes, 2) static models that do not improve with experience, 3) static models that do not represent uncertainty well. Rowe and colleagues have been pursuing a Markov decision process (MDP) approach for GIFT over the last few years, but bootstrapping or initializing these models has been a limitation of applying this approach in a large number of instructional domains. We are recommending the development of simulated students to initialize MDP approaches and further reduce workload associated with creating complex, self-improving decision processes in GIFT.

Applying Cognitive Modeling in CAISs

According to the Oxford English Dictionary (2019), cognition is defined as "the mental action or process of acquiring knowledge and understanding through thought, experience, and the senses". Thought, experience and senses neatly align with learning theories discussed earlier: cognitivism, constructivism, and behavioralism respectively. Cognition includes mental functions and processes such as understanding, attention, knowledge acquisition and generation, memory, perception, judgment, reasoning, problem solving and decision making, and language generation. Cognitive processes use existing knowledge to construct and deconstruct mental models and thereby generate new knowledge.

A cognitive model is "an approximation to animal cognitive processes (predominantly human) for the purposes of comprehension and prediction" (Wikipedia, cognitive model, 2019) and are usually developed using a cognitive architecture. Cognitive models are generally focused on a single cognitive process (e.g., image recognition). Cognitive architectures are generally focused on the structural properties of a system, and help constrain the development of cognitive models. According to Laird (2012) prototypical cognitive architectures have memory (short-term symbolic, long-term declarative, and long-term procedural) and processes to simulate perception, learning (e.g., declarative, procedural, conceptual), action selection, and action execution. Goals are also represented in the symbolic short-term memory. There are several cognitive architectures, but three popular examples include Soar (Laird, 2012), Sigma (Rosenbloom, 2013) and ACT-R (Anderson, 1996).

So what is the relevance of cognitive modeling to AIS and specifically GIFT? Currently GIFT represents few of the cognitive functions and processes which might be used to enhance its instructional decision-making capabilities. However, the open question is whether the benefits associated with developing and integrating cognitive models within GIFT outweigh the workload. Effectiveness studies should be undertaken to answer this question or energy should be invested to provide a simplified methodology to author perception-decision-action processes in GIFT that can be generalized across domains. An additional area in which cognitive models might be useful are in GIFT's learner module. A representation of the learner's cognitive processes, capabilities, and limitations might enable GIFT to paint a more comprehensive picture of the learner and better tailor learning experiences.

As noted earlier, the National Research Council (2000) identified important principles for structuring learning experiences. There are dozens of instructional strategies from rules of thumb to modeling the habits of expert instructors (Lepper, Drake & O'Donnell, 1997). A few common strategies that lend themselves to a variety of domains follow:

- Set goals and objectives – what do you want the learner to accomplish during their adaptive experience and how will the AIS measure it

- Tailoring instruction – this is the principle which GIFT is built upon; adapt instruction to the needs, goals, and preferences of the individual or team

- Practice – provide opportunities for practice during adaptive instruction

- Concept mapping - visual representations of functions, ideas, facts, tenets, concepts, and terms create by learners to organize their knowledge

- Summarizing and note taking – paraphrase and reflect on concepts to be learned; take notes to capture essential information to help support recall

It is important to consider mechanisms (e.g., widgets) in the adaptive instruction that support the desired behaviors associated with each strategy. The design of these mechanisms could be improved over time by capturing data about their use.

Understanding Environmental Conditions and Context in CAISs

Finally, we are ready to explore the third component of adaptive instruction, the environment. The environment encompasses all the objectives, content, feedback, interactions, conditions, constraints, misconceptions, policies, and other data that is specific to a domain of instruction. All this knowledge provides the context that is necessary for selection of appropriate, relevant actions by the AIS.

Figure 1 shows the execution and constant evaluation of policies as part of the agent-based tutor or AIS. Policies are rules or triggers for actions by the AIS. Policies may be generalized across domains (domain-independent) or be specific to a domain (domain-dependent). Policies are updated as their use is analyzed. A policy applied under a specific set of conditions yields an outcome. Over a large number of experiences, there may be recognition that the policy applied either increased, decreased or had no effect under the conditions of the learner or the environment. It could be that the performance of a policy is so consistent that it becomes a rule. For example, a policy that selects math problems based on difficulty and learner experience in the domain of instruction. More difficult problems are assigned to more experienced learners. However, it might be less obvious what the policy should be if it is a policy about modifying the difficulty of a scenario based on age of the learner and the environmental conditions. Usually, there is some degree of uncertainty associated with environmental conditions and the application of appropriate policies. This uncertainty is generally not fully represented in AIS architectures like GIFT. A greater understanding of the influence of conditions and elements in the AIS environment could be developed through real-time machine learning techniques armed with sufficient data. This would also result in more effective decision-making more in line with the expected performance of a CAIS.

Summary of Recommendations and Future Research

We began by discussing complexity theory and relating this theory to AISs to make a case for something we labeled a CAIS. In an effort to demonstrate, the complexity of AISs, we reviewed and extended discussion about the LEM and the interactions between agent-based tutor, learner(s), and the environment that the learner experiences during adaptive instruction. Throughout this chapter, we discussed the view that CAIS decisions are made largely in the presence of uncertainty. We also stressed throughout the need for embedded, automated processes to manage the high level of complexity and uncertainty that are naturally occurring in CAISs and their components – the learner, the tutor, and the environment.

In an effort to expand the complexity and improve the effectiveness of CAIS instructional decisions, we made several recommendations related to expanding the *variables of interest* within CAIS learner models. An important element of understanding learning is its relationship with perception. In our opinion, perception is not adequately represented in AISs today. We posit that by tracking more about the learner, the CAIS can make more effective tailoring decisions during adaptive instruction, and by automating most (if not all) of these decisions, we can drastically reduce the workload of CAIS authors. We also suggested that cognitive models closely related to the cognitive capabilities and limitations of the learner might also be implemented in order to better understand the impact of CAIS recommendations.

As part of our examination of adaptive instructional processes, we also suggested that the use of cognitive models to represent the decision making functions of the CAIS might improve the effectiveness of instructional decisions in much the same way that chess players understand how their current moves impact future options. Cognitive models might also allow easier implementation of common instructional strategies (e.g., goalsetting, practicing, concept mapping, and note taking) to be generalized across GIFT domains.

While we understand the need to simplify the CAIS authoring process, and laud the implementation of two instructional frameworks in GIFT – CDT and ICAP, we also recommend further research and efforts to automate the decisions associated with selection of recommendations, strategies, and tactics in GIFT. Understanding the relationship between learner and environmental conditions and CAIS options will help quantify the influence of those variables and allow more automated processes for adaptive instruction.

We end this chapter by noting that complexity is not necessarily bad. It can enable us to more fully represent the nuances and inner workings of adaptive instruction while still reducing authoring workload through automation by taking advantage of machine learning processes.

References

Anderson, J. R. (1996). ACT: A simple theory of complex cognition. *American Psychologist*, 51(4), 355.

Atkinson, R. C., & Shiffrin, R. M. (1968). Human memory: A proposed system and its control processes1. In Psychology of learning and motivation (Vol. 2, pp. 89-195). Academic Press.

Baddeley, A. (1992). Working memory. *Science*, *255*(5044), 556-559.

Bar-Yam, Y. (2002). General features of complex systems. Encyclopedia of Life Support Systems (EOLSS), UNESCO, EOLSS Publishers, Oxford, UK, 1.

Bassok, M. & Novick, L. (2012). Problem Solving. In Oxford Handbook of Thinking and Reasoning. New York, NY: Oxford University Press.

Card, S. K., Mackinlay, J. D., & Schneiderman, B., (1999). Readings in information visualizations: Using vision to think. San Francisco, CA: Morgan Kaufmann Publishers.

Chi, M. T. (2009). Active-constructive-interactive: A conceptual framework for differentiating learning activities. *Topics in Cognitive Science*, *1*(1), 73-105.

Dunker, K. (1945). On problem-solving. (L.S. Lees, Trans.) Psychological Monograph, 58 Whole No. 270). (Original work published in 1935).

Eaton, M. (2012). Applied IDT: Instructional Design & Theory. Retrieved from: http://www.appliedidt.com/blog/2013/07/03/instructional-theory-versus-learning-theory/

Hayes, J.R., & Simon, H.A. (1977). Psychological differences among problem isomorphs. In N.J. Castellan, D.B. Pisoni, & G. R. Potts (Eds.), Cognitive theory (2): 21-44. Hillsdale, NJ: Erlbaum.

Johnson-Laird, PN. (2012). Inference with mental models. In Oxford Handbook of Thinking and Reasoning. New York, NY: Oxford University Press.

Jung, C. G. (1932). Psychological types. London: Pantheon Books.

Laird, J. E. (2012). *The Soar cognitive architecture*. MIT press.

Lakoff, G. & Johnson, M. (1980). Metaphors we live by. Chicago, IL: University of Chicago Press.

Lepper, M. R., Drake, M. & O'Donnell-Johnson, T. M. (1997). Scaffolding techniques of expert human tutors. In K. Hogan & M. Pressley (Eds), Scaffolding learner learning: Instructional approaches and issues (pp. 108-144). New York: Brookline Books.

Merrill, D. M. (1983). The Descriptive Component Display Theory. In C. M. Reigeluth, *Instructional Design Theories and Models: An Overview of their Current States* (p. 360). Routledge; 1 edition (August 1, 1987).

Miller, John H., and Scott E. Page (2007-01-01). *Complex adaptive systems: an introduction to computational models of social life*. Princeton University Press. ISBN 9781400835522.

National Research Council. (2000). *How people learn: Brain, mind, experience, and school: Expanded edition*. National Academies Press.

Oxford English Dictionary. (2019). Cognition. Retrieved from: https://en.oxforddictionaries.com/definition/cognition.

Rosenbloom, P. S. (2013). The Sigma cognitive architecture and system. AISB Quarterly, 136, 4-13.

Rowe, J., Mott, B., Lester, J., Pokorny, B., Peng, W., & Goldberg, B. (2015, August). Toward a modular reinforcement learning framework for tutorial planning in GIFT. In Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym3) (p. 89).

Sottilare, R. Considerations in the Development of an Ontology for a Generalized Intelligent Framework for Tutoring. International Defense & Homeland Security Simulation Workshop in Proceedings of the I3M Conference. Vienna, Austria, September 2012.

Sottilare, R., & Brawner, K. (2018, June). Component Interaction within the Generalized Intelligent Framework for Tutoring (GIFT) as a Model for Adaptive Instructional System Standards. In *the Adaptive Instructional System (AIS) Standards Workshop of the 14th International Conference of the Intelligent Tutoring Systems (ITS) Conference, Montreal, Quebec, Canada*.

Sottilare, R. A., Brawner, K. W., Goldberg, B. S., & Holden, H. K. (2012). The generalized intelligent framework for tutoring (GIFT). Orlando, FL: *US Army Research Laboratory–Human Research & Engineering Directorate (ARL-HRED)*.

Sottilare, R. A., Brawner, K. W., Sinatra, A. M., & Johnston, J. H. (2017). An updated concept for a Generalized Intelligent Framework for Tutoring (GIFT). Orlando, FL: *US Army Research Laboratory–Human Research & Engineering Directorate (ARL-HRED)*.

Sottilare, R. A., Burke, C. S., Salas, E., Sinatra, A. M., Johnston, J. H., & Gilbert, S. B. (2018). Designing adaptive instruction for teams: A meta-analysis. *International Journal of Artificial Intelligence in Education*, 28(2), 225-264.

Wikipedia. (2019). Cognitive Model. Retrieved from: https://en.wikipedia.org/wiki/Cognitive_model.

Chapter 5 – Toward Self-Improving Multi-Agent Tutoring Architectures: Progress and Future Directions

Benjamin D. Nye[1], Raj Thaker[1], Nitin Surana[1], Daniel Auerbach[1], and Keith Brawner[2]
[1]University of Southern California, Institute for Creative Technologies,
[2]U.S. Army Combat Capabilities Development Command – Soldier Center – Simulation and Training Technology Center

Introduction

In an ideal world learning technologies should slowly improve over time, by leveraging data from prior learner sessions to help improve the content that exists: either by changing the content or by prioritizing content that appears to be more effective. Likewise, components of intelligent systems should become more effective: either by directly improving due to new data or by intelligently selecting between different services that provide the same capabilities (e.g., providing hints, estimating learner knowledge). However, in practice such capabilities are challenging to develop: systems often require different data, longitudinal data needed to meaningfully improve a system might not be available (e.g., performance on later learning tasks after the current lessons), and protocols to improve system components are still being developed. This means that improvements to systems typically require a human in the loop who reviews, revises, and publishes changes (e.g., see the Cognitive Tutor; Ritter, 2015).

This chapter describes how we have approached these problems through the Multi-Agent Architecture (MAA) effort for the Generalized Intelligent Framework for Tutoring (GIFT; Nye, Auerbach, Mehta, & Hartholt, 2017). The GIFT MAA project has added an experimental module to GIFT called the AgentContainer, which can be used to register and instantiate arbitrary services that are associated with a specific course (Nye et al., 2017). These services communicate using messages and from the outside are a black box: they could be either purely reactive or could be intelligent. A guideline for these services is that they should be fault-tolerant, doing their best to respond to messages based on the information that they have available. As such, the GIFT MAA agents framework offers a possible testbed for exploring self-improving systems. This chapter reports on the results of a usability study conducted to look at the ease-of-use for adding such services to a framework such as GIFT for new users.

In addition to conducting a usability study, the framework has added two relevant capabilities since its initial architecture was published (Nye et al., 2017). The first capability is a generic ability to connect to external services (e.g., REST), so that GIFT can coordinate or learn from systems outside of its runtime environment. The second capability is the addition of "proposal pattern" capabilities, in which a default process exists for identifying and selecting between different services that might offer the same information or capabilities (i.e., same message signatures). The process of building these capabilities has given insight into some of the advantages and limitations of the current approach for building a multi-agent architecture for tutoring more generally, informing the overall question of the long-term goal of building a lightweight architecture for intelligent tutoring systems (ITSs; Nye & Morrison, 2013).

Methods

The overall approach to registering and communicating between agents, the Multi-Agent-Architecture project, is described in Nye et al. (2017). These agents leverage the SuperGLU framework (https://github.com/GeneralizedLearningUtilities/SuperGLU) in which all services and agents are looselycoupled and do not directly connect to each other. Instead, all services and agents connect to "gateway" nodes which handle relaying messages to appropriate listeners. As such, services do not directly transmit messages to any other service. Instead, they broadcast messages (fan-out by default) to all other services in the graph. The messages will then be filtered either by the receiving services or by gateways if they appear to be not useful to the receiving services. In this way, services do not need to know about the topography of the network for services and do not need to rely on hard-coded bindings to any particular service location.

The result is an acyclic undirected graph where each service/agent connects to one (and only one) gateway but gateways might connect to multiple other gateways or services. In the GIFT MAA project, a special module was added (AgentContainer) where course-specific services could be registered and initialized when a course starts (and subsequently updated during course activities, though this functionality has not been used in practice). Figure 1 (below) shows an example of this type of structure which was used for the usability studies described next. A Websocket gateway (GIFT_SuperGLU HTML5 Gateway) enables listening or transmitting to a SuperGLU-compliant webpage via a Websocket frame which receives messages via HTML5 postMessage (on the client side). A second gateway (GIFT_VHT_Converter) exists to convert messages from GIFT format to a Virtual Human Toolkit (VHT) format consumed by a VHT Tutor Controller. Both of these gateways connect to an ActiveMQ gateway that ties into the GIFT main ActiveMQ service. In this way, GIFT modules can communicate with services that run directly in-memory inside the AgentContainer (e.g., VHT Tutor Controller) and also with services accessible through gateways that connect to remote services (e.g., an HTML Practice page in a browser reached via the GIFT_SuperGLU HTML5 Gateway).

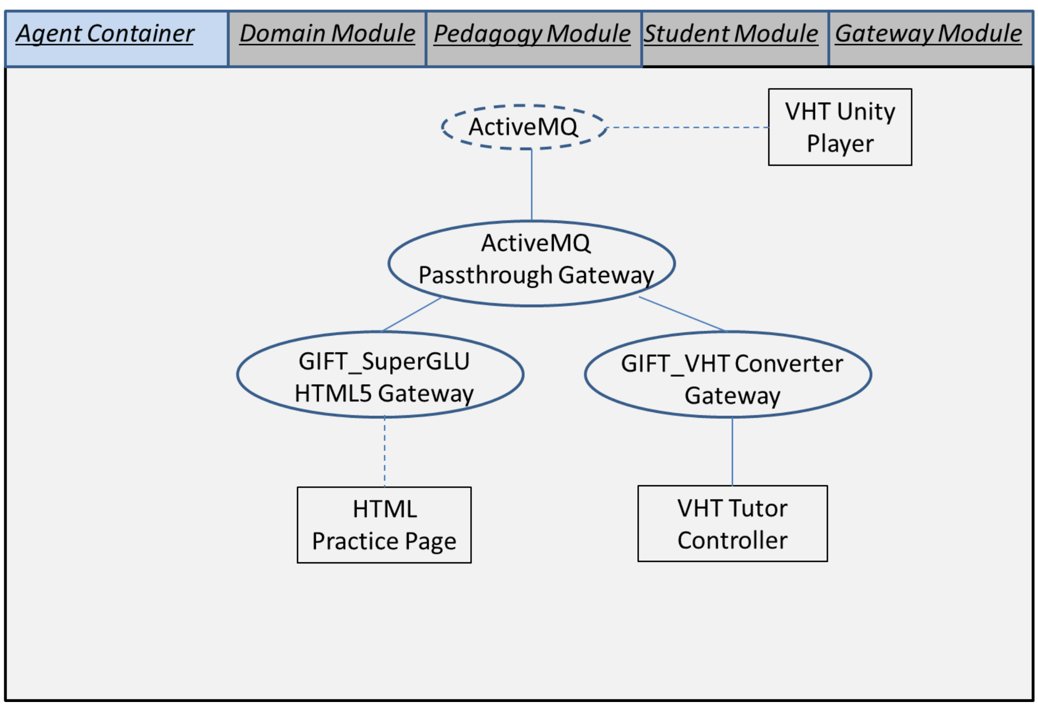


Figure 1: Agent Container Diagram for Usability Study

Since the original paper describing this work, new functionalities have been added for configuration to the GIFT MAA branch. Specifically, an HTML configuration tool exists that enables building the graph of services and gateways that a course should have. This tool outputs a JSON configuration file that is associated with the service. Additionally, interfaces have been added to GIFT to enable agents for the course by checking off boxes that can a) activate the AgentsContainer module (initializing all services and gateways in the AgentsConfig.json file) and b) enable specific webpages embedded in a GIFT course to send messages to a Websocket listener gateway (if one exists). Figure 2 shows the AgentsContainer configuration authoring tool. Figure 3 shows flag for enabling the Websocket listener (SocketIOGateway).

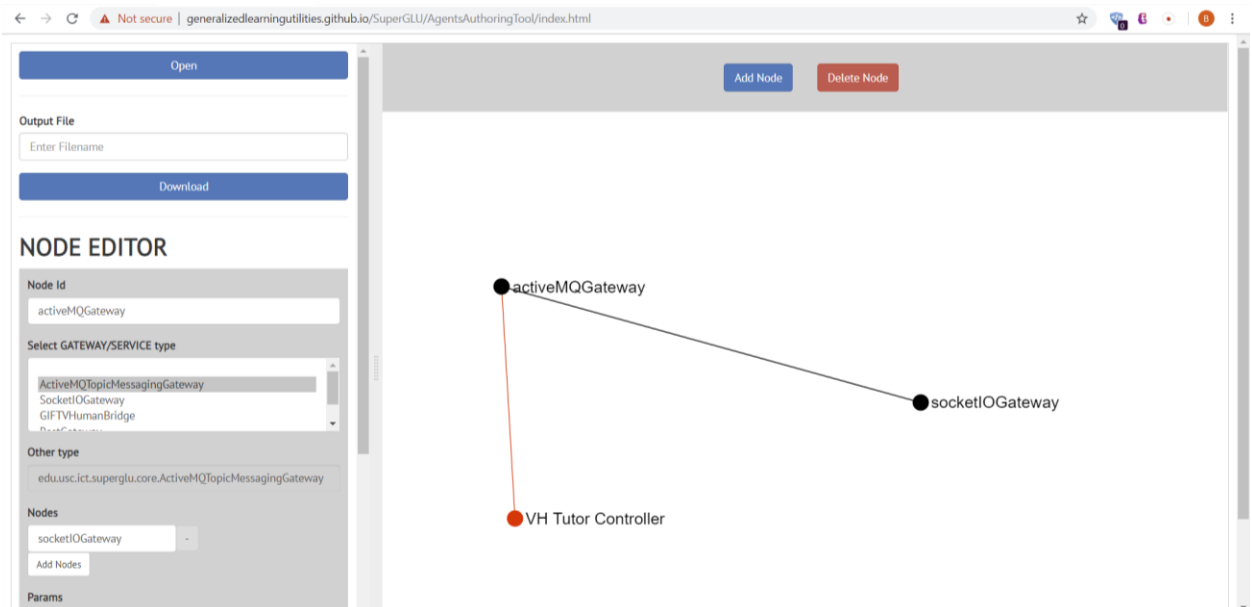


Figure 2: Agents Configuration Tool to Edit the Service Graph for AgentContainer Module

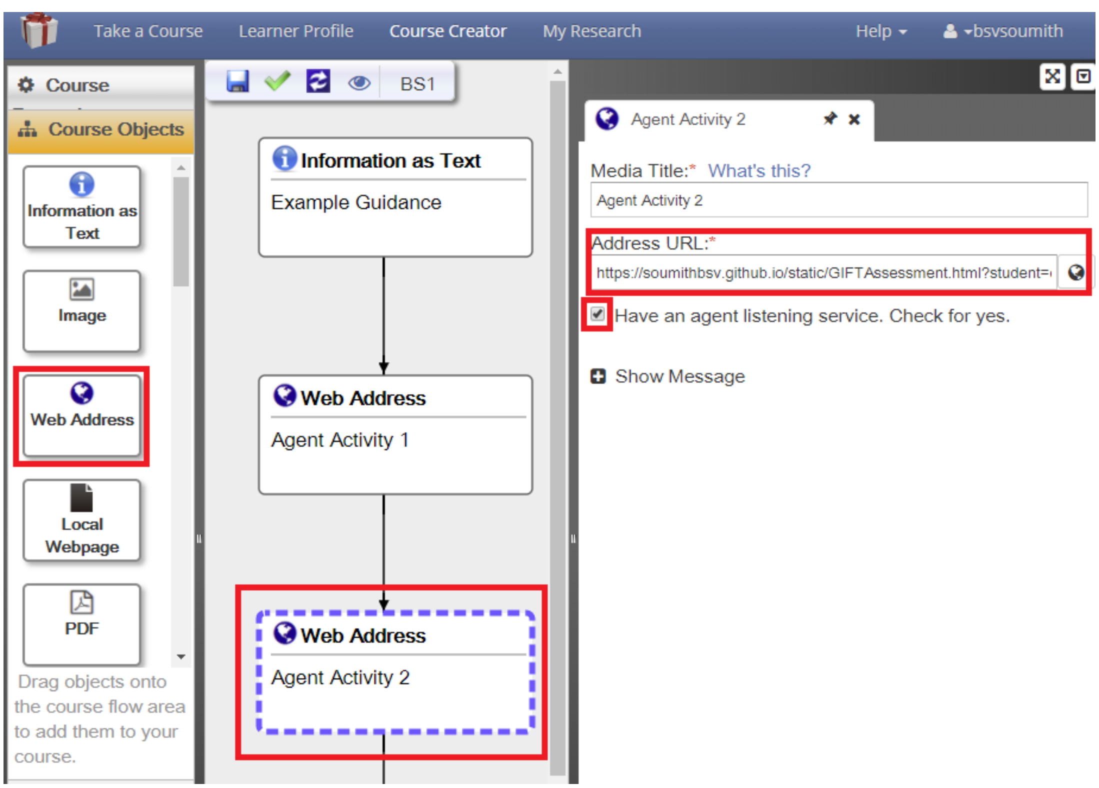


Figure 3: Enabling PostMessage-to-Websocket Agent Listener for Web Page

Design: A series of internal usability trials were conducted to evaluate the ease of leveraging Multi Agent Capabilities that were designed in a GIFT branch. In particular, two types of capabilities were added. The first was integrating Virtual Humans as talking heads for courses, which could be enabled. The second was adding a general “agent container” which could be used to spin up course-specific Java modules that either run agents/services in the local Java app or that are connectors to externally hosted services. The use-case for testing this container was the ability to have any external web page in GIFT be able to send performance scores that were received by the GIFT learner module and used to adaptively present different content. A tutorial was designed that covered these different use cases.

The trials were formative with users following a tutorial and completing a post-survey at the end. There were seven tutorial steps, as shown in Table 1.

Table 1: Usability Trial Tutorial Steps

Phase	Description
P1. Registration	Making an account to run GIFT and logging in.
P2. Running GIFT	Running the GIFT program (note: we mostly just started it running for them to save time in later rounds, since it was just running installers).
P3. Creating Agents-Enabled Course	Creating a new course with an adaptive course element that would respond to some performance on a web page.
P4. Running Course	Running the course and testing it out.
P5. Custom Course-VHT Talking Head	Modifying the agent configuration file to use the agent’s talking head (note: this is no longer required to get talking heads, so deprecated. As such, summer feedback and ratings on these are no longer as accurate).
P6. Run Custom Course	Running the modified course to see the talking heads.
P7. Substitute Custom Web Page	Replacing the web page providing performance data with a modified version of the web page that provides scores.

System: The system was mostly the same for about half the participants, but changed substantially before the last set of participants. Overall, due to changes between the Spring and Summer, tutorial steps 4 and 5 are not particularly comparable. Talking heads no longer needed that setup step (since they were drop-in replacements for Media Semantics) and there were also some issues with talking heads not working due to the course being run in Internet Explorer. The survey and protocol was not changed since we wanted to be able to analyze the data coherently, but in retrospect this may have been a mistake for those steps (where users hit some significant issues).

Participants: The users were a convenience sample of students who were new to the Institute for Creative Technologies (ICT) (mostly in Learning Sciences) but who had never used GIFT before. So these ranged from one student programmer hired to work on GIFT MAA who had not tried it yet, to students who were hired to work on other projects and donated a couple of hours to do this, to a set of six visiting students who were visiting ICT this summer. Participants can be thought of in three blocks: Winter (3 participants in December-January), Spring (3 in March-May), and Summer (6 in June). The summer participants were all 3-week visiting students, which meant they had a much more varied skill set: the prior participants were all Computer Science majors, while only some Summer students had any programming skills. This meant that they were not the ideal participants for the later parts of the experiment, which became fairly technical (e.g., modifying a web page). One participant in the Spring cohort only filled out a subset of the survey for some reason, or experienced data loss (oddly enough, had text fields input but no ratings for system components).

Results

The high level variables of interest were the time it takes to set up a course with agents and the usability and liking ratings of the functionality involved. The total time to create a course from scratch and to modify it with a new web page and to enable virtual agents was under 2 hours. Table 2 presents the time spent by each cohort, as well as their mean ratings on a scale of Completely Disagree (1) to Completely Agree (6) that the system was easy and useful. Figure 4 which adds up self-reports by section for time, put the estimates at about 91 minutes (67 for Winter, 90 for Spring, 105 for Summer). As such, the amount of time to make a first time mini course based on the tutorial was fairly reasonable, particularly since a significant amount of time was spent setting up the basic GIFT course.

Table 2: Overall Self-Reported Time in Tutorial and Avg. Ratings of GIFT vs. Agents

OVERALL	N	Time (in h)	GIFT	Agents
ALL	12	1.7	3.4	3.5
WINTER COHORT	3	1.7	3.3	3.5
SPRING COHORT	3	1.0	5	5.3
SUMMER COHORT	6	2.0	2.7	2.3

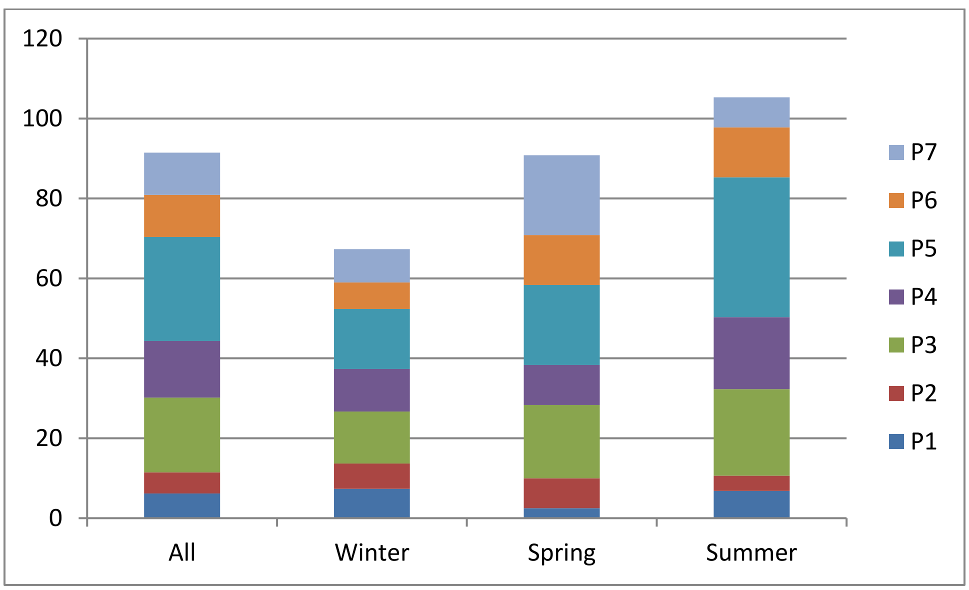


Figure 4: Self-Reported Time per Phase of Tutorial (by Cohort, in Minutes)

User ratings are a bit more difficult to interpret. While different cohorts had very different ratings in some cases (summer students were quite negative), the implications and trends for these ratings are more subtle. First, users seemed to not be able to rate the agents capabilities very distinctly from GIFT authoring and courseware itself, in this context. This is unsurprising, as they had no initial familiarity with either, and the tasks required for each were intermixed into a single tutorial (e.g., to get meaningful agents, you needed to build a survey, add an adaptive module, etc.). In Table 2, there are some minor deviations that students who would be starting research at ICT rated agents subtly more highly, while summer students rated them slightly lower, but these are very minor differences compared to the overall magnitude of difference between cohorts about both areas overall. Likewise, while there were average underlying questions (e.g., about ease of use, liking, usefulness expectancy), those questions tended to be fairly well anchored to each other (e.g., if one was high, they all tended to be fairly high).

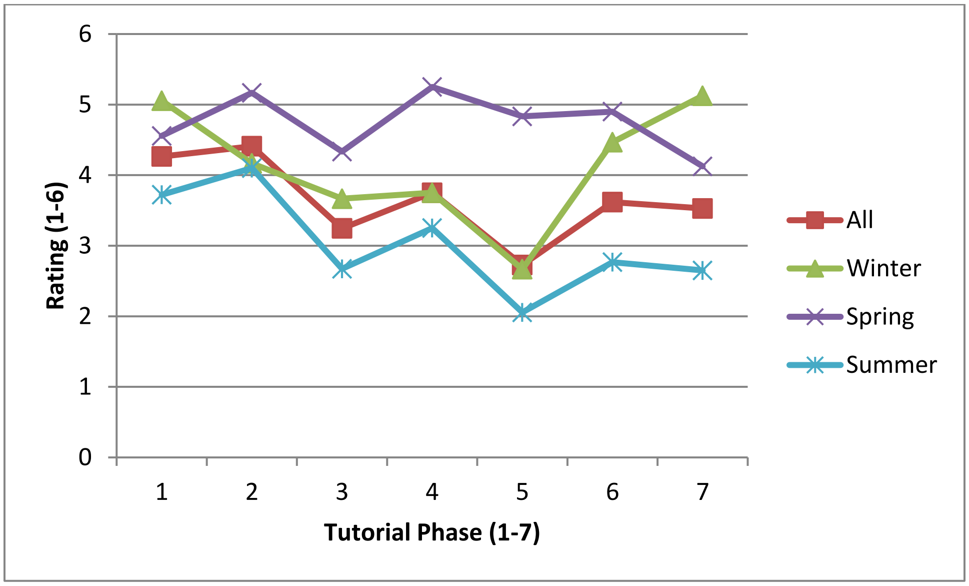


Figure 5: Self-Reported Overall Ratings of GIFT by Phase

With that said, looking at the ratings for different parts of the system are more useful (see Figure 5). These trends show that Spring students tended to be fairly universally positive across all phases, while Winter students found problems in Phases 4 and 5, respectively (e.g., running a GIFT course and modifying the agents container JSON file). The Spring students likely rated these items higher due to bug fixes done to make agents more stable in between that period, as well as the addition of a visual web-based tool to configure agents. Finally, one winter user was pretty harsh on the overall UI feel for GIFT (did not like the dual-pane design, did not like the feel, etc.)

Among the summer students, we see that ratings are pretty similar for basic setup (Phase 1+2), but crater when actually using the system. To a significant degree, most of them simply found it hard to author and understand the system (e.g., felt that it took a lot of steps to set things up). This was not universal (one student liked it), but was a strong feeling. This negative attitude was amplified by bugs with the virtual humans not playing in Internet Explorer which some hit (which was new since the more stable Spring build), and which is likely an issue that will need to be confronted directly in terms of user expectations. Finally, they were generally less technical and struggled to modify the web page for the final steps. In general, while they were not the expected user population to do that kind of customization, this strongly reinforced that even the graphical configuration tool for setting up services was pretty alien to this cohort.

For the agents themselves, attitudes were mixed. Some people liked them and appreciated the guidance. Others felt they did not necessarily need them. This is a pretty common attitude, so it is not surprising. It appears reinforced however by any incidences of bugs or slow-downs (e.g., agents are okay so long as they are not in the way, but as soon as they cause issues, people move to dump them). Particularly for a proof of concept course like the one from the tutorial, this is to be expected, but this would probably be a common attitude among any real instructors or courseware builders as well.

Discussion

The results of the usability study indicate that it is possible for users to fairly quickly activate and configure agent-based services in a system such as GIFT. However, the usability ratings also indicate that this process is non-trivial and particularly challenging to users without computer science skills. Even without programming, the concepts of configuring graphs of services is not intuitive to a typical course designer. As such, typical users (e.g., content experts rather than computer science experts) are likely best-off having a set of agent configuration templates to choose from that represent common use cases (with manual configuration only performed by advanced users). Additionally, these studies did not demonstrate the ability of the agents to self-improve over time. While it is possible to plug agents into a course (and these agents might have the ability to apply machine learning to improve their performance), the practical realities for a generalized tutoring framework are not as simple.

From a practical standpoint, a few issues must be addressed before plug-and-play self-improving services can be deployed for learning technologies. The primary issues which are of interest for ongoing follow-up work are noted below:

- *Service Availability:* Where is the service? What happens if the service is unavailable?
- *Persistence:* How is data stored to enable self-improvement? How do we determine what data streams are comparable (e.g., can be analyzed together to improve future ones)?
- *Rewards/Feedback:* How can we establish reliable improvement metrics?

Service Availability

The work done thus far as a follow-up has focused mostly in service availability. This has been approached from two angles: re-use of existing (non-SuperGLU) services and graceful fail-soft policies to enable choosing between services or connecting to alternate services when needed.

REST Messenger Service: First, service gateways and connectors were developed to assist connecting to services beyond the current execution context. As noted, gateways can connect to any arbitrary other gateway (via websockets, ActiveMQ, HTML5 postMessage, etc.). However, there will always be web services that are not built to connect via specialized gateways and which utilize simple standards. As a result, a REST Messenger service wrapper was developed that enables SuperGLU services in GIFT to describe the API of a REST service which plugs into the gateway/service graph.

The REST Messenger service accepts messages of type RESTMessage. The RESTMessage has four parameters that define it: The HTTP request verb, the destination URL, an optional StorageToken for JSON bodies, and a list of header key/value pairs. When the REST Messenger receives a RESTMessage, it converts it into an HTTP request and sends it to the specified service. The body of the response it gets back from WebService is again packaged into a new RESTMessage and sent back to the network of various connected SuperGLU services.

Proposal Pattern (Selecting Services): A second behavior that is important for self-improving services is the ability to choose between different possible services. Modern Artificial Intelligence (AI) and machine learning depend heavily on ensemble models, with varied data. However, not all services are consistently available and some services may be preferred over others (e.g., speech-to-text is best done with large cloud services, but can be done locally on smartphones if lower quality is acceptable). Learning to select between different services and sources of information is an important step for any self-improving service.

To address this problem, the GIFT MAA project has studied proposal patterns for agents adapted from standards-based approaches to this problem (FIPA, 2013). In a Proposal Pattern for GIFT MAA, the Sender service proposes a contract to the Responder services that states a) the template for the request messages that it will send and b) the template of the response message that it expects. In these patterns, Sender and Responder are arbitrary roles, determined only by their behavior in making proposals or responding to certain proposals that they receive (as opposed to ignoring them). In essence, the proposal allows services to agree on a request-response communication pattern before sending the main request. When a proposal is sent, services that could fulfill the pattern reply with Acceptance messages and then one is Confirmed to complete the three-way-handshake. In this framework, a three-way-handshake is important because accepted proposals can still be arbitrated by the Sender (e.g., accept one, accept none, accept all). As shown in Figure 6, the types of messages involved are:

1. *Proposal Request:* Describes the Proposed Message template and the Proposed Message Acknowledgement template.
2. *Proposal Acceptance:* Describes a Responder service that is willing to handle Proposed Messages.
3. *Proposal Confirmation:* Accepts a Responder who should reply to a Proposed Message. Multiple can be sent (e.g., can solicit replies from many or all Responders). Responders can also still reply to Proposed Message, but by default the Sender will ignore non-confirmed Responders.
4. *Proposed Message:* A template for the message that the Sender will transmit after a proposal is confirmed (potentially sending it many times). This template determines the allowed values for different message fields.
5. *Proposed Message Acknowledgement:* A template for the message the Sender expects in response to the Proposed Message. This represents the data or update signals that the Sender is seeking out.

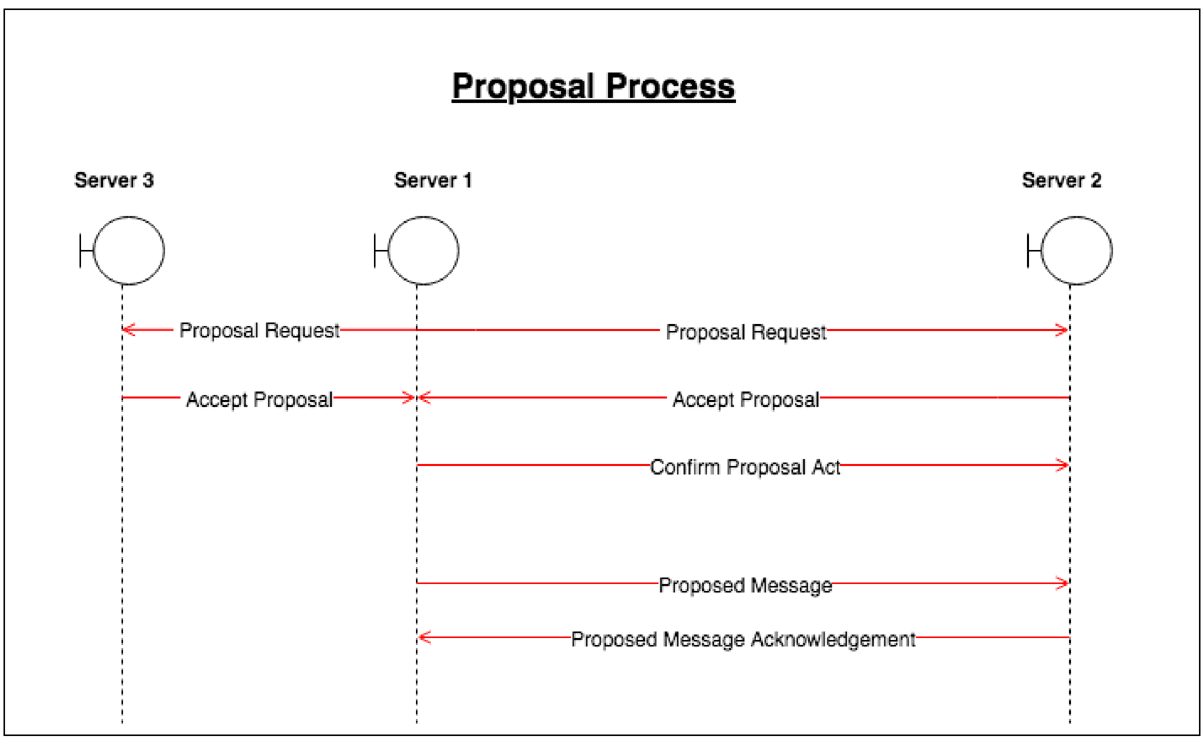


Figure 6: Proposal Process for Three-Way Handshake and One Cycle of Proposed Message/Acknowledgement

This pattern is a simplified version of the general case of Proposal Patterns, which can be quite complex for the general case (e.g., requiring description logics or often detailed constructor/factory design patterns to build a single proposal). In the Selection process of pairing a Responder to a Sender, the Proposal Request, Acceptance, and Confirmation are the message types that drive the interaction. They form the key components of the traditional three-way handshake protocol to decide which responder it would be choosing to move ahead. Proposed Message and Proposed Message Acknowledgement are the message types that are used to describe the proposal contract.

Initially, the Sender broadcasts the Proposal Request Message to all the Responders available to it. The Message consists of the format/type it will be expecting in the response of the main query message sent later. All the Responders who have the capability to comply with the contract may respond with a Proposal Acknowledgement Message. The Sender then can apply different rule-sets (either default or ad-hoc) to select a Responder (or multiple Responders) which it will Confirm. After Responders are confirmed, the Sender will at some points send a valid Proposed Message which is intended for specific Responder(s). This Proposed Message is an instance that matches the template used in the original Proposal. The Responder finally sends a message that matches the Proposed Message Reply template to the Sender. As one final note, while this discussion treats these communications as point-to-point, the communication in general is still relayed using service gateways and the messages transmitted might be handled, stored, or analyzed by other connected services.

The process of selecting a proposal is part of the self-improving system. In this research, we have so far designed pragmatic default strategies for accepting proposals when multiple exist: first-past-the-post (accept the first one), accept all (receive and handle many responses), deprioritize failures (prefer services with fewer timeouts), and ad hoc selection between Responder bids (e.g., collect bids over a time span and pick the service which is most preferred). These strategies are simple, but provide some rudimentary learning capabilities. In particular, deprioritizing failures means that previously-confirmed services that did not respond promptly can be reduced in priority when soliciting new bids. This general, simple network strategy offers some basic self-improving adaptation.

Proposal Pattern (Fail-Soft): In practice in a multi-agent system, there are also many possibilities where the message transmissions fail. In such cases, many Fail-Soft Strategies can be put in place to deal with it. These strategies are all controlled by configurable parameters outlined in the Proposal Message. The purpose of this pattern is to enable a Sender service that previously confirmed a Responder to either retry a message or to cancel the contract so that it can identify a different Responder to confirm (if any is available). Three fail-soft criteria can be configured for the initial proposal three-way-handshake:

1. *Number of Proposal Attempts*: In case the transmission of the Proposal Message fails or no response is received, this parameter defined the number of times the system will retry.
2. *Timeout for Proposal Acceptance*: The time that the Sender will wait to receive the Proposal Acceptance Messages before deciding which one(s) to confirm.
    a. *Fixed Duration*: Reviews Acceptances received within the given number of seconds,
    b. *Open Call*: Keeps on Accepting and decides to Confirm Responders based on acceptance criteria (e.g., First-Past-the-Post, Prioritization, Accept All).
3. *Prioritized Proposal:* A special version of the timeout approach is to accept proposals based on prioritization criteria. The specific prioritization logic can be customized by a service. This could mean an early termination of a proposal (e.g., highest-priority service accepted). The current implementation uses a deprioritization approach to prefer acceptances by Responders that have not previously triggered a fail-soft event that canceled an earlier proposal.

The behavior for an individual Proposed Message also contains fail-soft behavior, to enable restarting the Proposal pattern in case of non-performance by a Responder.

1. *# of Proposed Message Attempts*: If the Proposed Message fails or no response is received (timeout), this parameter defined the number of times the system will retry before canceling the proposal contract.
2. *Timeout for Proposed Acknowledgment*: The cutoff time to determine a non-response
3. *Fail Soft Trigger*: The criteria applied to determine when to void the contract:
    a. Attempt Count: Break a proposal when too many failures occur overall or in a row
    b. Timeout: Break a proposal when only failures have occurred for some amount of time.

The default behavior for GIFT MAA services after a Fail Soft event has been triggered is to cancel the old proposal and start a new Proposal Message. However, this process is not memoryless: the old failed service problems are stored in the session data. These are used to re-propose with deprioritization (i.e., sorting) based on the number of prior failures that resulted in triggering a fail-soft event. The prioritization helps in the selection of the Responder after receiving a bulk of Proposal Acceptances, since alternate services can be selected that may be more reliable. Other types of Quality of Service/Quality of Experience metrics could also be used for this purpose to choose alternate services based on learning outcomes or other metrics.

Persistence

At present, ongoing work has not developed an ideal model for persistence in a self-improving, multi-agent learning architecture yet. The general problem is due to the competing demands of efficient, real-time data access versus well-defined service boundaries and encapsulation. Thus far, the most promising model for persistent data that supports self-improvement requires a three-pronged approach:

1. *Canonical Data:* A data set that represents the authoritative data for learning events related to the user and system (e.g., such as an xAPI learning record store). This data set should not change (largely immutable) but is not suitable for real-time analytics.
2. *Data Views:* Calculated views of the canonical data are generated that are sufficient for each service to use. These views are used to maintain data on examples and analytics that help past events improve the system for future users. These exist to prevent expensive calculations on arbitrary amounts of raw data for real-time application needs.
3. *Context-Specific Requests:* Messages indicate which specific criteria/subsamples that the data views need to help each service to respond to the ongoing needs of other services (e.g., recommending a class activity vs. an at-home one). These will often be session-based, so they transmit a level of temporal and activity context with persistent state from other services.

An implicit requirement for this approach to work is that two criteria are followed. First, all critical services need to contribute to the central canonical data store in order to have substantial shared knowledge. Second, the data views should be persistent but must be possible to rebuild from the canonical data set at any point. From a practical standpoint, they should also generally be modified using incremental update functions (e.g., notified about new data to the canonical set which triggers an update to the view). A form of this approach has been followed by a related project in the same group called the Personal Assistant for Life-Long Learning (PAL3; Swartout et al., 2016). This approach has been particularly effective for dealing with services that require an offline mode, in which canonical data can be synchronized between a local device and backend services (single main point of synchronization) but services can update their data views accordingly based on the canonical data. In this way, it is less direct than a whiteboard architecture (e.g., where all services share and act on one canonical data store) but retains many of the same advantages (e.g., only need to update one location, with changes propagating to other stores). Research is ongoing to consider how this might be used for a self-improving architecture based on the GIFT MAA approach.

Rewards/Feedback

The most important but least-understood part of a self-improving learning technology is the reward structure that should be satisfied by that system. For a self-improving system to "improve" implies that some optimization criteria exist and that feedback is available to determine when improvement has occurred. These rewards or feedback might be automated (e.g., based on user performance/improvement) or could be supervised (e.g., based on instructor ratings, experimenter tags, etc.). However, the proper emphasis on optimizing learning is overall a complex issue: should a system optimize for near-transfer (immediate testing on similar tasks), far transfer (delayed testing or less-similar tasks), preparation for future learning (rate of learning new tasks), on-the-job metrics, or some other reward? This is also impacted by the fact that learning systems often lack access to the measures that are most important to learners (e.g., far transfer, on-the-job performance).

This problem can be partially simplified by considering the reward structure on a per-service basis, where each service attempts to optimize a well-specified set of tasks. So then, a dialog classifier might only optimize its ability to select an appropriate response to a single question, rather than to improve overall user satisfaction with the dialog. This approach is more tractable, but then means that each service needs its own reward/feedback mechanism. Overall, self-improvement is likely to be limited by the data that is available to assess its quality. These limitations fall into two categories: relevance (does the service assessment align well to the overall system purpose) and availability (do we have the data to support this feedback signal). Based on these criteria, a few reward/feedback signals for self-improvement have been considered for the GIFT MAA project: benchmarking data, re-test performance, transfer task performance, and preparation for future learning.

Benchmarking Data (Supervised): For some services, particularly classifiers/estimators, it is possible to have a set of supervised tags that provide a reasonable space of quality testing. This might be one or more data sets. In this case, a self-improving service could use real data to improve on a subset of benchmarking data (internal benchmarks) while the performance on out-of-sample data could be determined using a hold-out set (external benchmarks). This approach can then be continued by updating data-driven models until a point is reached where overfitting is observed (i.e., internal benchmarks improve but external ones decrease). This approach can have high availability if data is easy to label, but may have low relevance if new user data does not follow similar patterns as the benchmarking set. This approach is most commonly applied to classical machine learning problems, such as classifiers or estimators.

Selection Data (User Choices): For a different subclass of services, data will readily be available at all times. For example, Google search can be improved by identifying the rate that users click search results and do not need to re-search/continue to further pages. Usage of hints, clicking on recommender outputs, and similar services can fall into this category. However, if done in isolation with many services this could lead to optimizing for user attention for the sake of attention. To ensure that selection-based rewards are valid, it must simultaneously be determined if they are also valuable (i.e., frequency*value). Worse, the changes that would be required to improve such metrics may be user interaction changes that cannot be automated (e.g., re-designing how recommendations are presented). As such, these data sources may ultimately be less valuable for directly optimizing for learning. They are likely more useful for improving engagement (e.g., visiting more parts of a system, reminders about when to return to a system).

Re-Test Performance (Near Transfer): A second fairly general approach is to examine the rate of a user's learning curve when having them perform similar tasks repeatedly. This approach aligns to what is often done with adaptive systems for mathematics, which considers how many practice examples are needed to master a topic (where mastery means reaching a certain reliability in completing certain problem types). While this approach can give high availability of data, it has the pitfall where the system could easily overfit a certain set of task examples (e.g., low relevance to practical use). For example, a system that optimizes rote memorization might be able to satisfy this metric. This approach is most commonly used for generative systems, which can create a large number of examples (e.g., assessed simulations, math problem generators).

Transfer Task Performance (Far Transfer): Transfer tasks fall on a continuum ranging from the most direct re-try performance, where it is not always clear what characteristics make for a reasonable and predictive transfer task. An ideal transfer task would be one that is more-similar to the actual use-cases where the learned skills need to be applied, to indicate that the acquired skills are likely to be useful. However, this would require building a substantial set of such tasks. Moreover, these tasks would typically be withheld until after reaching a sufficient performance on re-test/near transfer tasks. As such, data availability is lower since only a few data points are available for each learner. However, the relevance benefits may be important to prevent optimizing for fragile knowledge. The most common method to gather this kind of feedback is with outside tests or performance tasks.

Preparation for Future Learning: A variant that combines these two metrics (near and far transfer) is to study the learning curve of later skill that requires the earlier skill. So then, if certain changes make learning faster in the future, then those changes are prioritized. However, as with transfer tasks, this requires an increasing amount of data to be effective. Moreover, it assumes that learning will be generally monotonic. However, there are cases where learning progressions result in shifting mental models that improve performance on some skills while decreasing performance on others.

Overall, when considering a generalized tutoring system, it seems unlikely that hand-crafted metrics will consistently be available and that the availability of data may be insufficient to do data-intensive approaches such as preparation for future learning (at least for every domain). As such, it seems that the low-hanging fruit probably include benchmarking data for services with well-defined metrics, leveraging user choice/selection behavior to help optimize certain types of engagement, and a weighted combination of easily-available near transfer outcomes with a limited number of transfer tasks either interleaved or delayed to a post-test.

Recommendations and Future Research

From the current state of generalized tutoring systems research, both in GIFT and in the broader space of learning technology, end-to-end self-improving systems remain a long-term vision. However, the technology exists to make significant improvements to individual component services and agents that should still improve learning outcomes. In particular, benchmarking approaches could be highly effective for improving classifiers and estimators used for certain assessments or dialog-based interactions. User selection of choices generated by services could also be useful, particularly when user adoption or engagement is the metric of interest. Finally, with proper instrumentation and composition of learning resources, it should be possible for services that personalize learning to optimize the efficiency of acquiring new skills. However, as noted in the prior user study, these functionalities are likely too complex for a course author to access.

Instead, authors should likely be selecting between course templates which help optimize their behavior under the hood. So then, a service might exist which attempts to optimize which of two other hint-generation services it should use with a learner (i.e., leveraging proposal patterns to select services). Likewise, a service might utilize failsoft techniques not just for network failures, but for intervention failures: stopping using the results from a service when the learner does not benefit from those interventions. The instructional designer would then simply pick which services *could* be used to personalize the course, which would then self-organize to optimize their overall benefit to the user through confirmed vs. dissolved proposal interactions.

These gains will be pushing against a broader issue for tutoring systems: training groups of learners. At the same time that self-improving services are increasingly realistic in an ITS framework for an individual learner, teams mean that services must be prepared to support multiple learners simultaneously changing a task. This means that individual learning curves and states represent only part of the larger picture (e.g., all learners may be at different levels, may have different roles, etc.). This variability may make it harder for automated systems to optimize the behavior of their components. More generally, team training is a challenge that will impact not just self-improving components of adaptive learning systems but all components. However, it will also open up additional opportunities for feedback and rewards information (e.g., peer assessments, covering behavior where a more-expert user needs to step in and complete tasks).

Acknowledgements

The effort described here is supported by the U.S. Army (ARL W911NF-04-D-0005). Any opinion, content or information presented does not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References

FIPA (2013). *FIPA ACL message structure specification*. www.fipa.org/specs/fipa00061/ SC00061G.html.

Nye, B. D., Auerbach, D., Mehta, T. R., & Hartholt, A. (2017). Building a Backbone for Multi-Agent Tutoring in GIFT (Work in Progress). In *Proceedings of the 5th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym5)* (pp. 23-35).

Nye, B. D., & Morrison, D. M. (2013). Toward a Generalized Framework for Intelligent Teaching and Learning Systems: The Argument for a Lightweight Multiagent Architecture. In *Artificial Intelligence in Education (AIED) 2013 Workshop on the Generalized Intelligent Framework for Tutoring (GIFT)* (pp. 105-115).

Ritter, S. (2015). Authoring for the product lifecycle. *Design Recommendations for Intelligent Tutoring Systems, Volume 3: Authoring Tools and Expert Modeling Techniques*. Sottilare, R., Graesser, A., Hu, X., & Brawner, K. (eds.), (pp., 137-144).

Swartout, W., Nye, B. D., Hartholt, A., Reilly, A., Graesser, A. C., Vanlehn, K., et al. (2016). Designing a Personal Assistant for Life-Long Learning (PAL3). In *Proceedings of the 29th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2016* (pp. 491–496).

SECTION II – MACHINE LEARNING

Dr. Xiangen Hu, Ed.

Chapter 6 – Introduction to the Machine Learning Section

Xiangen Hu
University of Memphis and Central China Normal University

Core ideas

Effective and efficient "instructional systems" are a well-designed ecosystem that includes motivated human learners, appropriate learning resources, proper learning environments, and optimized processes that involve the human learners interacting with the learning resources in the learning environment. The three chapters included in this section have addressed the key issues and challenges of "adaptivity" of "instructional systems." Although these three chapters focus on different aspects of the adaptive nature of "instructional systems", they all use algorithms of **Machine Learning** (ML) to showcase their unique approaches of "self-improving" of adaptive instructional systems.

Individual chapters

The chapter by **Ritter, Baker, Rus, and Biswas** provides insights on how to apply two machine learning algorithms: *Sequence Mining* and *Discriminant Sub-sequence Analysis. Sequence mining* is used to analyze action subsequences and identify differences in learner's approaches in solving problems. A *sequence mining algorithm* can be used to discover important differences between high- and low-performing students in terms of their strategy use. *Discriminant Sub-sequence Analysis* is used to detect good vs. bad tutoring sessions by comparing the sequence of Scaffolding (S) and Fading (F) in tutoring dialogue. Adaptive Instructional Systems (AISs) that are enhanced by these machine learning algorithms can differentiate high-/ low-performing students in problem solving and good/bad tutor-tutee interaction in tutoring dialog. AISs enhanced with such machine learning algorithms are self-improvable AIS.

The chapter by **Leung** and **Williams** suggests implementing the multi-armed bandit (MAB) technique that has been popularized by ML researchers and widely used in other domains. The authors point out that the potential application of the MAB technique to self-improving learning systems in learning could be productive. In their chapter, they have also pointed out potential challenges from ethical, technical, and implementational perspectives. One of the challenges that they have pointed out is the trust in MAB applications in learning systems. They suggested that implementing MABs in a well-studied framework such as GIFT would help to find solutions for the challenges when implementing MAB techniques in self-improving learning systems.

The chapter by **Shen, Shimmei, Chi, and Noboru** presents two case studies demonstrating the utility of Reinforcement Learning (RL) to Self Improving Educational Systems. In their first case study, a RL-induced reward policy was implemented in a test learning system (Deep Thought) to explore the impact of different pedagogical strategies on slow and fast learners. The second case study used RL techniques to content validation for online courseware. Their implementation of RL techniques is in the form of RAFINE (**R**einforcement learning **A**pplication **F**or **IN**cremental courseware **E**ngineering). Applying RAFINE to a section of inefficiently made courseware, they were able to detect ineffective instructional elements in existing online courseware with students' learning activity log data. Both of the case studies offered insight on how ML techniques would help building self-improving learning systems, either at the level of optimizing pedagogical strategies or producing better courseware.

Chapter 7 – Identifying Strategies in Student Problem Solving

Steven Ritter[1], Ryan S. Baker[2], Vasile Rus[3], and Gautam Biswas[4]
[1]Carnegie Learning, [2]University of Pennsylvania, [3]University of Memphis, [4]Vanderbilt University

Introduction

As instructional systems support more open-ended problem-solving, it becomes essential to understand the particular strategies or approaches that students take to solving problems. Increasingly, machine learning techniques to detect student strategies will be used in these instructional systems. One advantage of such self-improving systems is that, as usage broadens, they become more sophisticated in their analysis of student behavior and ability to support a wide variety of learning approaches. In this way, such systems become more educationally effective, particularly for students who employ relatively rare approaches to solving problems.

We consider a strategy to be a sequence of steps or operators taken in a problem space with the goal of accomplishing a given task or solving a problem (Newell & Simon, 1972). In theory, any variation in problem solving may represent a different strategy. In practice, however, we often group insignificant variations in problem solving steps into a single strategy and consider those that represent "significantly" different approaches to represent different strategies. Consider Figure 1. The solutions represented in Strategy A and Strategy B both consist of three similar steps. In the first step, the student subtracts a variable term from both sides of the equation (5x in Strategy A; 3x in Strategy B). In the second step, the student using Strategy A subtracts 4 from both sides, and the student using Strategy B adds 6 to both sides. In the third step, each student divides both sides of the equation by the coefficient. A student employing Strategy C combines the first two steps into a single step, subtracting 3x-6 from both sides of the equation. Clearly Strategy A and Strategy B are similar approaches and might be considered variants of a single strategy. A student who is able to recognize and execute Strategy C is illustrating a more sophisticated approach to problem-solving and might be considered to be using a strategy very different from either A or B.

Strategy A	Strategy B	Strategy C
$3x+4 = 5x-6$	$3x+4 = 5x-6$	$3x+4 = 5x-6$
$-2x+4 = -6$	$4 = 2x-6$	$10 = 2x$
$-2x = -10$	$10 = 2x$	$5 = X$
$x = 5$	$5 = x$	

Figure 1. Three strategies for solving an equation.

The decision about whether to consider a different sequence of steps to be a different strategy depends on the goals of the educational system, so it is important to consider the reasons why we might need to understand and distinguish different strategies. Identification of strategies may allow systems to:

- **Infer the student's level of knowledge**. Sometimes, use of a particular strategy is an indicator of the student's level of understanding. Lemaire and Siegler (1995) found that students' use of more sophisticated strategies and of better sensitivity to appropriateness of strategies changed as students learned single-digit multiplication.

- **Identify misconceptions**. Strategies are approaches to problem solving, but they need not be correct approaches. A student may, for example, incorrectly believe that numbers with more decimal places are larger than numbers with fewer decimal places (Isotani et al., 2010). In the above example, the student may have a misconception about the negation operator, and consider $3x - 5x = 2x$ (always subtract the smaller number from the large). Identification of incorrect problem-solving steps relating to this misconception can greatly aid remediation.

- **Indicate student's sensitivity to problem characteristics**. In some cases, different strategies are more appropriate for different problem types, and part of the target of learning may be that students understand how to develop strategies for different problem types, and then apply appropriate strategies when solving problems. For example, in solving simultaneous equations, substitution and linear combination are both valid strategies that can be applied to a set of equations, but, depending on the equations, one of the strategies may be easier to apply than the other. Proficient students should be flexible enough to choose the appropriate strategy for the problem.

- **Identify use of suboptimal or incorrect strategies**. Like the last bullet, this may occur when students have just learned the individual skills needed to solve problems, but have not had opportunities and practice to combine the use of these skills to solve problems. Students may combine their skills in suboptimal ways when trying to come up with a sequence of steps to solve a problem.

- **Provide opportunities for reflection and generalization**. Exposing students to multiple strategies may encourage greater procedural flexibility (Crowley & Siegler, 1999; Rittle-Johnson & Star, 2007). An educational system may wish to provide students with worked examples and problem-solving opportunities that employ different strategies. This approach would require knowledge of the strategies a particular student employs.

- **Identify metacognitive strategies**. In addition to strategies used to solve individual problems, students employ metacognitive strategies, including self-explanation and use of worked examples. Instructional systems will be more effective to the extent that systems can identify and support productive metacognitive strategies.

- **Identify conceptual understanding.** Work by Rowe and colleagues (2014, 2017) finds that behavioral strategies within learning games can be indicative of differences in conceptual understanding, correlating highly with alternate measures of those same concepts.

- **Indicate students' sensitivity to contextual characteristics.** Different strategies may require different amounts of time to execute or different resources (such as working memory load). Siegler (1988) found that some students employed different strategies under time pressure than they would apply with unlimited time. Such sensitivities may interact with student characteristics. Beilock and DeCaro (2007) found that low working memory capacity students more readily switched their strategies (to simpler heuristics) under pressure than high working memory capacity students.

Methods

In this section, we consider several methods for determining student strategies, given a stream of data describing the student's actions in a problem-solving episode.

Model Tracing

Model-tracing tutors (Corbett, 2001) monitor student activities in problem solving and attempt to map the student's actions to particular problem-solving strategies. Such actions are generally classified as belonging to a correct strategy, an incorrect strategy or an unrecognized strategy. Since strategies (both correct and incorrect) are pre-specified in the model, such systems can recognize strategies that students employ. Steps that are not recognized as belonging to a strategy are considered to be incorrect (but uninterpretable). One downside of this approach is that unique or rare strategies may not be recognized.

The strategies encoded in model-tracing tutors are typically initially discovered through cognitive task analysis (Clark & Estes, 1996; Lovett, 1998), but such systems can be extended through analysis of data collected through such systems, as described here.

In some cases, model-tracing systems can recognize cases where strategy recognition has failed and adapt to incorporation recognition of the new strategy in the future. Ritter (1997) describes one such method by which model-tracing tutors can learn to recognize new strategies. In model-tracing tutors for domains where correct solutions can be evaluated but not modelled, the tutor can recognize and provide feedback for more optimal strategies. For example, an equation solving tutor can learn more efficient strategies that are "demonstrated" by the student.

Detector-based strategy recognition

Another method for identifying student strategies is to develop a machine-learned model, often referred to as a "detector", which recognizes strategies that human beings can identify but cannot easily reduce to a straightforward set of rules, as is necessary for most uses of model tracing. This approach relies upon first obtaining human labels of when the strategy is present or absent, distilling features of the data that are reasonably likely to correspond to that strategy, and then using machine learning to train a model that can replicate the human judgments.

In the first step of this process, software is developed to display a substantial number of examples of student interaction to a set of human coders who can recognize the behavior. These displays can be presented through either a screen replay of learner behavior (e.g. Aleven et al., 2004) or a text replay (e.g. Baker, Corbett, & Wagner, 2006), pretty-printed log files. Then, typically, two coders code the same subset of the total data set and check for acceptable inter-rater reliability. After this, one or both of the coders label the remaining data.

In the second step of the process, data features thought to correspond to the strategy of interest are distilled from the data. This step, often termed feature engineering, can vary in its degree of sophistication, from a single researcher brainstorming a set by herself/himself, to a more structured brainstorming process involving multiple types of expertise, to an in-depth process of interviewing the coders and discussing draft models of their reasoning processes (Paquette et al., 2014).

In the third step of the process, off-the-shelf machine learning algorithms are used to derive a model that replicates the human judgments with reasonable reliability, and the resultant models are tested using cross-validation.

The first strategy this approach was used to study was gaming the system, when a student misuses learning software to proceed without learning. Baker and de Carvalho (2008) took data on student use of a Cognitive Tutor, distilled text replay clips of student behavior, labeled them, and built a decision tree to capture this strategy. Later work extended this approach to SQL-Tutor (Baker, Mitrovic, & Mathews, 2010) and demonstrated that better results could be obtained through a more sophisticated feature engineering process (Paquette et al., 2014). Related work investigated whether a player was seriously attempting to complete quests in an online story-based role playing game (DiCerbo & Kidwai, 2013).

Sao Pedro and his colleagues extended this approach to modeling scientific inquiry strategies, including both whether a student could design a controlled experiment across a set of trials of a simulation (Sao Pedro et al., 2013), and whether the learner planned using a table (Montalvo et al., 2010). A refined version of Sao Pedro's original detector is now used in the commercial Inq-ITS platform (Gobert et al., 2015).

Elizabeth Rowe and her colleagues extended this approach to modeling the strategies students engaged in while playing a conceptual physics game, demonstrating that it was possible to capture gameplay strategies associated with conceptual understanding of Newton's Laws (Rowe et al., 2014) and that these detectors correlated with external measures of Newton's Laws (Rowe et al., 2017).

Together, these examples demonstrate the feasibility of identifying student strategies through an automated detector machine learning approach.

Sequence Mining Methods for Strategy Detection

In the learning sciences and educational psychology research, strategies have been defined as consciously-controllable processes for completing tasks (Pressley, et al., 1989). Within this framework, it is possible to characterize strategies as a sequence of actions that a learner performs to complete a task or subtask in the learning environment. Strategies are further characterized by the context in which they are applied and the specific relations among component activities that make up a strategy. Take, for example, learning environments such as Betty's Brain, where students learn about scientific processes (e.g., climate change) by teaching a virtual agent named Betty (Leelawong & Biswas, 2008: Biswas, et al., 2016). They do this by constructing a visual causal map that represents the relevant scientific process as a set of concepts connected by directed links that represent causal relations. Once taught, Betty can use the map to answer causal questions and explain those answers. The goal for students using Betty's Brain is to teach Betty a correct causal map that matches a hidden, expert model of the domain. The students' learning and teaching tasks are organized around three activities: (1) reading hypertext resources that provide information on the science concepts and causal relations between the concepts, (2) building the causal map using a visual drag and drop interface, and (3) assessing the correctness of the map by getting the agent Betty to take quizzes and evaluating her answers.

Students strategies in this environment revolve around how they combine the three activities to accomplish higher level goals or tasks, such as acquiring information and constructing a part of the causal map (e.g., human activities that cause the greenhouse effect) or correcting errors in a section of their map (e.g., analyze Betty's quiz results, identify causal links that are related to incorrect answers, and correct the erroneous links). However, identifying students' strategies from their activity logs is not an easy task. In open-ended learning environments (OELEs), such as Betty's Brain, the fact that students have choice in the way they go about constructing their models, interpreting their strategies requires an understanding of the task that they are currently working on and an interpretation of their actions in the context of this task.

We have developed sequence mining approaches to derive frequent action sequences when they work in the Betty's Brain environment (Kinnebrew, Loretz, & Biswas, 2013; Kinnebrew, Segedy, & Biswas, 2014). In general, Sequential Pattern Mining (Agrawal & Srikant, 1995) algorithms are designed to find frequent sequential patterns, i.e., series of action that occur in many of the students' activity sequences provided (e.g., the sequential pattern A then B then C occurs in both of the sequences C → A → B → C and A → B → C → A). Researchers have applied sequence mining techniques to a variety of educational data in order to better understand learning behaviors (e.g., Amershi & Conati, 2009; Kinnebrew et al., 2013; Nesbit et al., 2007; Perera et al., 2009; Su et al., 2006; Tang & McCalla, 2002).

To extract the activity sequences of student work in Betty's Brain for sequence mining, log events captured by the learning environment abstracted student activities into a few primary categories with some additional subcategories (Kinnebrew & Biswas, 2012; Kinnebrew, et al., 2013). The primary actions extracted from the logs to generate the action sequences were:

1. *Information acquisition (IA) actions*: (a) *Read*: reading one or more of the science resource pages; and (b) *Note*: entering information into the note-taking tool provided in the system;

2. *Solution construction (SC), i.e., Edit actions*: these include operations on the causal map, with actions further divided by: (i) whether they operate on a causal link or concept and whether the action was an addition (Add), removal (Remove), or modification (Change), e.g., LinkAdd or ConceptRemove;

3. *Solution assessment (SA) actions*: (a) *Query*: students use a template to ask Betty a question, and she answers the question using a causal reasoning algorithm (Leelawong & Biswas, 2008); (b) *Quiz*: students assess how well they have taught Betty by having her take a quiz, which is a set of questions chosen and graded by the Mentor agent; and (c) *Explain*: students probe Betty's reasoning by asking her to explain her answer to a question (either from the quiz or from a query).

Strategies derived from analyzing students' logs across multiple studies with Betty's brain are discussed in a number of our papers (e.g., Kinebrew & Biswas, 2012; Kinnebrew, et al., 2013; Kinnebrew, et al., 2014; Kinnebrew, et al., 2017; Munshi, et al., 2018). We provide some illustrative examples in this chapter. For example, a study conducted in 6th grade science classrooms in 2015, showed frequent use of the *IA → SC* strategy, i.e., they read the resources and then added to or made changes in their causal map. In this case, frequent implied five or more uses of this strategy per student during the course of the intervention. When comparing the use of this strategy by high and low performers, i.e., those who had high versus low scores in their final map scores, we found that high performers used this strategy to make correct changes to their map (i.e., added a correct link or deleted an incorrect one) 62% (*SD* = 9%) percent of the time, whereas low performers made correct changes to their map only 53% (*SD* = 16%) of the time.

To better understand how students employed this general strategy, we considered two specific variants of this strategy in the Betty's Brain environment: *IA → Add a causal link* to the map and *IA → Correct a causal link* in the map by changing or removing an incorrect causal link in the map. Results indicate that *IA → Add a causal link* was used by high performing students on average 23.9 (*SD* = 16) times, and they performed this task correctly 59.3% (*SD* = 13.1%) of the time, whereas low performers performed this strategy on average only 8.9 (*SD* = 7.5) with a 48.4% (*SD* = 22.2%) correct use. Similarly, for the *IA → Correct a causal link* strategy, the numbers were 5.3 (*SD* = 4) with 75.1% (*SD* = 17.2%) correct use for the high performers, whereas the numbers were 2.9 (*SD* = 3.6) with 81.1% (*SD* = 23.5%) correct use by low performers. Though the accuracy numbers are not much different, the high performers use the strategy many more times than the low performers, thus generating better maps and better learning gains overall than the low performers. Linked to this strategy, we also found that the high performers had significantly longer *read* time per pattern than the low performers.

Another frequent pattern used by students corresponds to a debugging strategy, i.e., *SA → LinkAdd → SA* with two specific variants: (1) *Quiz Explanation → Link Add → Quiz* and *Quiz → LinkAdd → Quiz*.

These patterns suggest an informed *guess-and-check* strategy in which the quiz results (either the overall results or the information gleaned from a *Quiz Explanation* for a specific quiz question) are used to suggest a potentially-missing link, which is then added to the map. This is followed by checking the correctness of the "*guess*" by taking another quiz. As expected, for a guess-and-check strategy, the link added was usually incorrect (average percentage of correct additions per student = 19%, *SD* = 14%).

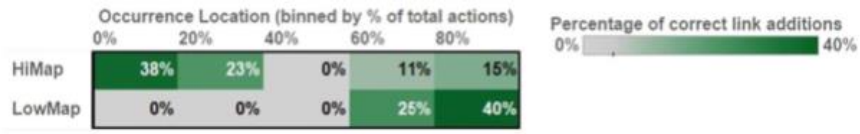


Figure 2. Heatmap of percentage correct links added in $SA \rightarrow LinkAdd \rightarrow SA$ over time.

Given the more detailed information available from the quiz question explanation, we initially expected better performance in adding a correct link for that variant of the strategy. However, this variant actually had a marginally lower percentage of correct additions compared to the other variant. Further, analysis of effectiveness over the course of students' work in the environment illustrated that performance (correctness percentage) with $SA \rightarrow LinkAdd \rightarrow SA$ was better early and late while being especially poor in the middle. The performance heat map shown in Figure 2, indicates that while the high performing (HiMap) students performed best with this strategy early and (to a lesser extent) late in the intervention, the low performing (LowMap) students did not make correct link additions with this strategy until relatively late (after at least 60 percent of their total actions on the system). This may imply that it took the low performing students until late in the intervention to understand how to interpret and use the quiz results. On the other hand, the high performers used this strategy with more success in the early phases of map building. The HiMap students' effectiveness with this strategy may have dropped off once they started dealing with the more difficult material (for which they had little prior knowledge) toward the middle of their activities, finally rebounding some as they gained proficiency. In addition to illustrating the importance of incorporating the overall informed guess-and-check strategy in the strategy model, analysis of this high lift pattern suggests that there may be additional interactions with prior knowledge and skills worth investigating through further experiments.

The results of combining sequence mining algorithms with additional analysis of the relations between actions, showed potentially important differences between high- and low-performing students in terms of their strategy use. Overall, an effective analysis framework applied to the rich behavioral data produced by OELEs has the potential to enable deeper analyses of students' cognitive and metacognitive behavior in complex learning tasks. Ultimately, we believe that this analysis framework can form the basis for designing richer learner modeling schemes that characterize students' activities by analyzing their learning behaviors and performance with respect to their cognitive and metacognitive processes.

Discriminant Sub-sequence Analysis from Tutorial Dialogues

A key research question in intelligent tutoring systems and in the broader instructional research community is understanding what expert tutors do (Rus, D'Mello, Hu, & Graesser, 2013). This goal is motivated by research showing that expert tutors are very effective (Bloom, 1984).

Indeed, understanding what expert tutors do has been a research goal undertaken by theoreticians and empiricist alike. A typical operationalization of this goal of understanding of what good tutors do is to define the behavior of tutors based on their actions. To this end, the learner-tutor interactions are broken down into primitive actions and then significant differences between expert tutors and less accomplished tutors are reported. For instance, Boyer and colleagues (2011) modelled the learner-tutor interaction as sequences of task actions (e.g., opening a file) and dialogue acts, i.e. actions behind utterances, while Cade and colleagues (2008) used just dialogue acts to model the learner-tutor interaction.

In a discriminant sub-sequence analysis approach (Rus, et al., 2017; Maharjan, Gautam, & Rus, 2018), tutorial dialogues are modeled as dialogue-act sequences because there are no other types of actions, e.g. task actions as in Boyer and colleagues (2011), considered in the analysis (Rus, et al., 2017; Maharjan, Gautam, & Rus, 2018). This view of a tutorial dialogue as a sequence of actions is based on the language-as-action theory (Austin, 1962; Searle, 1969). According to the language-as-action theory, when we say something we do something. Therefore, all utterances in a tutorial dialogue are mapped into corresponding dialogue acts using, in our case, a predefined dialogue or speech act taxonomy. The taxonomy was defined by educational experts and resulted in a two-level hierarchy of 17 top-level dialogue acts and a number of dialogue subacts. The exact number of subacts differs from dialogue act to dialogue act. The taxonomy identifies 129 distinct dialogue act and sub-act combinations. Further, we have a set of 17 different dialogue modes defined by experts as in the following: Assessment, Closing, Fading, ITSupport, Metacognition, MethodID, Modeling, OffTopic, Opening, ProblemID, ProcessNegotiation, RapportBuilding, RoadMap, SenseMaking, Scaffolding, SessionSummary and Telling. A detailed description of the dialogue modes is available (Morrison et al., 2015). It should be noted that automatically discovered dialogue act taxonomies are currently being built (e.g. Rus, Graesser, Moldovan, & Niraula, 2012) but it is beyond the scope of this chapter to automatically discover the dialogue acts in our tutoring sessions.

A large corpus of about 19K tutorial sessions between professional human tutors and actual college-level, adult students was collected via an online human tutoring service. Students taking two college-level developmental mathematics courses (pre-Algebra and Algebra) were offered these online human tutoring services at no cost. The same students had access to computer-based tutoring sessions through Adaptive Math Practice, a variant of Carnegie Learning' Cognitive Tutor (Ritter et al., 2007). A subset of 500 tutorial sessions containing 31,299 utterances was randomly selected from this large corpus for annotation with the requirement that a quarter of these 500 sessions would be from students who enrolled in one of the Algebra courses (Math 208), another quarter from the other course (Math 209), and half of the sessions would involve students who attended both courses.

This research investigated which distinctive subsequences of dialogues, dialogue acts and modes comprise effective and less-effective sessions. To this end, each tutorial session was rated by Subject Matter Experts (SMEs) using a 1-5 scale (5 being best score) along two dimensions: evidence of learning (EL) and evidence of soundness (ES). The ES score reflects how well tutors applied pedagogically sound tactics in tutorial sessions. On the other hand, the EL score reflects how well students learned from tutorial sessions. The EL and ES scores were found to be highly correlated (Pearson coefficient of 0.7). The research categorized all human annotated sessions having ES and EL scores less than 2 as ineffective, and all sessions rated with ES = 5 and EL = 4 as good or effective sessions.

Then sequence pattern mining was conducted using the Traminer package in R. The Traminer algorithm first finds the most frequent subsequences by counting their distinct occurrences and then applies a Chi-squared test (Bonferroni-adjusted) to identify sub-sequences that are statistically more (or less) frequent in each group. A *p*-value < 0.4 threshold was used to select likely distinctive sub-sequences, with dialogue acts, actsubacts and mode-switches used as observations. The observations were granularized further by adding speaker information.

It should be noted that a subsequence is not necessarily a contiguous sequence of observations, but the order of the observations is preserved. For example, (Assertion)-(Expressive) is a valid sub-sequence of dialogue acts formed from the (Assertion)-(Request)-(Expressive) contiguous sequence fragment. Sub-sequences were generated up to length 7 from all the annotated tutorial sessions.

The discriminant sub-sequences mined indicated that good tutors use more Expressives and prompt students more in sessions of high learning gains. It was observed that all discriminant sub-sequences of acts contain Expressive acts initiated by tutors or students. The good tutors often prompt students to confirm the students are following their tutoring or, to elicit further answers or reasoning from the students. Furthermore, the tutors' expressions of praise (T-Expressive-Positive) and farewell (T-Expressive-Farewell) and, the students expressing thanks (S-Expressive-Thanks) are highly predictive of effective sessions. The tutors often praise students to keep them engaged to the task or, when they answer correctly. The tutee expressing thanks (S-Expressing-Thanks) might suggest the tutee is satisfied with the tutoring. Moreover, the tutor expressing farewell indicates the tutoring is coming towards the end. The sessions having proper closing also might suggest that both student and tutor are satisfied with the tutorial Session.

The discriminant subsequent analysis for modes revealed interesting patterns as well when analyzed through a pedagogical lens. Good tutorial sessions have Scaffolding (S) and Fading (F) as the dominant strategies i.e. the good tutors do more Scaffolding and Fading to get the problem solved by the students themselves. The sub-sequences S-S, F, S-F, F-F are very strong indicators of good sessions (p-value<0.05) while F-S, F-F-S, S-F-S also fairly indicate the sessions of top quality. Another interesting observation is that the Closing mode (p-value=0.0475) is also a very strong indicator of top sessions. Moreover, a Fading-Closing (p-value=0.002) sub-sequence is even more predictive than Closing alone. We also observed that switching to Scaffolding or Fading modes after ProblemIdentification is more effective as evidenced by sub-sequences O-P-F (p-value=0.1764), P-F (p-value=0.0198) and P-S-S (p-value=0.0362).

Discussion

The methods described here allow us to identify differences in students' approaches to solving problems. It is important to remember that not all differences may be instructionally relevant. Within each method (or combination of methods), instructional system designers will need to make decisions about which approaches represent strategies that are indicative of different instructional needs or differing assessments of students' capabilities and knowledge.

In many cases, it is inappropriate to talk about a student's strategy as being a property of that student. The particular strategy employed may differ due to problem characteristics and also due to that student's state of knowledge. A particularly interesting case is where changes in a student's strategy over time can be considered a measure of the student's learning of the target knowledge. Corbett et al. (2000) describe one such case. Students were asked to complete a table representing the mathematical encoding of a word problem. Novice students typically started to complete the table in an order that did not take the problem structure into account: top-to-bottom and left-to right. As students came to understand the hierarchical structure of the mathematical terms underlying the problem, the order in which they filled in the cells in the table came to match the underlying structure. In cases like this, the students' strategy may be an indicator of the student's understanding of the mathematics, somewhat independent of their success in the task itself.

Recommendations and Future Research

Given the importance of understanding the particular strategies that students employ when solving problems, it is essential that such systems be able to detect and respond to different student strategies. In more open educational environments, where different solution strategies are encouraged, the ability to understand varied student strategies is even more important. In fact, widely used educational systems that support a wide variety of strategies may be the best source for data on the strategies that students use. Using the techniques outlined here, such systems may greatly benefit from becoming self-improving systems that are able to detect and react to novel student problem-solving approaches.

Within the Generalized Intelligent Framework for Tutoring (GIFT) architecture, instructional systems would benefit from recording and sharing student strategies employed while solving particular problems, in addition to the activity and evaluative information already stored. Domain models might benefit from knowledge of likely strategies, which could be used for activity selection, especially in cases where appropriate strategy choice and problem characteristics are strongly linked. Task designers should take strategy use into account. In some cases, tasks might be designed to be maximally flexible, allowing students to employ strategies that they pick. In other cases, an educational goal might be to ensure that students master more than one strategy. For this kind of goal, task designers might design multiple variants of the user interface, each of which constrains the student to employ a particular strategy.

References

Agrawal, R., & Srikant, R. (1995, March). Mining sequential patterns. In *ICDE* (p. 3). IEEE.

Aleven, V., McLaren, B.M., Roll, I., Koedinger, K.R. (2004). Toward tutoring help seeking: Applying cognitive modeling to meta-cognitive skills. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems (ITS 2004)*, 227-239.

Amershi, S., & Conati, C. (2009). Combining unsupervised and supervised classification to build user models for exploratory. JEDM| *Journal of Educational Data Mining*, *1*(1), 18-71.

Austin, J.L. (1962). *How to do Things with Words*. Oxford University Press, 1962.

Baker, R.S.J.d., Corbett, A.T., Wagner, A.Z. (2006). Human Classification of Low-Fidelity Replays of Student Actions. *Proceedings of the Educational Data Mining Workshop at the 8th International Conference on Intelligent Tutoring Systems*, 29-36.

Baker, R.S.J.d., de Carvalho, A. M. J. A. (2008). Labeling Student Behavior Faster and More Precisely with Text Replays. *Proceedings of the 1st International Conference on Educational Data Mining*, 38-47.

Baker, R.S.J.d., Mitrovic, A., Mathews, M. (2010). Detecting Gaming the System in Constraint-Based Tutors. *Proceedings of the 18th Annual Conference on User Modeling, Adaptation, and Personalization*, 267-278.

Beilock, S. & DeCaro, M. (2007). From poor performance to success under stress: Working memory, strategy selection, and mathematical problem solving under pressure. *Journal of Experimental Psychology. Learning, Memory, and Cognition*, *33*, 983-98. 10.1037/0278-7393.33.6.983.

Biswas, G., Segedy, J.R., & Bunchongchit, K. (2016). From design to implementation to practice – A learning by teaching system: Betty's brain. *International Journal of Artificial Intelligence in Education*, *26*(1), 350-364.

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, *13*(6), 4-16.

Boyer, K. E.; Phillips, R.; Ingram, A.; Ha, E. Y.; Wallis, M.; Vouk, M.; and Lester, J. 2011. Investigating the relationship between dialogue structure and tutoring effectiveness: A hidden markov modeling approach. *International Journal of Artificial Intelligence in Education*, *21*(1-2), 65–81.

Cade,W., Copeland, J., Person, N., & DMello, S. (2008). Dialogue modes in expert tutoring. In *Intelligent tutoring systems*, 470–479. Springer.

Clark, R. E., & Estes, F. (1996). Cognitive task analysis. *International Journal of Educational Research*, *25*(5), 403–417.

Corbett, A. T., McLaughlin, M., Scarpinatto, C. & Hadley, W. S. (2000). Analyzing and Generating Mathematical Models: An Algebra II Cognitive Tutor Design Study. In G. Gauthier, C. Frasson and K. VanLehn (Eds.) *Intelligent Tutoring Systems: 5th International Conference, ITS 2000*. New York: Springer, pp. 314-323.

Corbett, A.T. (2001). Cognitive computer tutors: Solving the two-sigma problem. *User Modeling: Proceedings of the Eighth International Conference, UM 2001*, 137-147.

Crowley, K., & Siegler, R. S. (1999). Explanation and generalization in young children's strategy learning. *Child Development*, *70*, 304-316.

DiCerbo, K., Kidwai, K. (2013) Detecting Player Goals from Log Files. *Proceedings of the 6th International Conference on Educational Data Mining*.

Gobert, J. D., Kim, Y. J., Sao Pedro, M. A., Kennedy, M., & Betts, C. G. (2015). Using educational data mining to assess students' skills at designing and conducting experiments within a complex systems microworld. *Thinking Skills and Creativity*, *18*, 81-90.

Isotani, S., McLaren, B.M.,& Altman, M. (2010). Towards intelligent tutoring with erroneous examples: A taxonomy of decimal misconceptions. In V. Aleven, J. Kay, J. Mostow (Eds.). *Proceedings of the 10th International Conference on Intelligent Tutoring Systems (ITS-10),* Lecture Notes in Computer Science, 6094 (pp. 346-348). Berlin: Springer.

Kinnebrew, J., Segedy, J.R. & Biswas, G. (2017). Integrating Model-Driven and Data-Driven Techniques for Analyzing Learning Behaviors in Open-Ended Learning Environments. *IEEE Transactions on Learning Technologies*, *10*(2), 140-153.

Kinnebrew, J.S., Segedy, J.R., & Biswas, G. (2014). Analyzing the Temporal Evolution of Students' Behaviors in Open-Ended Learning Environments. *Metacognition and Learning*, *9*(2), 187-215.

Kinnebrew, J.S., Loretz, K.M., & Biswas, G. (2013). A Contextualized, Differential Sequence Mining Method to Derive Students' Learning Behavior Patterns. *Journal of Educational Data Mining*, *5*(1), 190-219.

Kinnebrew, J.S., & Biswas, G. (2012). Identifying Learning Behaviors by Contextualizing Differential Sequence Mining with Action Features and Performance Evolution. In *Proceedings of the 5th International Conference on Educational Data Mining*. Chania, Greece.

Leelawong, K., & Biswas, G. (2008). Designing learning by teaching agents: The Betty's Brain system. *International Journal of Artificial Intelligence in Education*, *18*(3), 181-208.

Lemaire, P., & Siegler, R. S. (1995). Four aspects of strategic change: Contributions to children's learning of multiplication. *Journal of Experimental Psychology: General*, *124*(1), 83-97.

Lovett, M. C. (1998). Cognitive task analysis in service of intelligent tutoring system design: a case study in statistics. In *Intelligent Tutoring Systems, Proceedings of the Fourth International Conference*. (pp. 234-243).

Maharjan, N., Gautam, D., & Rus, V. (2018). Discovering Effective Tutorial Strategies in Human Tutorial Sessions, *Proceedings of the 31st International Florida Artificial Intelligence Research Society Conference (FLAIRS 2018)*, Melbourne Beach, FL, USA May 21-23, 2018.

Montalvo, O., Baker, R.S.J.d., Sao Pedro, M.A., Nakama, A., Gobert, J.D. (2010) Identifying Student' Inquiry Planning Using Machine Learning. *Proceedings of the 3rd International Conference on Educational Data Mining*, 141-150.

Morrison, D. M., Nye, B., Rus, V., Snyder, S., Boller, J., & Miller, K. (2015, June). Tutorial dialogue modes in a large corpus of online tutoring transcripts. In International Conference on Artificial Intelligence in Education (pp. 722-725). Springer, Cham.

Munshi, A., Rajendran, R., Ocumpaugh, J., Biswas, G., Baker, R. S., & Paquette, L. (2018). Modeling Learners Cognitive and Affective States to Scaffold SRL in Open Ended Learning Environments. In *Proceedings of the International Conference on User Modelling, Adaptation and Personalization* (UMAP), Singapore, pp. 131-138.

Nesbit, J. C., Zhou, M., Xu, Y., & Winne, P. (2007). Advancing log analysis of student interactions with cognitive tools. In *12th biennial conference of the European association for research on learning and instruction* (EARLI) (pp. 2-20).

Newell, A. and Simon, H. A. (1972). *Human Problem Solving*. Prentice-Hall.

Paquette, L., de Carvalho, A.M.J.A., Baker, R.S., Ocumpaugh, J. (2014) Reengineering the Feature Distillation Process: A Case Study in the Detection of Gaming the System. *Proceedings of the 7th International Conference on Educational Data Mining*, 284-287.

Perera, D., Kay, J., Koprinska, I., Yacef, K., & Zaïane, O. R. (2009). Clustering and sequential pattern mining of online collaborative learning data. *IEEE Transactions on Knowledge and Data Engineering*, 21(6), 759-772.

Pressley, M., Goodchild, F., Fleet, J., Zajchowski, R., & Evans, E. D. (1989). The challenges of classroom strategy instruction. *The Elementary School Journal*, *89*(3), 301-342.

Ritter, S. (1997). Communication, cooperation and competition among multiple tutor agents. In du Boulay, B. and Mizoguchi, R. (Eds). *Artificial Intelligence in Education: Knowledge and media in learning systems* (pp 31-38). Amsterdam: IOS Press.

Ritter, S., Anderson, J.R., Koedinger, K.R., & Corbett, A. (2007). The cognitive tutor: Applied research in mathematics education. *Psychonomics Bulletin & Review*, *14*(2), pp. 249-255.

Rittle-Johnson, B. and Star, J. R. (2007). Does comparing solution methods facilitate conceptual and procedural knowledge? An experimental study on learning to solve equations. *Journal of Educational Psychology*, *99*(3), 561–574.

Rowe, E., Baker, R.S., Asbell-Clarke, J., Kasman, E., Hawkins, W.J. (2014) Building Automated Detectors of Gameplay Strategies to Measure Implicit Science Learning. *Proceedings of the 7th International Conference on Educational Data Mining*, 337-338.

Rowe, E., Asbell-Clarke, J., Baker, R.S., Eagle, M., Hicks, A.G., Barnes, T.M., Brown, R.A., Edwards, T. (2017) Assessing implicit science learning in digital games. *Computers in Human Behavior*, *76C*, 617-630.

Rus, V., Maharjan, N., Tamang, L.J., Yudelson, M., Berman, S., Fancsali, S.E., Ritter, S. (2017). An Analysis of Human Tutors' Actions In Tutorial Dialogues, *Proceedings of the 30th International Florida Artificial Intelligence Research Society Conference (FLAIRS 2017)*, San Marco Island, Florida, USA May 16-18, 2017.

Rus, V., Moldovan, C., Niraula, N., & Graesser, A. C. (2012). Automated Discovery of Speech Act Categories in Educational Games. International Educational Data Mining Society.

Sao Pedro, M.A., Baker, R.S.J.d., Gobert, J., Montalvo, O. Nakama, A. (2013) Leveraging machine-learned detectors of systematic inquiry behavior to estimate and predict transfer of inquiry skill. *User Modeling and User-Adapted Interaction*, *23* (1), 1-39.

Searle, J. 1969. *Speech acts*. Cambridge University Press.

Siegler, R. S. (1988). Individual differences in strategy choices: Good students, not-so-good students, and perfectionists. *Child Development*, *59*, 833-851.

Su, J. M., Tseng, S. S., Wang, W., Weng, J. F., Yang, J. T. D., & Tsai, W. N. (2006). Learning portfolio analysis and mining for SCORM compliant environment. *Educational Technology & Society*, *9*(1), 262-275.

Tang, T. Y., & McCalla, G. (2002, July). Student modeling for a web-based learning environment: A data mining approach. In *AAAI/IAAI* (pp. 967-968).

Chapter 8 – Multi-Armed Bandits in Education

Weiwen Leung and Joseph Jay Williams
University of Toronto

Introduction

Any digital educational resource has the potential to become a self-improving system, if its components can be enhanced by creating multiple versions and testing out which are effective (for different people), and using this data to provide better versions for future learners. Machine learning algorithms can be applied to dynamically analyze data from these experiments and present the most effective versions of a resource to future students, leading to continual improvement.

Intelligent self-improving systems are becoming increasingly popular. Such approaches have been applied to create intelligent self-improving systems in the display of advertisements or creation of a user in large technology companies such as Facebook and Google, where the algorithms gradually learn consumer preferences, such as presenting interface features that users engage with, or displaying advertisements that certain subgroups of consumers are likely to click on. There are two key advantages to such systems. First, such algorithms can automatically converge towards a good outcome with little or no human effort. Second, such algorithms can provide personalized experiences; in the context of online advertising, this means that individual users are likely to see advertisements about products that interest them.

However, such systems are still not common in educational settings. For example, while websites such as Coursera, Khan Academy, or edX provide a plethora of educational material to learners, the same material is often presented to learners, regardless of ability. Such material is often static, as instructors may not have time to edit the content after it is released. It is easy to see that different users can benefit from different kinds of content; for example, high ability students may benefit from challenging questions while lower ability students may benefit from easier questions.

The contextual multi-armed bandit is an easy way to introduce personalization and self-improvement in education. In its simplest form (this special case is called a "multi-armed bandit" or MAB in short), there is an agent who can choose from multiple actions. Each action gives a reward that is in part deterministic and in part stochastic. The agent generally knows nothing about the reward distribution of each action, but learns them over time. To give a concrete example in an educational setting, an agent could be an instructor, while actions could include different types of explanations to each problem. Rewards could be the actual learning of a student from an explanation (perhaps proxied by their test scores on future related questions). A typical MAB chooses between explanations (i.e. actions) randomly initially, but as students complete questions, explanations that are viewed as more effective are shown more frequently. Contextual multi-armed bandits (sometimes called contextual bandits) are generalizations of MABs in that the optimal explanation can vary by learner subgroup.

Several studies have shown the potential widespread applicability of MABs in education. To give a few examples, MABs have shown potential to recommend the best personalized actions to learners (Clement, Roy, Oudeyer, & Lopes, 2015; Lan & Baraniuk, 2016), what teaching materials to display (Liu, Mandell, Brunskill, & Popovic, 2014), and what explanations to show (Williams, et al., 2016). However, there are many barriers to adopting MABs in educational settings. As such, this chapter continues by discussing possible challenges in MAB adoption through an economic framework. Next, it goes on to list possible strategies to increase adoption and trust in MABs. It then goes on to suggest possible fertile areas of research on MAB adoption in education. Fourth, the chapter goes on to describe implications for GIFT, and then concludes.

Challenges in Multi-Armed Bandit Adoption

We discuss possible challenges regarding the adoption of MABs through the lens of an economic framework: specifically, we analyze possible challenges to demand (from instructors) and supply (from suppliers such as developers).

Demand Factor #1: Ethical considerations

Some ethical concerns revolve around fairness: one natural question to ask when introducing MABs is whether two students in the same class should receive different kinds of instruction, e.g. two different explanations to the same question, or even work on two different quiz problems. Other issues include informed consent: not having informed consent can be viewed as ethically questionable, but having informed consent could change the participants' behavior as participants know they are in an experiment, and selection biases if some students opt-out.

A natural counterpoint to the fairness argument is that it is too narrowly construed (List, 2011): not experimenting would not be fair to future students, as they do not benefit from improved instruction. Even today we (and our children) are benefitting from experiments done previously. Yet another counterargument is that giving all students the same instructional content and support can also be viewed as doing an experiment where everyone receives the same treatment.

Certain issues such as informed consent are more delicate. Here, it is better for researchers to carefully weigh the possible harm inflicted against the scientific benefits of conducting the study without informed consent when making applications to their Institutional Review Board (IRB), and work with their IRBs to reach a defensible solution. In addition, alternatives to standard informed consent such as superset consent may be feasible (Desposato, 2014).

Demand Factor #2: Reluctance to experiment

Another factor is that when using multi-armed bandits, one is conducting an adaptive experiment, and people are generally reluctant to experiment. On a personal level, Mullainathan (2017) documents his personal reluctance to even experiment trying something as simple as generic soda, while Larcom, Rauch and Willems (2017) show that forced experimentation with new transportation routes due to strikes led to many London commuters re-optimizing their travel plans, suggesting that London commuters under-experimented with transportation routes. Levitt (2016) also shows that people are likely excessively cautious in keeping the status quo.

We are unaware of any literature that directly sheds light on overcoming the human reluctance to experiment, though the Porter hypothesis suggests that exogenously imposed constraints (e.g. government regulations) may trigger innovation through experimentation (Porter, 1991). However, in the next section, we explain what insights from the literature on algorithmic trust may be applicable in increasing the trust that people have in MABs.

Supply Factor #1: Cost of programming software with MABs

We now consider supply side considerations. Educational software that uses MABs can be expensive to build. While simple MABs are not difficult to program, more complex contextual MABs can be more difficult to program. Perhaps the most costly factor in using such systems is the need to design multiple versions of a piece of content (e.g. multiple explanations for a given quiz question), in order to test out which works best.

However, it is possible to greatly reduce the time required to produce the required content through techniques such as learnersourcing (or more generally, crowdsourcing). For example, learners can be asked to write their explanations as they solve questions (Williams, et al., 2016), which benefits both future students and learners themselves as self-explanations can help to reinforce one's understanding of a topic (Chi, De Leeuw, Chiu, & Lavancher, 1994). In addition, some of the software created by academics has been released open-source (Williams, et al., 2018), which other developers (commercial and non-commercial) can potentially build on or incorporate in their platforms.

Supply Factor #2: Difficult to interface bandits with offline teaching materials

Another obstacle is that much of teaching material is "offline" in the sense that they are completed with pen and paper, rather than using information technology. This limits the applicability of educational software that uses MABs and hence its supply. The supply of software that uses MABs could be higher if MABs could be configured to, for example, provide personalized recommendations for problems to solve in physical textbooks and analyze the answers that students write on paper.

There is no easy solution to overcome this problem. However, the increasing popularity of online learning will increase the applicability of MABs. Moreover, one can imagine some partial solutions even at this point. For example, software developers could work with textbook authors to provide problem recommendations for students.

Increasing Trust in Multi-Armed Bandits

Having considered possible methods to increase MAB adoption, we now turn to the literature on algorithmic trust to understand possible factors that may lead to increased trust and hence greater continued use (i.e. less attrition) of MABs among those who start using MABs. (That said, these factors may also increase MAB adoption).

A first plausible factor is reminding people that they should compare MAB performance against a human's, rather than a perfect algorithm. In a study where participants were asked whether they want to rely on algorithmic forecasts of MBA student's future success when deciding which MBA students to admit, participants quickly lost confidence in the algorithm once they saw it make mistakes. In contrast, they were more forgiving of human mistakes, whether their own or other people's (Dietvorst, Simmons, & Massey, 2014). Another study found evidence that suggests that this could be due to people having higher expectations for algorithms compared to humans (Dietvorst, 2016). Hence, it may help to raise awareness that people should compare MAB and human performance against the same yardstick.

A second plausible factor is explainability: A study found that people are averse to recommender systems, despite such systems producing higher quality recommendations than humans, be they strangers, friends, or family (Yeoman, Shah, Mullainathan, & Kleinberg, 2018). Part of this aversion is due to the perception that human recommendation processes are easier to understand. Hence, producers of educational software that use MABs should provide careful explanations regarding how their algorithms work.

A third plausible factor is modifiability: when given the power to modify an algorithm's forecasts, people are considerably more likely to rely on the algorithm rather than human judgement (Dietvorst, Simmons, & Massey, Overcoming Algorithm Aversion: People Will Use Imperfect Algorithms If They Can (Even

Slightly) Modify Them, 2016). This result held even when participants were severely restricted in how they could modify the algorithm, suggested that people had preferences from even a little control over the algorithm. Hence, designers should consider letting end-users modify experimental parameters such as those which affect the exploration-exploitation tradeoff.

A fourth possible factor is getting end-users themselves to make forecasts about which experimental treatments will be most effective. In a proof-of-concept study of educational software that used MABs, instructors appeared to be humble after their initial predictions about which experimental treatments would be most effective were proven wrong (Williams, et al., 2018). Hence, designers can consider adding features that prompt people for their *a priori* predictions before they launch an experiment. Showing that their long-held beliefs are not necessarily correct could make people want to continue use of software that uses MABs. Indeed, political beliefs are perhaps some of the most strongly held beliefs people have, but even political beliefs can change after people make forecasts that subsequently turn out to be wrong (Mellers, Tetlock, & Arkes, 2018).

Finally, emphasizing objective aspects of certain decisions may help to increase continued use of MABs. In contrast to Dietvorst, Simmons, and Massey (2014), Logg (2017) found that people have preference for algorithms (rather than algorithmic aversion). However, this “algorithmic appreciation” only held for decisions that were perceived to be objective (e.g. financial decision making), rather than subjective decisions such as dating. Hence, it may help to emphasize the objective aspects of certain areas (or to promote MABs in areas where decision making is perceived to be an objective process).

Open Questions on Multi-Armed Bandit Adoption

There are many fertile areas of research regarding MAB adoption. How to overcome challenges in MAB adoption is an example of an open question. While we went through several challenges and gave some plausible solutions, these ought to be empirically tested. In addition, the ideas we gave for increasing trust in MABs were based on the algorithmic trust literature. While some of those papers used algorithms which had processes similar to MABs (e.g. prediction), and others used systems which could incorporate MABs (recommender systems), the findings of those papers may not necessarily generalize to MABs.

Recommendations and Future Research

In this section, we provide design recommendations for the Generalized Intelligent Framework for Tutoring (GIFT) and future Intelligent Tutoring Systems, and discuss future research needs where gaps are anticipated to persist.

GIFT can incorporate functionality for randomized experiments to be conducted on components of educational resources, such as hints, explanations, motivational messages, and problems. Contextual bandit algorithms can then be used to automatically discover which alternative versions work for different learners, leading to continual self-improvement of the educational system. Currently GIFT supports extensive capacities for authoring intelligent personalized systems, but these require careful engineering and development of the rules or other processes for personalization. By incorporating the capacity for randomized experimentation that machine learning algorithms can improve, GIFT can enable continual improvement anywhere an experiment can be conducted.

This approach can benefit from taking into account the findings of the algorithmic trust literature mentioned in the “Increasing Trust” section. Experimental parameters in GIFT software can be made more modifiable. For example, GIFT users can more easily change the degree to which the system tries less tested options (as compared to choosing options with the highest expected payoff). GIFT developers can also add insightful explanations about how GIFT works to the software. Teachers may be more likely to adopt GIFT if they are clear on how it works. Also, GIFT users can also be nudged to make predictions about the most effective experimental treatments, and then shown how their predictions compare with the actual results; realizing that one's predictions are not always correct will highlight the usefulness of GIFT. A second recommendation is that when building and disseminating GIFT, it would be good to pay more attention to overcoming the human reluctance to experiment.

There are many future research needs with respect to improving current systems. First, existing systems can be generalized to handle the lack of incentives to be the first to contribute content (or the "incentive problem"). Imagine an MAB system that assigns explanations to students. The system will likely give better explanations to students who complete questions later on. As such, some students may complete quizzes and assignments later than they otherwise would. A second problem is non-stationarity: even without the incentive problem, the reward function can change as students who complete an assignment close to the deadline may be different from those who complete the assignment well before the deadline.

Conclusion

This chapter started off by considering the potential widespread applicability of software that uses MABs in educational settings, and then went on to consider possible challenges to the adoption of such software. We then considered potential hindrances to the use of MABs, as well as ways these could be surmounted by using insights from the algorithmic trust literature. We also gave implications for GIFT, as well as areas for future research. We hope that future developers continue to bring in insights to the field by adapting innovations in other areas.

Acknowledgements

This research was supported by funding from the Office of Naval Research (Grant N00014-18-1-2755).

References

Chi, M., De Leeuw, N., Chiu, M.-H., & Lavancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, 439-477.

Clement, B., Roy, D., Oudeyer, P.-Y., & Lopes, M. (2015). Multi-armed bandits for intelligent tutoring systems. *Journal of Educational Data Mining*, 20-48.

Desposato, S. (2014). Ethical Challenges and Some Solutions for Field Experiments. *Working Paper*, 1-11.

Dietvorst, B. (2016). People reject (superior) algorithms because they compare them to counter-normative reference points. *Working Paper*.

Dietvorst, B., Simmons, J., & Massey, C. (2014). Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General*, 1-13.

Dietvorst, B., Simmons, J., & Massey, C. (2016). Overcoming algorithm aversion: People will use imperfect algorithms if they can (even slightly) modify them. *Management Science*, 1155-1170.

Lan, A., & Baraniuk, R. (2016). A contextual bandits framework for personalized learning action selection. *Proceedings of the 9th International Conference on Educational Data Mining*, (pp. 424-429).

Larcom, S., Rauch, F., & Willems, T. (2017). The benefits of forced experimentation: Striking evidence from the london underground network. *The Quarterly Journal of Economics*, 2019-2055.

Levitt, S. (2016). Heads or tails: The impact of a coin toss on major life decisions and subsequent happiness. *Working Paper*, 1-40.

List, J. (2011). Why economists should conduct field experiments and 14 tips for pulling one off. *Journal of Economic Perspectives*, 3-16.

Liu, Y.-E., Mandell, T., Brunskill, E., & Popovic, Z. (2014). Trading Off Scientific Knowledge and User Learning with Multi-Armed Bandits. *Proceedings of the 7th International Conference on Educational Data Mining, (pp. 161-168).*
Logg, J. (2017). Theory of machine: When do people rely on algorithms? *Working Paper*.
Mellers, B., Tetlock, P., & Arkes, H. (2018). Forecasting tournaments, epistemic humility and attitude depolarization. *Cognition*.
Mullainathan, S. (2017, Dec 1). Why Trying New Things Is So Hard to Do. *New York Times*.
Porter, M. (1991). America's green strategy. *Scientific American, 264, 168*.
Williams, J. J., Kim, J., Rafferty, A., Maldonado, S., Gajos, K., Lasecki, W., & Heffernan, N. (2016). AXIS: Generating Explanations at Scale with Learnersourcing and Machine Learning. *Learning at Scale* (pp. 1-10). Edinburgh: ACM.
Williams, J., Rafferty, A., Tingley, D., Ang, A., Lasecki, W., & Kim, J. (2018). Enhancing Online Problems Through Instructor-Centered Tools for Randomized Experiments. *Computer-Human Interaction* (pp. 1-12). Montreal: ACM.
Yeoman, M., Shah, A., Mullainathan, S., & Kleinberg, J. (2018). Making sense of recommendations. *Working Paper*.

Chapter 9 – Applications of Reinforcement Learning to Self Improving Educational Systems

Shitian Shen, Machi Shimmei, Min Chi, and Noboru Matsuda
North Carolina State University

Introduction

Interactive e-learning systems such as Intelligent Tutoring Systems (ITSs), and educational games have become increasingly prevalent in educational settings. While these systems hold great promise, they are difficult and expensive to construct and are often brittle and inflexible in their interactions with students. In order to design an effective interactive e-learning system, developers must form the core of the system and then determine *what* and *how* to teach the desired content. Many interactive e-learning systems exist for Science, Technology, Engineering and Math (STEM) domains, but they are not all capable of the adaptive pedagogical decision-making that is central to achieving the potential learning gains afforded by such systems. These limitations are due in part to the fact that they typically rely on a small set of hand-crafted rules when making pedagogical decisions. Because there are a lack of validated theories of decision-making in interactive e-learning systems, these rules are often project-specific and are rarely evaluated. Thus, there is a clear need to advance data-driven approaches to pedagogical decision-making.

Reinforcement Learning (RL) offers one of the most promising approaches to data-driven decision-making for improving student learning in interactive e-learning systems. RL algorithms are designed to induce effective policies that determine the best action for an agent to take in any given situation so as to maximize a cumulative reward. Optimal decision making in complex interactive environments is challenging. In ITSs, for example, the system's behaviors can be viewed as a sequential decision process where at each step the system chooses an appropriate action from a set of options. *Pedagogical strategies* are policies that are used to decide what action to take next in the face of alternatives. Each of these system decisions will affect the user's subsequent actions and performance. Its impact on outcomes cannot be observed immediately and the effectiveness of each decision is dependent upon the effectiveness of subsequent decisions. A number of researchers, including the authors of this chapter, have studied the application of existing RL algorithms to improve the effectiveness of interactive e-learning systems. In this chapter, we will describe two case studies on applying RL to improve the effectiveness of educational systems.

RL & Markov Decision Process (MDP) Framework

The Markov Decision Process (MDP) is one of the most widely used RL frameworks. In general, an MDP is defined as a 4-tuple $\langle S, A, T, R \rangle$, where S denotes the observable state space, defined by a set of features that represent the interactive e-learning environment; A denotes the space of possible actions for the agent to execute; T represents the transition probability where $p(s, a, s')$ is the probability of transiting from state s to state s' by taking action a. Finally, the reward function R represents the immediate or delayed feedback: $r(s, a, s')$ denotes the expected reward of transitioning from state s to state s' by taking action a. Since we apply the tabular MDP framework, reward function R and transition probability table T can be easily estimated from the training corpus. The goal of an MDP is to generate the deterministic policy $\pi : s \rightarrow a$ that maps each state onto an action.

Once the tuple $\langle S, A, T, R \rangle$ is set, the optimal policy π^* for an MDP can be generated via dynamic programming approaches, such as Value Iteration. This algorithm operates by finding the optimal value for each state $V^*(s)$, which is the expected discounted reward that the agent will gain if it starts in s and follows the optimal policy to the goal. Generally speaking, $V^*(s)$ can be obtained by the optimal value function for each state-action pair$Q^*(s, a)$, which is defined as the expected discounted reward the agent will gain if it takes an action a, in a state s and follows the optimal policy to the end. The optimal state value $V^*(s)$ and value function $Q^*(s, a)$ can be obtained by iteratively updating $V(s)$ and $Q(s, a)$ via equations 1 and 2 until they converge:

$$Q(s, a) \coloneqq \sum_{s'} p(s, a, s')[r(s, a, s') + \gamma V_{t-1}(s')] \tag{1}$$

$$V(s) \coloneqq Q(s, a) \tag{2}$$

where $0 \leq \gamma < 1$ is a discount factor. When the process converges, the optimal policy π^* can be induced corresponding to the optimal Q-value function $Q^*(s, a)$, represented as:

$$\pi^*(s) \coloneqq Q^*(s, a) \tag{3}$$

where π^* is the deterministic policy that maps a given state into an action. In the context of an ITS, this induced policy represents the pedagogical strategy by specifying tutorial actions using the current state.

Case Study 1: Apply RL to induce Pedagogical Strategy for Deep Thought

In this case study, Shen and Chi investigated the impact of both immediate and delayed reward functions on RL-induced policies using MDP framework and empirically evaluated the effectiveness of induced policies within an ITS called Deep Thought. As described above, RL focuses on inducing effective decision making policies for an agent with the goal of maximizing the agent's cumulative reward. In many domains RL is applied with immediate reward functions. In an automatic call center system, for example, the agent can receive an immediate reward for every question it asks because the impact of each question can be assessed instantaneously (Williams, 2008). Immediate rewards are generally more effective than delayed rewards for RL-based policy induction. This is because it is easier to assign appropriate credit or blame when the feedback is tied to a single decision. The more we delay the rewards or punishments, the harder it becomes to assign credit or blame properly. The most appropriate reward to use in ITSs are student learning gains which are typically unavailable until the entire training process is complete. This is due to the complex nature of the learning process which makes it difficult to assess students' learning moment by moment and more importantly, many instructional interventions that boost short-term performance may not be effective over the long-term. Therefore, in this study, Shen and Chi explored both immediate and delayed rewards in our policy induction and empirically evaluated the impact of the induced policies on student learning.

Moreover, prior research has shown that some learners are less sensitive to the learning environment and can always learn; while others are more sensitive to variations in learning environments and may fail to do so (Cronbach & Snow, 1977). We refer to the former as high learners and the latter as low learners. It is not fully understood why such differences exist. One hypothesis is that low learners lack crucial skills such as general problem-solving strategies and meta-cognition. In order to be effective and to honor the promises of learning environments, a system should support both high and low learners effectively, especially the low learners. In Case study 1, our hypothesis is that our induced pedagogical strategies may have different impacts on students with different learning competence. More specifically, in this study, we divide students into *Fast* and *Slow* groups based upon their average response time and we found that the RL-induced pedagogical strategies had significantly more impact on Slow learners than on their Fast peers. That is, Slow learners in this study behave more like low learners in that they are more sensitive to effectiveness of the RL-induced pedagogical strategies while Fast learners are more like high learners in that they can learn equally effectively regardless of the pedagogical strategies employed.

To summarize, in Case Study 1, we applied RL to induce two sets of policies: Immediate and Delayed. We focused on one important tutorial decision: whether to provide students with a Worked Example (WE) or to require them to engage in a Problem Solving (PS). Our primary research questions are: 1) would our induced policies improve students' learning? 2) which policy, Immediate or Delayed, would be more effective?

Case Study 1: Methods

Deep Thought

Deep Thought is a data-driven ITS used in the undergraduate-level Discrete Mathematics (DM) course at North Carolina State University (Mostafavi & Barnes, 2017). Deep Thought provides students with a graph-based representation of logic proofs which allows students to solve problems by applying logic rules to derive new logical statements, represented as nodes. The system automatically verifies proofs and provides immediate feedback on rule application (but not strategy) errors. Every problem in Deep Thought can be presented in the form of either Worked Example (WE) or Problem Solving (PS). In PS (shown in Figure 1) students are tasked with solving a problem using the assistantance of the ITS. In WE (shown in Figure 2) by contrast, students are given a detailed worked example showing the expert solution for the same problem. Note that by focusing on the pedagogical decisions of choosing WE vs. PS, it allows us to strictly control the content to be *equivalent* for all students.

Figure 1. Problem Solving on Deep Thought

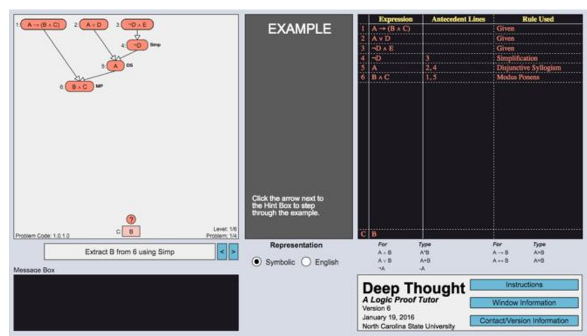


Figure 2. Worked Example on Deep Thought

Training Corpus

Our dataset was collected from Deep Thought. It included a total of 303 undergraduate CS students who used Deep Thought as part of a class assignment in Fall 2014 and Spring 2015. The average amount of time spent in the tutor was 416.60 minutes. When the students started each new training problem, Deep Thought made a simple decision: should it ask the student to solve the next problem (PS), or should it provide them with a worked example (WE). In order to model the students' learning process, we extracted a total of 134 state feature variables, which can be grouped into the following five categories:

1. **Autonomy (AM):** the amount of work done by the student: such as the number of problems solved so far *PSCount* or the number of hints requested *hintCount*.

2. **Temporal Situation (TS):** the time related information about the work process: such as the average time taken per problem *avgTime*, or the total time for solving a problem *TotalPSTime*.

3. **Problem Solving (PS):** information about the current problem solving context, such as the difficulty of the current problem *probDiff*, or whether the student changes the difficulty level *NewLevel*.

4. **Performance (PM):** information about the student's performance during problem solving: such as the number of right application of rules *RightApp*.

5. **Student Action (SA):** the statistical measurement of student's behavior: such as the number of non-empty-click actions that students take *actionCount*, or the number of clicks for derivation *AppCount*.

Inducing Immediate vs. Delayed Policies

The reward function in Deep Thought datasets is calculated based upon the level score $LevelScore_i, i \in [1,6]$, which is calculated based upon the students' performance on the last problem in each level without receiving any formative feedback from the system. As described below, students were trained either on high track or low track on each level (except on level 1) so it is hard to compare their performance directly. Therefore, a student's level score at level i is calculated based on *rank* of the student performance score at level i relative to whole population performance scores at the same level.

Our experimental results indicate that there exists a significant correlation between students' performance on last level ($LevelScore_6$) and their final test score taken at the end of the semester: R^2 = 0.396, p-value = 1.433e-11. This suggests that the students' level score indeed reflects their knowledge level.

We designed two types of reward: immediate and delayed reward with the goal of measuring the students' learning gains. The immediate reward is defined as $R_i = LevelScore_i - LevelScore_{i-1}$ where $i \in [1,6]$, $R_1 = LevelScore_1$, it reflects the change in students' performance level by level. The delayed reward is defined as $R_{delay} = LevelScore_6 - LevelScore_1$, which determines the change in students' performance across levels. For convenience, we denote the Deep Thought datasets with immediate reward as *DT-Immed* and that with delayed reward as *DT-Delay*. Note that the sum of each student's immediate rewards will equal their final delayed reward. Apart from the reward functions, both two datasets are identical.

Both Immediate and delayed policies were induced using the same general procedure. We apply the ensemble feature selection method to the corresponding dataset. In order to extract the state feature set that can represent learning context compactly and accurately, we set the maximum number of state feature size to be 8. The ensemble method comprises 6 correlation-based methods and 4 RL-based methods and more details are described in Shen and Chi (2016b). Based upon the extracted feature set, we induce our policy using the toolkit developed by Tetreault and Litman (2008). The effectiveness of policy is evaluated by Expected Cumulative Reward (ECR) (Chi, VanLehn, Litman, & Jordan, 2011b) defined as:

$$ECR = \sum_i \frac{N_i}{N} \times V^{\pi}(S_i) \tag{4}$$

where N denotes the number of initial states in training corpus, N_i means the number of state S_i as initial states. The higher the ECR value of a policy, the better the policy is supposed to perform.

Induced Immediate and Delayed Policies

Our best Immediate policy has a feature size of seven and our best Delayed policy has a feature size of six, as shown in Table 1.

Table 1. Selected Features in Immediate Policy and Delayed Policy

Feature	Definition	Category
Immediate Policy		
TotalPSTime (f_{I1})	Total time for solving a problem	Temporal Situation
NewLevel (f_{I2})	Whether current solved problem is in the new level	Problem Solving
WrongApp (f_{I3})	Number of wrong application of rules	Performance
TotalWETime (f_{I4})	Total time for working on an example	Temporal Situation
UseCount (f_{I5})	Number of different types of applied rules in Use category	Problem Solving
AppCount (f_{I6})	Number of clicks for derivation	Student Action
NumProbRule (f_{I7})	Number of expected distinct rules for a solved problem	Problem Solving
Delayed Policy		
stepTimeDev (f_{D1})	Step time deviation	Temporal Situation
probDiff (f_{D2})	Difficulty of current solved problem	Problem Solving
symbolicRCount (f_{D3})	Number of whole problems for symbolic representation	Problem Solving
actionCount (f_{D4})	Number of non-empty-click actions that students take	Student Action
SInfoHintCount (f_{D5})	Number of System Information Hint	Student Action
NSClickCountWE (f_{D6})	Number of next step click in Work Example	Student Action

Immediate policy used seven features f_{I^*}, which are listed in Table 1. It is interesting to note that the Immediate policy contained features from every category except Autonomy. Table 2 shows the induced Immediate policy. Each row of table represents one combination of the first four features and each column represents the combination of final three. The black cells indicate that the tutorial action that is associated with the state is PS, while the white cells indicate the action is WE. The gray cells indicate that no rule was learned for the state. There are a total of 86 rules for the Immediate policy, of which 21 are associated with PS and 65 are associated with WE. More specifically, we found that the states in the row 0:0:1:0 almost always associate with PS; while the states in rows 0:0:0:0, 0:0:0:1, 1:1:1:0 almost always relate to WE; the rules in rows 0:1:0:0, 0:1:0:1, 0:1:1:0, 1:1:0:0, 1:1:1:1 do not contain PS; there are no rules for row 0:1:1:1. In general, the Immediate policy favors WE.

Table 2. Immediate Policy

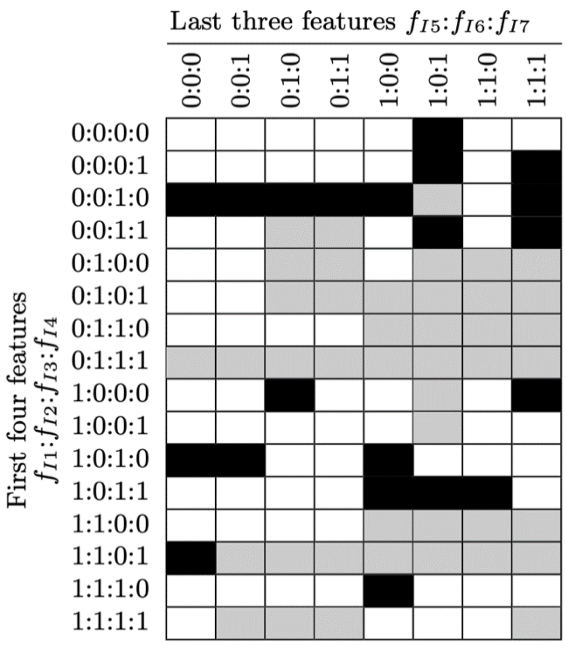
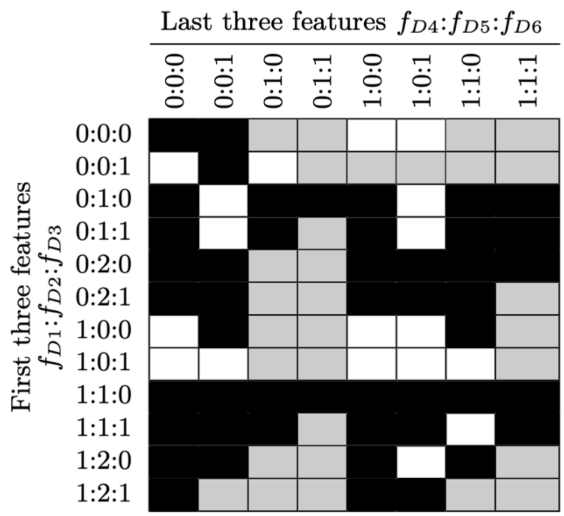


Table 3. Delayed Policy



Delayed policy used six features f_{D^*}, which are listed in Table 1. They are drawn from the Temporal Situation, Problem Solving and Student Action categories. No features are drawn from the Autonomy and Performance categories. Table 3 shows the Delayed policy, where each row represents one combination of the first three features and each column represents one combination of last three. Each cell has the same meaning as described above. There are a total of 68 rules for the Delayed policy, of which 48 are associated with PS and 20 are associated with WE. More specifically, states in row 1:1:0 only associate with PS; states in rows 0:0:1, 0:1:1, 1:1:1 almost always correspond to PS; while the rules in row 1:0:1 only contain WE. Therefore, the Delayed policy is more likely to take PS.

Immediate Policy vs. Delayed Policy. The *ECR* of the best Immediate policy was 137.97 while the *ECR* of the best Delayed policy was 14.06. This is likely due to the credit assignment problem. The more we delay success measures from a series of sequential decisions, the more difficult it becomes to identify which of the decision(s) in the sequence are responsible for our final performance. Furthermore, this may explain why, for the slow learners, the low-Immediate students learned more than the low-Random and the low-Delayed students. The former difference was statistically significant while the latter was marginal. We found no significant difference between the low-Random and low-Delayed students.

Case Study 1: Experiment

Our primary goal of Case Study 1 was to empirically evaluate the effectiveness of the RL induced policies. In order to do so, we incorporated them back into the Deep Thought tutor and empirically compared them against a baseline policy that makes random decisions. Thus, we have three conditions: Immediate, Delayed and Random. Apart from the differing policies, the remaining components of the system, including the Graphical User Interface (GUI), the training problems, and the tutorial scripts, were the same for all three conditions. Next, we will describe our experimental details.

Participants and Conditions

The study was conducted in "Discrete Mathematics for Computer Science"', a course offered at North Carolina State University in the Spring of 2016. 106 undergraduate students were assigned to complete the task as one of their regular homework assignments[2].

The participants were randomly distributed into three conditions. The group sizes were as follows: $N = 30$ for Random, $N = 38$ for Delayed, and $N = 38$ for Immediate condition. A total of 98 students completed the experiment and were distributed as follows: $N = 28$ or Random, $N = 37$ for Delayed, and $N = 33$ for Immediate. We performed a x^2 test of independence to examine the relationship between completion rate and condition. We found no significant differences among three groups: $x^2(2, 106) = 1.4,\ p = 0.49$.

Performance Measure

When inducing both the Immediate and Delayed policies, we calculated our reward function based upon the students' level scores $LevelScore_i, i \in [1,6]$. Both Immediate and Delayed policies are induced to maximize the students' improvement from level 1 to level 6. This can be calculated as: $LevelScore_6 - LevelScore_1$. Here we treat $LevelScore_6$ as the posttest score, $LevelScore_1$ as pretest score, and calculate each student's learning gain as $LevelScore_6 - LevelScore_1$. In addition to raw learning gains, we also used the normalized learning gain (NLG) as the reward value. This measures students' learning gain by considering their *incoming competence* and has been widely used for measuring student learning performance in the field of ITSs. The NLG is defined as: $NLG = (posttest - pretest)/(Max(posttest) - pretest)$, that is, how much did the student learn given how much he/she can learn. In this study, we calculated NLG as: $NLG = \frac{LevelScore_6 - LevelScore_1}{Max(LevelScore_6) - LevelScore_1}$, Here $Max(LevelScore_6)$ is the maximum score a student can get. We will report our students' performance using both the raw learning gain and the NLG. Both scores were normalized to [0,1].

Case Study 1: Results

As expected, no significant difference were found among the three conditions in terms of their level 1 score: $F(2, 97) = 0.04, p = 0.96$. To investigate our hypothesis that the induced pedagogical strategies may have different impacts on students with different learning competence, we further divided students into Fast ($n = 49$) and Slow ($n = 49$) groups based upon their average response time on Level 1. As expected, there was a significant difference between the Fast and Slow students on $LevelScore_1$: $F(1, 98) = 10.05, p = 0.002$ in that the Fast students had significantly higher scores than the Slow ones. Combining the three conditions with incoming competence (Fast vs. Slow), we partitioned all students into six groups: Immediate-Fast ($n = 16$), Immediate-Slow ($n = 17$), Delayed-Fast ($n = 17$), Delayed-Slow ($n = 20$), Random-Fast ($n = 16$), and Random-Slow ($n = 12$). No significant difference was found among the Slow groups on their $LevelScore_1$: $F(2, 46) = 0.56, p = 0.58$. Nor did we find any significant difference among the three Fast groups on their $LevelScore_1$: $F(2, 46) = 0.64, p = 0.53$.

A two-way ANOVA based upon Condition {Immediate, Delayed, Random} and Incoming Competence {Fast, Slow} showed no significant differences among the three conditions on overall training time: $F(1, 98) = 0.13, p = 0.87$. However there was a significant Incoming Competence effect: the fast learners spent less time on task than the slow learners: ($M = 387, SD = 81$) for fast learners and

[2] Although participation were not required, most of students complete the assignment.

($M = 662, SD = 83$) for the slow learners and the difference was significant: $F(1, 98) = 5.54, p = 0.02$. In addition there was no interaction effect. Therefore, we can conclude that the fast learners spend less time on task than the slow learners across all three conditions.

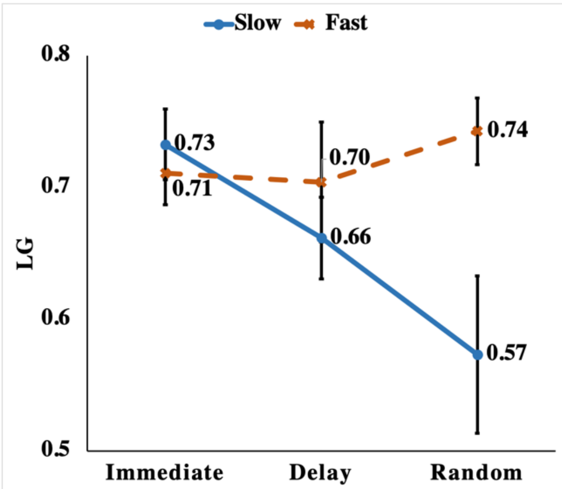
Learning Performance

Next we will report the impact of RL policies on students' performance and then discuss the characteristics of the induced policies. We performed two one-way ANOVAs using condition as the factor and the student's raw learning gain or NLG as the dependent measure respectively. We found no significant difference among the three groups: $F(2, 97) = 1.54, p = 0.22$ for raw learning gains and $F(2, 97) = 2.15, p = 0.12$ for NLG scores. While no significant difference was found, the comparatively large *SD* suggests that both fast and slow students may benefit differently from the induced policies. For example, on the raw learning gain scores, $M = 0.72, SD = 0.10$ for the Immediate group, $M = 0.68, SD = 0.16$ for the Delayed group, and $M = 0.67, SD = 0.17$ for the Random group. The same pattern was repeated for NLG.

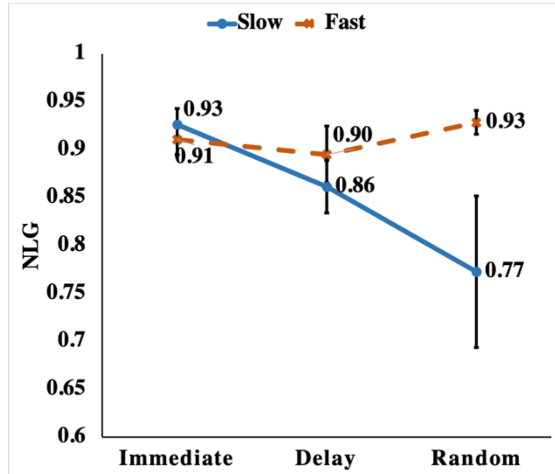
A two-way ANOVA using Condition {Immediate, Delayed, Random} and Incoming Competence {Fast, Slow} as two factors and the student's raw learning gain or NLG as the dependent measure showed an significant interaction effect, $F(2, 97) = 3.43, p = 0.037$ for the raw learning gain and $F(2, 97) = 3.48, p = 0.035$ for NLG. Additionally, we also found a significant main effect from the *Incoming Competence*: $F(1, 98) = 4.68, p = 0.033$ for the raw learning gain and $F(1, 98) = 4.90, p = 0.029$ for the NLG. Therefore the fast learners learned significantly more than the slow learners: $M = 0.719, SD = 0.14$ for the fast learners vs. $M = 0.656, SD = 0.145$ for the slow learners on the raw learning gain. In other words, this results confirmed our assumption that Fast learners can be seen as the high learners while Slow learners can be seen as the low learners. Finally, the main effect of Condition was not significant: $F(1, 98) = 1.54, p > 0.05$.

Figure 3 shows that the raw learning gain and NLG results are consistent with our hypothesis: no significant difference was found among the three fast groups on either the raw learning gain or NLG: $F(2, 46) = 0.38, p = 0.69$ for the raw learning gain and $F(2, 46) = 0.64, p = 0.53$ for NLG respectively.

On the other hand, Figure 3 shows a significant difference among the three slow groups: $F(2, 46) = 3.99, p = 0.025$ for the raw learning gain and $F(2, 46) = 3.22, p = 0.049$ for NLG. Pairwise t-tests showed that the Immediate-Slow group significantly outperformed the Random-Slow group on both measures: $t(27) = 2.69, p = 0.012$ for the raw learning gain and $t(27) = 2.23, p = 0.034$ for NLG. The Immediate-Slow group outperformed the Delayed-Slow group on both measures. However, these differences were only marginally significant: $t(35) = 1.67, p = 0.098$ for the raw learning gain and $t(35) = 1.94, p = 0.06$ for NLG respectively. Furthermore, we found no significant differences between the Delayed-Slow and the Random-Slow groups. Thus, our results suggest that all three Fast groups learned equally well after training on DT while the Slow learners are indeed more sensitive to induced policies. For three Slow groups, Immediate policies significantly outperforms Random ones and there is a trend that the Immediate policies beat the Delayed ones while no significant difference between Delayed and Random.



(a) Learning Gain



(b) NLG

Figure 3: Learning Performance across three groups

Finally, we compared the fast and slow groups across all three conditions. For the Immediate condition, we found no significant differences among the Immediate-Fast and Immediate-Slow groups on either the raw learning gain or NLG: for the $F(1,33) = 0.38, p = 0.55$ raw learning gain and $F(1,33) = 0.45, p = 0.51$ for NLG. Likewise, for the Delayed condition, no significant difference was found between the Fast and Slow groups: $F(1,37) = 0.62, p = 0.43$ for the raw learning gain and $F(1,37) = 0.71, p = 0.41$ for NLG. Therefore, both fast and slow groups learned equally well when following the RL induced policies.

For Random group, however, the Random-Slow students learned significantly less than their Random-Fast peers: $F(1,27) = 8.18, p = 0.008$ for the raw learning gain and $F(1,27) = 5.03, p = 0.034$ for NLG respectively.

Overall, our results suggest that the Fast learners are not sensitive to the effectiveness of the pedagogical strategy while the Slow learners will learn more with the effective pedagogical strategy. We found that for the Slow learners the Immediate policy is more effective than the Random policy, and is marginally more effective than the Delayed policy.

Log Analysis

Having compared the individual student's learning performance and the characteristics of the induced policies, this subsection will compare the log file variations across the conditions. More specifically, we focused on the total number of problems that students encountered (TotalCount); the total number of problems that the students solved (PSCount); the total number of WEs reviewed (WSCount); the total number of difficult problems that the students solved (DiffPSCount); and the total number of difficult WEs (DiffWECount).

Table 4 shows comparisons of the different counts across the conditions. A two-way ANOVA using Condition {Immediate, Delayed, Random} and Incoming Competence {High, Low} on all behavior counts showed a significant main effect for the Condition on all measures. Both the main effect of Incoming Competence and the interaction effect were not significant. Additionally, Table 4 summarizes one-way ANOVA comparisons on the different counts among the three conditions. Columns 2-4 list the three groups in comparison and their corresponding mean and SD scores. The last column lists the statistical results of the one-way ANOVA comparisons. Table 4 shows that the Immediate group solved significantly fewer total problems than the Delayed group. The former solved significantly fewer difficult problems than the other two groups. On the other hand, the Immediate group studied significantly more worked examples on both total and difficult problems than the other two groups.

Table 4. Statistical measurement of several features among three groups

Feature	Immediate	Delay	Random	Significance	
TotalCount	19.42 (2.07)	20.73 (2.08)	19.81 (2.09)	$F(2,98) = 3.67$	$p = 0.03$
PSCount	8.09 (1.37)	13.41 (1.38)	12.34 (1.39)	$F(2,98) = 140.14$	$p = 0.00$
diffPSCount	4.21 (1.90)	7.83 (1.91)	6.74 (1.92)	$F(2,98) = 32.60$	$p = 0.00$
WECount	11.33 (1.32)	7.32 (1.32)	7.47 (1.33)	$F(2,98) = 97.11$	$p = 0.00$
diffWECount	7.78 (1.69)	4.38 (1.70)	5.50 (1.71)	$F(2,98) = 35.78$	$p = 0.00$

Case Study 1: Summary

In this study, Shen and Chi investigated the impact of different reward functions (Immediate vs. Delayed) on the effectiveness of the induced RL policies. Our results show that the two policies include substantially different state features and that the policies generate different patterns of decisions. The Immediate policies are more likely to give worked examples while the Delayed policies are more likely to require problem-solving. Additionally, the Expected Cumulative Reward (ECR) for our immediate policies was an order of magnitude higher than the delayed ECR.

We also investigated the impact of RL-induced policies on different groups of learners. We divided students into fast and slow learners based upon their average response time in the first level. Our results support our hypothesis that the fast learners in all three conditions learned more than their slow peers and that there was no meaningful differences among them across three conditions; the slow learners, on the other hand, were more sensitive to the learning environment. The Random-Slow students learn the least while the Immediate-Slow group learned the most and in fact, it learned as much as their Immediate-Fast peers. Indeed, the Immediate-Slow students learned significantly more than the Random-Slow students, and marginally more than the Delayed-Slow students. Therefore, the Immediate policies appear to be more effective than the Delayed policies and are significantly better than the Random policy.

Finally, our preliminary log analysis showed that students using the Immediate policies studied significantly more worked examples, in terms of both total and difficult problems, than the other two groups. And more importantly, they solved significantly fewer problems, especially difficult problems. Previous research on WE versus PS has primarily relied on fixed or hand-coded adaptive rules to decide whether to present the next question as a WE or PS, this is the first study in which we applied RL to induce adaptive pedagogical strategies directly from students' logs to decide whether to present the next question as a WE or PS. We showed that the induced policies are indeed effective at improving students' learning especially for Slow learners.

Case Study 2: Apply RL To Evidence-based Learning Engineering

In the current case study, Shimmei and Matsuda proposed a transformative method for evidence-based learning engineering that provides a foundation of self-improving online courseware. Building effective online courseware is remarkably costly and requires intensive knowledge in the theory of learning and teaching (Clark & Mayer, 2003; Slavich & Zimbardo, 2012). Like other software engineering, the iterative design-engineering-testing cycle (Fishman, Marx, Blumenfeld, Krajcik, & Soloway, 2004) is a norm to improve the effectiveness of the online courseware. In the current study, we focus on developing a device that automatically identifies a weakness of the online courseware content.

The proposed method, called RAFINE (Reinforcement learning Application For INcremental courseware Engineering), identifies a portion of existing online courseware that is relatively ineffective given students' learning activity and achievement data. Toward an effort of developing a self-improving online course technology, the current study focuses on our preliminary attempt to create a machine-assisted, human-centered method where a machine identifies issues on existing online courseware and provides human experts (e.g., courseware designers and developers) with insights into a next iteration of system improvement.

We presume that a record of activities that students demonstrated on the target online courseware can be converted into a state transition graph whose states represent students' internal learning status and edges represent the learning activities; each corresponds to taking an instructional element such as watching a video or answering a formative assessment, that were taken at certain states. We then hypothesize that the task of identifying ineffective instructional elements is solved by applying RL to the state transition graph. An intuition behind the current hypothesis is that since a policy computed by RL should indicate an optimal instructional element for a given state, instructional elements that frequently appear in a policy are likely to be effective relative to actual students' learning indicated in the log.

The majority of previous works on an application of RL have focused on computing optimal pedagogical decisions including hint messages (Martin & Arroyo, 2004), dialogue moves (Chi, VanLehn, Litman, & Jordan, 2011a; Tetreault, Bohus, & Litman, 2007), learning activities (Shen & Chi, 2016a), and navigation (Iglesias, Martinez, Aler, & Fernandez, 2009). RL has been applied to content generation such as model solution for logic proofs (Barnes & Stamper, 2008). To the best of our knowledge, no research has been applied to content validation for online courseware.

RAFINE: Technical Overview

RAFINE is a learning engineering method to compute the effectiveness of instructional elements of online courseware based on students' performance and learning outcomes. Given *learning log data* that shows students' learning activities, the RAFINE method converts the log data into a state transition graph ***G*** to apply the RL to compute a *converse policy* that shows the worst policy to take at any given state in the graph ***G***. This section describes details of a model representation and how RL is used.

Model Representation

RAFINE is applied to existing online courseware where learning log data will be collected through actual students' use. The learning log data contains a chronological record of individual students' behavior on the target online courseware—e.g., clickstream data showing page visits and students' response to quiz items annotated with the correctness. The aggregated students' learning activities are then represented as a directed graph with nodes representing *learning status* and edges representing *learning activities* as described below.

Let $\boldsymbol{E}$ be a set of instructional elements used in the target online courseware. $\boldsymbol{E}$ might include written paragraphs, videos, and assessment items (aka quizzes). Let a *learning activity* be an instructional element taken by a particular student at a particular time while learning on the online courseware. The *learning log* for student i (who had n_i activities) is a chronological record of learning activities: $\{a_1, a_2, \dots, a_{n_i} \mid a_k \in E,\ k = 1,2, \dots n_i\}$.

We assume the presence of a *skill model* that contains a set of skills (aka knowledge components) that represent a unit of knowledge that students should learn (Koedinger, Corbett, & Perfetti, 2012). That is, we assume that each instructional element on a target online courseware is tagged with a single skill. The RAFINE method is then applied for each individual skill separately.

Given a particular skill φ, we define learning status for student i at time T as an intermediate state of learning represented as a tuple <Page ID, Action History, Mastery Level> = $<pid, \boldsymbol{ah}_i^T, p_i^T(\varphi)>$. Page ID, *pid*, is a physical location of the courseware where a particular learning activity occurred (e.g., a page on which a video was watched). Action History $\boldsymbol{ah}_i^T$ is a binary vector $<ie_1^{i,T}, ie_2^{i,T}, \dots>$ with $ie_k^{i,T}$ showing whether student i has taken the instructional element ie_k by time T. Mastery Level $p_i^T(\varphi)$ is a scalar showing a predicted probability of student i applying skill φ correctly at time T.

In the current implementation, edges represent learning activities each of which represents an explicit use of instructional elements—the actual inclusion of instructional elements depends on the implementation, but the current study only includes watching a video and taking a quiz. Therefore, following a link to move to another page does not appear as a state transition. The Page ID of any given state represents a physical page on which the previous instructional element (i.e., the one that corresponds to the incoming edge) was taken. This implies that states may have more outgoing edges than the number of instructional elements available on the page specified with the Page ID of the state. Since students may take the same instructional element multiple times, there are also states with a loop back edge (since the Action History, which is a binary vector, does not change in this case).

Converting learning log data into a state transition graph $\boldsymbol{G}$ is straightforward. For each student, transactions in the learning log are chronologically traversed while counting the number of each type of learning activities to update $\boldsymbol{ah}_i^T$. Mastery Level is computed using an extension of Additive Factor Model (AFM). The original AFM (Cen, Koedinger, & Junker, 2006) is a logistic regression model for $p_i^T(\varphi)$ as a monotonic function of T that is a number of times the student answered a quiz for a particular skill. In the current model, Mastery Level is also increased when the student took another type of learning activity (i.e., watching a video and reading a written paragraph). In the current implementation, the amount of inflation is given a priori in an ad-hoc fashion. To reduce the number of states in the graph $\boldsymbol{G}$, the value of the Mastery Level is rounded to the nearest multiple of 0.05.

Once individual students' learning activities are converted into state transition graphs, they are aggregated into a consolidated state transition graph $\boldsymbol{G}$ by merging the same states. As a consequence, student IDs and time (i.e., the parameters i and T used to model an individual student's state transition graph), and states in G generally have multiple incoming and outgoing edges. States in G are therefore denoted as $<pid, \boldsymbol{ah}, p(\varphi)>$. States in G with Mastery Level higher than 0.85 are called *terminal states*. All outgoing edges at terminal states are discarded.

Reward

In the current model, a reward for reaching to a state *s* depends on how Masterly Level (ML) changed. For example, a student could have reached the state *s* by answering a quiz correctly (hence ML was increased from the preceding state to the current state) or incorrectly (ML was decreased). We would like to differentiate these two scenarios.

In the current model, a reward for the state s, $R(s)$, with n incoming edges (therefore n preceding states) is represented as a n-dimensional vector $R(s) = \{r(s'_i, s) | i = 1, \ldots, l\}$ where s'_i represents the i-th preceding state. Since we assume that it is preferable to achieve to the mastery quicker, the reward value $r(s'_i, s)$ is set to be small negative except for the terminal state as shown below. *ml*(s) represents Mastery Level at state s:

$$r(s'_i, a, s) = \begin{cases} -0.05 & \left(ml(s'_i) < ml(s)\right) \\ -0.14 & \left(ml(s'_i) = ml(s)\right) \\ \ \ 0.95 & \left(0.85 < ml(s)\right) \end{cases}$$

Converse Policy

With the above-mentioned reward, a utility function for state s coming from state s' given a policy π is defined as follows:

$$U^{\pi}(s', s) = r(s', s) + \gamma \sum_{s^*} P(s^* | s, a) U^{\pi}(s, s^*)$$

The probability $p(s' | s, a)$ will be derived from the learning log data collected from students who have actually used the corresponding online courseware—the ratio of students who reached the state s' by taking the instructional element a at the state s relative to all students who took the instructional element a at the state s.

Usually, a policy implies actions that maximize the expected utility. However, for the purpose of RAFINE, we need to know which instructional element (i.e., the "action") should not be taken at any given state. Therefore, we propose to modify the value iteration as follows:

$$U_{i+1}(s', s) \leftarrow r(s', s) + \gamma \min_{a \in A(s)} \sum_{s^*} P(s^* | s, a) U_i(s, s^*)$$

We then further propose to compute a *converse policy* that suggests the least effective actions:

$$\pi(s) = \operatorname*{argmin}_{a \in A(s)} \sum_{s^*} P(s^* | s, a) U^{\pi}(s, s^*)$$

Case Study 2: Evaluation

Our primary research question is to understand whether or not the converse policy would accurately identify ineffective instructional elements relative to students' learning outcomes. We hypothesize that instructional elements that frequently appear as a converse policy across many different states are likely to be ineffective; therefore, the converse policy would be an effective tool for the evidence-based iterative courseware improvement.

To test this hypothesis, we conducted an evaluation study with the generated hypothetical learning log data of 10,000 simulated students. A mock version online courseware was created with three pages (Page 0, 1, 2) each containing five to six instructional elements. Two types of instructional elements were video and quiz. There were a total of eight quiz items and eight videos. All simulated students started from Page 0. They had to take at least three instructional elements to move to any different page. Students were able to take the same instructional elements multiple times. In the current study, simulated students randomly took a total of 10 to 15 instructional elements, but not more than nine per page.

Mastery Level was simulated by assuming a model of logistic regression. Instead of regressing the model parameters, logit was computed in an ad-hoc fashion. Each time a simulated student took an instructional element, logit was updated based on two variables—*potential* of learning for individual students and *effectiveness* of instructional elements. The potential of learning was represented either as "High" or "Low." We presumed that the "High" potential students reached to a mastery level quicker than the "Low" potential students. In the current simulation study, 75% of the students were classified as "High" potential. The effectiveness of instructional elements was represented either as "Effective" or "Ineffective." We presumed that students' learning was facilitated more when they took "Effective" instructional elements than "Ineffective" ones. In the simulation study, 63% (5 out of 8) of quiz items and videos were assumed to be "Effective."

Given individual students' potential and instructional elements' effectiveness, students' Mastery Level was computed using the sigmoid function as a probability that each individual student answers a next quiz item (at time *t*) correctly:

$$p^t(\sigma) = \left[\frac{1}{1+e^{-Z^t}}\right]_{0.05}$$

The logit Z^t is a linear function of the learning history: $Z^t = Z^{t-1} + \delta$ where δ is a constant (as shown in Table 5) given a priori based on the potential and effectiveness values mentioned above.

Table 5. The amount of increase in the logit as a function of the "potential" and "effectiveness" constructs.

		Effectiveness of Instructional Element	
		Effective	Ineffective
Student's Potential	High	0.20	0.03
	Low	0.15	0.001

Table 6 is a summary of states in the simulation study showing the number of states in the state transition graph with specific number of outgoing edges. About 2/3 of states only have one outgoing edge—i.e., there was only one simulated student who visited the state.

Table 6. A summary of states generated in the simulation study.

Num. of outgoing edges	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Total
Num. of states	35106	7413	2152	1094	776	457	252	145	71	57	44	9	5	2	3	66	47652
% to total	0.74	0.16	0.05	0.02	0.02	0.01	0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	1.00

Case Study 2: Results

For the following analysis, let *IE-Ratio* be the ratio of ineffective to effective instructional elements. When IE-Ratio is applied to a set of outgoing edges, it represents a ratio of ineffective to effective instructional elements representing actions associated to those outgoing edges. When IE-Ratio is applied to a converse policy, it represents a ratio of actions (identified as a converse policy) that correspond to ineffective to effective instructional elements.

We first hypothesized that the accuracy of a converse policy as a detector for ineffective instructional elements depends on the IE-Ratio among the outgoing edges—i.e., the more the ineffective instructional elements contained in outgoing edges, the higher the probability of the converse policy identifying ineffective instructional elements. To test this hypothesis, we plotted the IE-Ratio for a converse policy relative to a group of states that have the same IE-Ratio among the outgoing edges. Figure 4 shows the plot. In the plot, each data point represents a set of states in the transition graph ***G*** that have the same IE-Ratio among the outgoing edges as indicated on the X-axis. The Y-axis shows the IE-Ratio for the converse policy relative to the states—i.e., the ratio of states where an ineffective instructional element is selected as a converse policy to the total number of states in a given data point. The 45-degree line shows a chance rate.

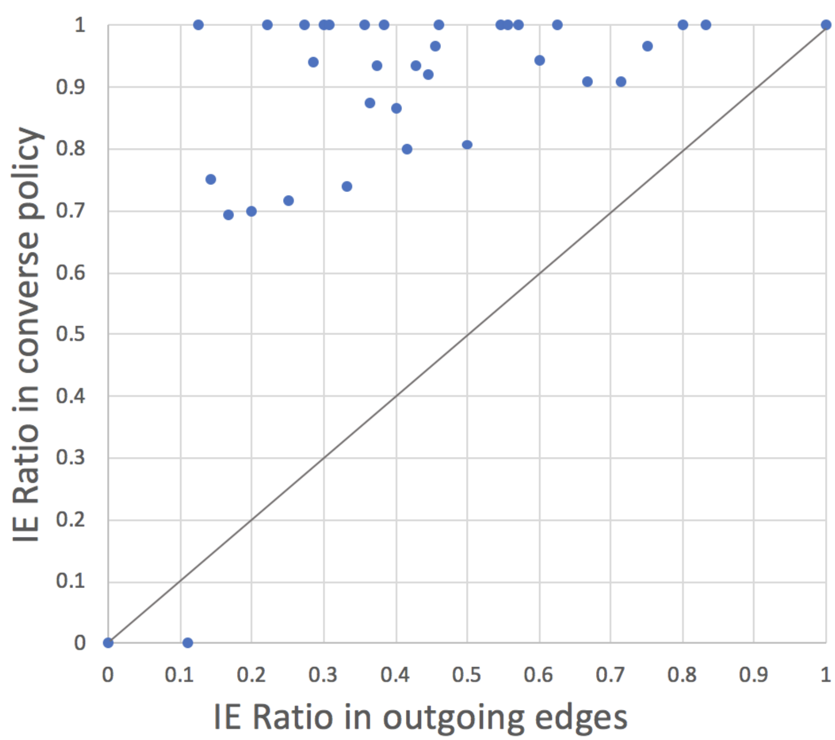


Figure 4: The overall accuracy of a converse policy identifying ineffective instructional elements. Each data point represents a collection of states where IE-Ratio among outgoing edges are shown in the X-axis.

Figure 4 apparently suggests that our hypothesis is not supported. The converse policy accurately identifies ineffective instructional elements regardless of the IE-Ratio among the outgoing edges. It is remarkable that even those states where less than 20% of outgoing edges correspond to ineffective instructional elements, the converse policy selected ineffective instructional elements with the accuracy of 70% and higher.

Next, we hypothesized that the accuracy of a converse policy depends on the number of outgoing edges—there is a sweet spot on the number of outgoing edges relative to the number of unique instructional elements where the accuracy of prediction becomes high. As a reminder, a state may have multiple outgoing edges all with a same action (i.e., instructional element). At an extreme case, if a state has only one outgoing edge, the instructional element corresponding to the edge is inevitably selected as a converse policy. Figure 5 shows a relationship between the IE-Ratio for the converse policy (Y-axis) and the number of unique actions among outgoing edges (X-axis). In the plot, numbers appearing at data points show the number of states—e.g., there were 66 states that had 16 unique actions among the outgoing edges. Figure 5 shows a trend that the more the number of unique actions among outgoing edges, the better the IE-Ratio of the converse policy. A regression analysis with the IE-Ratio of the converse policy as a dependent variable and the number of unique actions among outgoing edges as an independent variable reveals that the number of unique actions among outgoing edges has a statistically significant predictive power for the IE-Ratio; coefficient = 0.63, z = 51.85, $p(>|z|) < 0.001$.

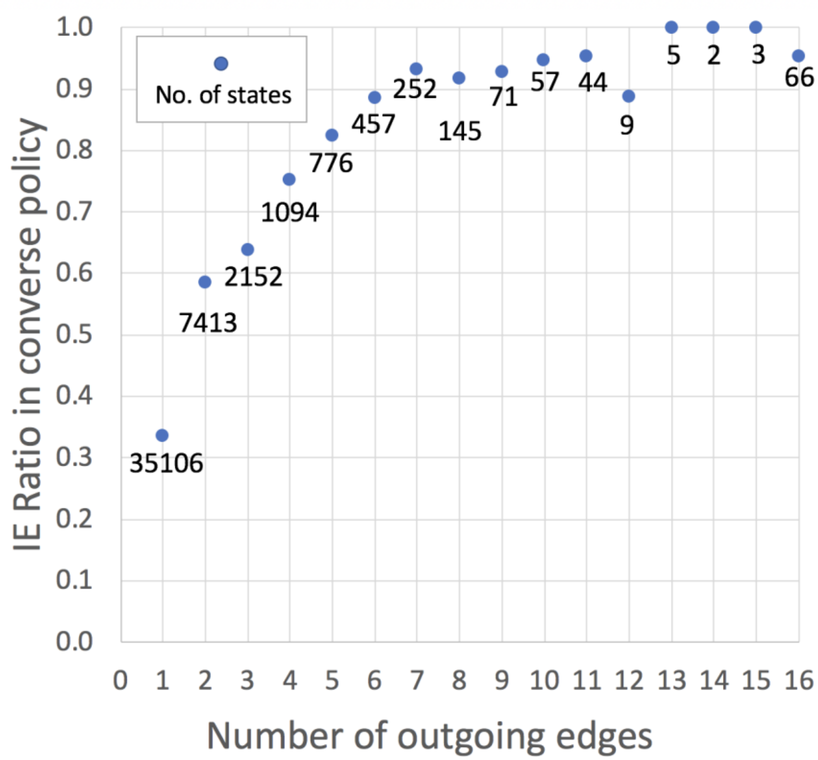


Figure 5. The relationship between the number of unique actions among outgoing edges (X-axis) and the accuracy of a converse policy, i.e., IE-Ratio (Y-axis).

A practical question when applying the RAFINE method to an actual online-courseware analysis is how the courseware developer should determine which instructional elements must be revised given a converse policy. There are states in the transition graph G where all outgoing edges corresponds to effective instructional elements. Yet, a converse policy must select one of them as a (relatively) worse action to take. We hypothesized that those actions that frequently appear as a converse policy are more likely to be ineffective instructional elements. Figure 6 shows the frequency of individual instructional elements (listed on X-axis) selected as a converse policy among states that have 16 unique actions. In Figure 6, the hatched and solid bars show effective and ineffective instructional elements respectively. The figure shows that ineffective instructional elements are 35 times more likely to be selected as a converse policy than effective instructional elements. The difference in the mean frequency between effective and ineffective instructional elements was statistically significant: $M_{effective}$ = 0.3±0.2 vs. $M_{ineffective}$ = 10.5±10.3, $t(5) = 7.73$, $p < 0.05$, d = 3.17.

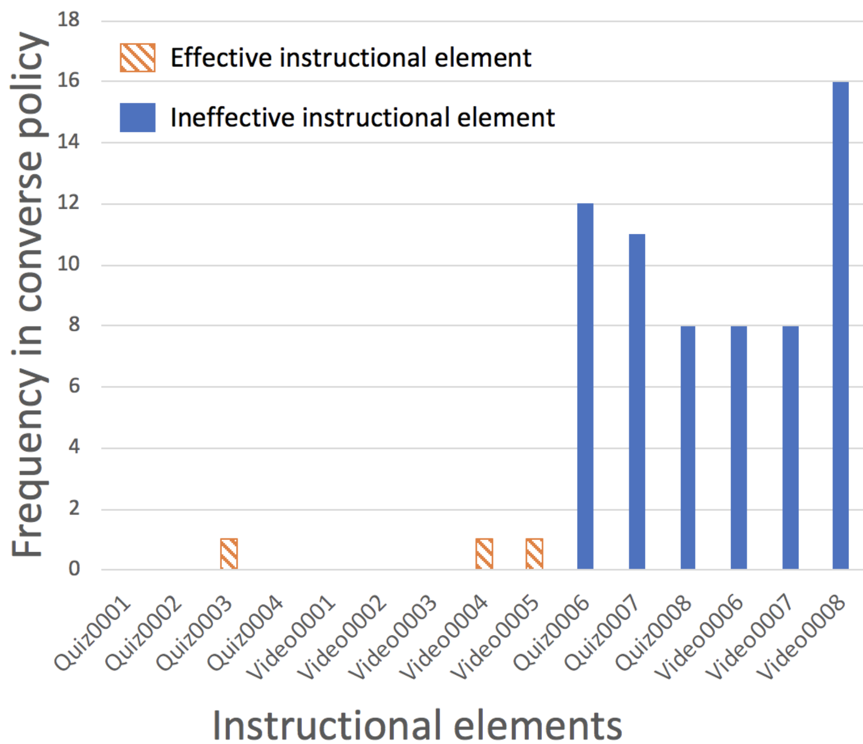


Figure 6. The frequency of individual instructional elements selected as a converse policy among states that have 16 unique actions.

Case Study 2: Summary

The current simulation study provided an empirical support for the hypothesis that RL, with a twist for the converse policy, would function as a tool for detecting ineffective instructional elements in existing online courseware with students' learning activity log data. To our surprise, our first hypothesis—the more outgoing edges for ineffective instructional elements a state has, the higher the accuracy of a converse policy detecting ineffective instructional elements—was not supported. If there is an ineffective instructional element(s) that students took at any certain states, it is highly likely that it is selected as a converse policy.

Our second hypothesis—there is a right amount for the number of unique instructional elements among the outgoing edges that maximize the accuracy of a converse policy. The current data indicate that the accuracy of the converse policy increases as the number of unique instructional elements per state increases. This finding suggests to us to only pay attention to the converse policy at states that have the largest number of unique instructional elements (i.e., 16 in the current study).

We found that, provided learning log data showing students activities on a given online courseware, the RL technique can be used to identify ineffective instructional elements on a given online courseware. The learning log data are translated into a state transition graph where states represent students' intermediate learning status and edges represent instructional elements (i.e., "action") students took. The value iteration technique is used to compute the least optimal action (called the converse policy) while finding actions that minimize expected utility.

As the popularity of MOOCs (Massive Open Online Courses) are rapidly increasing, studying transformative methods for evidence-based learning-engineering is critical for the future of self-improving online education. The proposed RAFINE method provides courseware designers and developers with insights into the iterative improvement of the courseware contents based on authentic students' activities and learning outcomes. Creating self-improving online courseware at scale requires various different scaffolding for designers and engineers. The current study made a contribution toward this line of research in a form of the machine-assisted, human-centered learning engineering. Further research is needed to enhance the integrated development environment for online course engineering where a machine suggests potential revision to overcome the deficits.

Recommendations and Future Research

There are several caveats in Case Study 1 which provide enlightenment regarding future work. First of all, we retrospectively split students into Fast vs. Slow groups using response time because we do not fully understand why the differences between Fast vs. Slow groups exist. To answer such a question, we need to perform deep log analysis for our future work. Second, although we detect different performance among the different RL-induced policies, it is still not clear what makes them effective or why they are effective. Future work is needed to shed some light on understanding the induced policies and to compare the machine induced policies with existing learning theory. Third, we mainly compare the RL-induced policies with a Random policy in our experiments and it is not clear if the same results would hold if we compare them against a stronger baseline such as those used in previous research (Salden, Aleven, Schwonke, & Renkl, 2010; McLaren, van Gog, Ganoe, Yaron, & Karabinos, 2014; Najar, Mitrovic, & McLaren, 2014). Furthermore, previous work (Renkl, 2002; Atkinson, Renkl, & Merrill, 2003; Gerjets, Scheiter, & Catrambone, 2006; Taylor, O'Reilly, Sinclair, & McNamara, 2006) has shown that adding self-explain steps in WE and PS (prompting for self-explanation) can significantly improve students learning. In the future, we will expand our research scope on not only WE vs. PS but also on whether or not to ask students to self-explain.

To effectively apply the RAFINE method to learning log data, students must have freedom to determine next instructional element to take. If the online courseware is exceedingly linear and everybody follows the same path, then the policy becomes of no use. A recommendation for Generalized Intelligent Framework for Tutoring (GIFT) authors is therefore to make a balance between a linearity of the courseware (to lower the cognitive load on the self-navigation) and a diversity in the learning path (to gain more analytic power for RL). If the courseware is noticeably linear and the majority of the students follow page by page in the same order, then a transition graph G has only a small number of unique paths hence the converse policy will be computed without competition. Allowing students to revisit pages and retake instructional elements increases the appearance of those instructional elements in the transition graph G in various different paths. This in turn results in having states with a large number of outgoing edges, hence it increases the chance of correctly detecting ineffective instructional elements.

Although, a key for a successful application of the RAFINE method is likely rooted in the diversity among students' activities, it is not clear, how the degree of diversity affects the accuracy of the prediction. Future research is needed to understand the relationship between the effort of making the online courseware less guided (i.e., giving students more independence to freely traverse courseware content) and the effectiveness of the RAFINE method to identify issues on the courseware.

Another future research question is to explore how the individual students' differences affect the "effectiveness" of the instructional elements. Instructional elements that are effective for one group of students might be ineffective for other group of students. We currently assume that big data override individual human factors and reveals a genuine trend (if any). Further research is needed to understand how to integrate individual differences in the model.

References

Atkinson, R. K., Renkl, A., & Merrill, M. M. (2003). Transitioning from studying examples to solving problems: Effects of self-explanation prompts and fading worked-out steps. *Journal of Educational Psychology*, *95*(4), 774.

Barnes, T., & Stamper, J. C. (2008). Toward automatic hint generation for logic proof tutoring using historical student data. In B. P. Woolf, E. Aïmeur, R. Nkambou & S. Lajoie (Eds.), *Proceedings of International Conference on Intelligent Tutoring Systems* (pp. 373-382): Springer.

Cen, H., Koedinger, K. R., & Junker, B. (2006). Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement. In M. Ikeda, K. Ashley & T.-W. Chan (Eds.), *Intelligent Tutoring Systems* (Vol. 4053, pp. 164-175): Springer Berlin Heidelberg.

Chi, M., VanLehn, K., Litman, D., & Jordan, P. (2011a). Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Modeling and User-Adapted Interaction*, *21*(1-2), 137-180.

Chi, M., VanLehn, K., Litman, D., & Jordan, P. (2011b). An Evaluation of pedagogical tutorial tactics for a natural language tutoring system: A reinforcement learning approach. *International Journal of Artificial Intelligence in Education*, *21*(1-2), 83-113.

Clark, R., & Mayer, R. E. (2003). *e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning*. San Francisco, CA: John Wiley & Sons.

Cronbach, L. J., & Snow, R. E. (1977). Aptitudes and instructional methods: A handbook for research on interactions. Irvington.

Fishman, B., Marx, R. W., Blumenfeld, P., Krajcik, J., & Soloway, E. (2004). Creating a Framework for Research on Systemic Technology Innovations. *The Journal of the Learning Sciences*, *13*(1), 43-76. doi: 10.2307/1466932

Gerjets, P., Scheiter, K., & Catrambone, R. (2006). Can learning from molar and modular worked examples be enhanced by providing instructional explanations and prompting self-explanations?. *Learning and Instruction*, *16*(2), 104-121.

Iglesias, A., Martinez, P., Aler, R., & Fernandez, F. (2009). Reinforcement learning of pedagogical policies in adaptive and intelligent educational systems. *Knowledge-Based Systems*, *22*(4), 266-270.

Koedinger, K. R., Corbett, A. T., & Perfetti, C. (2012). The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. *Cognitive Science*, *36*, 757-798. doi: 10.1111/j.1551-6709.2012.01245.x

Martin, K. N., & Arroyo, I. (2004). Agentx: Using reinforcement learning to improve the effectiveness of intelligent tutoring systems. In J. C. Lester, R. M. Vicari & F. Paraguaçu (Eds.), *Proceedings of the International Conference on Intelligent Tutoring Systems* (pp. 564-572).

McLaren, B. M., van Gog, T., Ganoe, C., Yaron, D., & Karabinos, M. (2014). Exploring the assistance dilemma: Comparing instructional support in examples and problems. In *International Conference on Intelligent Tutoring Systems* (pp. 354-361). Springer, Cham.

Mostafavi, B., & Barnes, T. (2017). Evolution of an intelligent deductive logic tutor using data-driven elements. *International Journal of Artificial Intelligence in Education*, *27*(1), 5-36.

Najar, A. S., Mitrovic, A., & McLaren, B. M. (2014, July). Adaptive Support versus Alternating Worked Examples and Tutored Problems: Which Leads to Better Learning?. In *International Conference on User Modeling, Adaptation, and Personalization* (pp. 171-182). Springer, Cham.

Renkl, A. (2002). Worked-out examples: Instructional explanations support learning by self-explanations. *Learning and instruction*, *12*(5), 529-556.

Salden, R. J., Aleven, V., Schwonke, R., & Renkl, A. (2010). The expertise reversal effect and worked examples in tutored problem solving. *Instructional Science*, *38*(3), 289-307.

Shen, S., & Chi, M. (2016a). Aim Low: Correlation-based Feature Selection for Model-based Reinforcement Learning. In EDM (pp. 507-512).

Shen, S., & Chi, M. (2016b). Reinforcement Learning: the Sooner the Better, or the Later the Better? *Proceedings of the International Conference on User Modeling and Adaptive Personalization* (pp. 13-17).

Slavich, G., & Zimbardo, P. (2012). Transformational Teaching: Theoretical Underpinnings, Basic Principles, and Core Methods. *Educational Psychology Review*, *24*(4), 569-608. doi: 10.1007/s10648-012-9199-6

Taylor, R. S., O'Reilly, T., Sinclair, G. P., & McNamara, D. S. (2006). Enhancing learning of expository science texts in a remedial reading classroom via iSTART. In *Proceedings of the 7th International Conference on Learning Sciences* (pp. 765-770). International Society of the Learning Sciences.

Tetreault, J. R., Bohus, D., & Litman, D. J. (2007). Estimating the reliability of mdp policies: a confidence interval approach *Proc. of HLT-NAACL* (pp. 276-283).

Tetreault, J. R., & Litman, D. J. (2008). A reinforcement learning approach to evaluating state representations in spoken dialogue systems. *Speech Communication*, *50*(8-9), 683-696.

Williams, J. D. (2008). The best of both worlds: Unifying conventional dialog systems and POMDPs. In Ninth Annual Conference of the International Speech Communication Association (pp. 1173-1176).

SECTION III – CONTENT AUTHORING

Dr. Keith Brawner, Ed.

Chapter 10 – Introduction to the Content Authoring Section

Keith Brawner
U.S. Army Combat Capabilities Development Command - Soldier Center – Simulation and Training Technology Center

Core Ideas

Intelligent tutoring systems (ITSs) present the idea that content can be customized to the level of the individual student. As a consequence of this individual personalization, the individual learns from content which was created or selected for them, and has increased learning gains - either decreased training time, improved performance, longer retention, or another result. A natural byproduct of this process is that **more** content must be available for the learner to learn from. While learning can be sped through the **omission** of wasteful material for high-performing students, lower-performing students presumably need something other than the material that they have already experienced.

This need for additional content is a significant cost behind the creation of an adaptive learning system. If it is desirable for two students to experience two different versions of content teaching the same learning objective, there must **be** two versions of this content to give. A recent industry survey revealed that the cost of developing an hour of passive e-learning instruction was 42 hours (Defelice, 2017). Doubling, and perhaps redoubling, this cost for individualization is somewhat intolerable.

As such, there is a need for ITSs to use the various mechanisms of a self-improving system to improve the content which they deliver. Within this section are papers which present differing manners of improving the system content over time. Each of these systems described in this section show how artificial intelligence and machine learning technologies can be applied to education. They also show the need for the separation of component parts of an ITS - each of these components makes it so that it can be algorithmically improved through application of the technology. The reader should reflect on this as they read the upcoming section - how can content creation be automated, how can these automations be spread across system components, and how can they be spread across systems?

Individual Chapters

The first of these papers by **Botelho and Heffernan** describes a manner to crowdsource instructor-to-student feedback among instructors, so as to automatically present short feedback messages and save manual feedback creation on the part of the instructor. They argue for the incorporation of data and feedback from the system users – specifically teachers. The creation of the quick messages relies on both the people who create such messages and upon the content available for the extraction of the reinforcement learning processes.

The second of these papers by **Cai, Graesser, Hu, and Cockroft** discusses the various ways in which the AutoTutor system uses content from its users to refine; including answer clarification, question-answer pairing, and question generation, with plans to expand these operations to its speech act classifier. They perform these actions using the wide variety of data available to the system – data from students, teachers, authors, and the system interactions – in order to rebuild components. They argue for the standardization of components for replicability and drop-in improvement.

The final of these papers, **Folsom-Kovarik, Rowe, Brawner, and Lester** discusses how content can be algorithmically created and delivered to both students, for use, and instructors, for selection and tuning for students. Content creation is automated through a relatively complicated procedure of evolutionary algorithms, reinforcement learning, simulated students, and exploration metrics. They argue the need for system-level information about learning objectives, starting content, and student data in order to either enable instructors to select content or to allow automatic system assignment.

Reference

Defelice, R. (2017). "How Long to Develop One Hour of Training? Updated for 2017." from https://www.td.org/insights/how-long-does-it-take-to-develop-one-hour-of-training-updated-for-2017.

Chapter 11 – Crowdsourcing Feedback to Support Teachers and Students

Anthony F. Botelho and Neil T. Heffernan
Worcester Polytechnic Institute

Introduction

There are 3.1 million public school teachers in the United States (NCES, 2018), and although some technologies have been proposed to replace the role of the teacher, few technologies have been developed to augment teachers' abilities. One of a teacher's most time-consuming tasks is to provide meaningful feedback to student classwork and homework. This feedback is particularly critical for open-ended problems that require students to explain their work. The National Council of Teachers of Mathematics (NCTM, 2018) and Common Core State Standards (CCSS-M, 2018) are making big pushes for students to become better communicators in the realm of mathematics. Cognitive science research supports the premise that students learn math concepts much better if they explain answers to cognitively demanding "deep" questions in their own words (Pashler et al., 2007). This report cited that simply asking students to explain their answers has learning benefits (King, 1992; Pressley, Tannebaum et al., 1990; McNamara, 2004) even if the students do not receive feedback on their answers, and it has been reported that these results are even stronger when feedback is provided (Shute, 2008). In a study by Craig, Sullins, Witherspoon, and Gholson (2006), students used computers to answer deep questions, and then received feedback from the program using Natural Language Processing (NLP), leading to large gains in learning. Despite the evidence that feedback plays an important role in the learning process, few technologies are effectively supporting teachers in providing feedback to students.

On an average school day, teachers do not have the time to write detailed feedback on every student response on a homework assignment. Self-report surveys and time-use diary methods have found that teachers spend about two-to-three hours a day outside of school (Department of Education, 2013; DePaepe, 2015) creating lesson plans and grading. One study found the planning time to be 8.4 hours per week, leaving about four hours per week for grading and commenting on student work. For a 7th-grade math teacher with approximately 100 students, a teacher would have 30 seconds each day per student to provide feedback on submitted work. This alarming statistic suggests that teachers are in need of better support in order to provide students with beneficial feedback.

Feedback comes in many forms and likely varies in effectiveness depending on the timing, deployment method, and content. In mathematics, computers have become very good at providing correctness feedback on a large number of problems. As most mathematics problems are "closed-form" where there is likely a single (or small number) of possible answers, computers can easily tell a student if an answer is correct. A benefit of computer grading is that data collected can be used to inform teachers of common incorrect answers and student mistakes. With open-ended problems that ask students to "explain," computers are still relatively poor at assessing student work and few, if any, learning systems provide any meaningful feedback for open responses. In fact, many of the widely used Intelligent Tutoring Systems, such as McGraw Hill's ALEKS™ and Carnegie Learning's Cognitive Tutor™, have no concept of open-response questions, likely for this reason. Other systems, such as ASSISTments™, allow teachers to assign both open-ended and closed-form problems, but requires teachers to manually grade the open-responses.

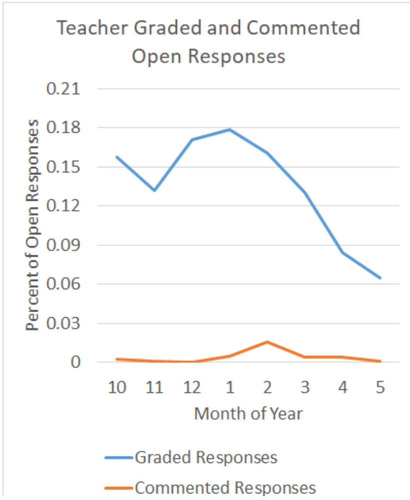


Figure 1. Of the 300,000 times a teacher has graded, we see that the amount of grading decreases by the end of the year.

This philosophy maintains the teacher's role in reviewing and providing feedback to students, even if it is time-costly, but also allows teachers to provide feedback beyond simply indicating if the response is correct; open-ended problems can provide teachers with greater insight into students' knowledge of material. With this, teachers are able to highlight the strengths of student work and offer instruction to those who may need it. In addition to giving teachers information about student knowledge, open-responses also provide an affect component (e.g., "My teacher pays attention to me and my learning") and accountability component (e.g., "My teacher is watching and sees how much effort I put in"). Rimm-Kaufman et al. have shown how important the one-on-one student-teacher relationship is at driving emotional engagement; if students think their teacher is paying attention, they are more likely to engage and put in effort, resulting in more learning (Rimm-Kaufman, Baroody, Larsen, Curby, & Abry, 2015). However, the difficulty of manually entering grades and detailed feedback on student work is evident through data collected from ASSISTments teachers shown in Figure 1. Over the last few years, ASSISTments teachers have graded over 300,000 (or 12% of all problems) open-ended responses and have written more detailed feedback for 60,000 (less than 1% of all problems) of those. Although this percentage of problems for which feedback is provided increases in the middle of the school year, there is a decline in the amount of feedback given to student open response problems over the course of the school year. These low percentages reflect the fact that teachers lack the time to grade and provide detailed comments to students' responses. The question therefore becomes how to augment the teacher's ability to provide this feedback to students and reduce the time cost to do so.

Crowd Sourcing Holds the Key

Crowdsourcing is a vital component in the development of adaptive learning systems. This component, explored in Heffernan et al. (2016), allows teachers to share their own created content (e.g. problems, instructional videos, feedback, explanations, and more) with other teachers to benefit student learning. In that article and here, crowdsourcing describes the task of obtaining and sharing contributions from multiple users (i.e. teachers and occasionally students), as opposed to acquiring content and curricula from only a single or small team of experts (Porcello & Hsi, 2013). Emulating the user-created content model of websites like Wikipedia, the sharing of user-created content, especially when vetted for effectiveness, provides a means of scaling the available content to match the needs of a growing population of users. As Heffernan et al. (2016) discusses, there is a large amount of evidence that supports the use of crowdsourcing in many domains, including human-computer interaction (Doan et al, 2011; Howe 2006; Kittur, et al., 2013). Crowdsourcing has gained attention in publication venues such as HCOMP (Conference on Human Computation and Crowdsourcing), CSCW (Computer Supported Cooperative Work and Social Computing), CHI (Computer-Human Interaction), and Collective Intelligence (see also Malone & Bernstein, 2015). It is only recently, however, that researchers and developers of adaptive learning systems have begun to take advantage of teacher- and learner-sourced materials.

ASSISTments is among the few known existing learning platforms to be actively developing infrastructure to support the creation and sharing of both teacher-created materials and expert-sourced content. A recently released tool called TeacherASSIST, for example, allows teachers to write feedback in the form of hints and explanations for any problem that they wish to assign to their students. Problems can be sourced from an existing open educational resource (OER) such as EngageNY or Illustrative Mathematics, from other available content, or even a problem written by another teacher, and the tool allows teachers to provide their own feedback to benefit their students. Similarly, teachers who are found to create effective feedback are able to share their content with all teachers in ASSISTments assigning the same content.

Methods

We suggest that crowdsourcing teacher feedback can address the aforementioned problem of time required to give students adequate feedback on their classwork and homework. Figure 1 illustrates that there are a small percentage of teachers providing feedback in the form of a grade and comments for open-ended problems. These teachers often repeat much of the feedback that they give to several students who answer with a similar type of response; two students who make the same mistake, for example, even if answered through different phrasing, may benefit from the same feedback. Similarly, it is likely that students in one class who find a piece of content difficult may exhibit the same misconception as students in a different class. The utility of reporting and studying common wrong answers has been recognized in previous work (Feng & Heffernan, 2006; Wang et al., 2015; Ostrow et al., 2015). Common wrong answers, as well as different correct answers, are likely found throughout student open responses and may be leveraged when developing teacher feedback tools. As many open-ended questions as are available to teachers, there are larger magnitudes of possible student responses. Considering the amount of variation and growing scale of open-ended mathematics problems to support (i.e. as new content is added or new open resources become available), crowdsourcing feedback from teachers can allow systems to grow. As teachers provide feedback to a variety of student responses every day, even considering the small percentage of teachers who do, such feedback could be shared with others and used to benefit future students.

Within ASSISTments, recent NSF support (NSF #1822830) has funded a project to develop a tool called DRIVER-SEAT (the *Dialogue Reinforcement Infrastructure for Volitional Exploratory Research – Soliciting Effective Actions from Teachers*), to augment the teacher's ability to provide meaningful feedback by sharing teacher-written feedback. While still in its early stages of development, DRIVER-SEAT aims to combine teacher-sourced content with machine learning to enhance teacher-student interactions. Inspired by Google's Smart Reply tool (Kannan et al., 2016) which helps users respond to email by suggesting a set of automatically-generated responses, DRIVER-SEAT attends to student activity and performance – including, but not limited to responses to open-ended problems – by suggesting feedback messages to send to students. This tool currently focuses on aiding teachers in assessing student responses to open-ended problems. In the future, it is planned to move beyond this task to crowdsource feedback pertaining to student behavior, utilizing many of the sensor-free detectors of affect and disengaged behavior that have been previously developed in ASSISTments (Walonoski & Heffernan, 2006; Paquette et al., 2015; Botelho et al., 2017).

The task of automatically assessing student open-ended work is by no means trivial and has been the focus of previous works for problems in non-mathematics domains (Ramachandran et al., 2015; Zhao et al., 2017; Riordan et al., 2017). Although far from the directed instructional feedback messages envisioned for the DRIVER-SEAT tool, understanding how teachers would assess student responses is a first step toward selecting appropriate feedback to deliver to students. When providing feedback, it would be helpful to have an estimate of how correct the student's response is. To accomplish this, NLP techniques such as bag-of-words, term frequency-inverse document frequency (tf-idf), and GloVe (Pennington et al., 2014), have been combined with machine learning models including decision trees and long short term memory (LSTM) recurrent deep learning networks to 1) understand the difficulty of predicting the teacher-assigned grade for student responses to open-ended problems, and 2) explore how the growing scale of teacher-sourced assessment labels is likely to impact the precision of our models. This approach, detailed further in (Erickson et al., under review) and described in this chapter, represents the first step toward helping teachers improve their ability to provide feedback and effectively communicate with their students through the use of teacher-sourced content.

The method itself involves an ensembling of two representations of student answers. First, the bag-of-words approach with tf-idf calculates a weighted-categorical representation of all words in the corpus. For all words (including numbers and equations) written by a student, a tf-idf weight, representing the estimated importance of each word as calculated by its frequency of use within and across all student responses, is used as input to a decision tree observing student correctness; words such as "is," "the", and "as" are given a lower weight as they likely appear frequently in many student responses, leaving other words with higher estimated importance for the decision tree to then associate with a given grade (i.e. if the student uses the fraction "3/4" but not the word "ratio" in the response, that student may be given a lower grade if the teacher expected to students to use such terminology).

This decision tree with tf-idf approach, considered simple in comparison to more recent developments in NLP techniques, works well with smaller sets of data since every word is able to be represented. This is not the case when using pre-trained models such as GloVe to represent words, but such methods are trained to capture the contextual and semantic meaning of words that is lost when using bag-of-words and tf-idf. Similarly, the use of LSTM models over decision trees may be beneficial as these models capture sequential relationships (such as the ordering of words) that is not observable when using the more simple modeling approaches. For this second method of representing words, GloVe is applied, resulting in a 100-dimensional numeric embedding to represent each word; words that are not recognized by the pre-trained GloVe model are given a default vector of zeros. While it is certainly possible to train a GloVe model, doing so requires a large set of data that is not often available (the pre-trained version used in the described work was trained on data from Wikipedia, the scale of which is far beyond what is available from student open responses. These GloVe embeddings are then used as input into a LSTM model that uses the sequence of words to predict what the final grade of the answer. As with most deep learning methods, a large set of data is needed to successfully train such models and, without such data, the models are prone to overfitting.

Results

While the work of Erickson et al. (under review) explores only a small number of what we consider to be simple models in comparison to what has previously been applied to similar tasks (Zhao et al., 2017), the results highlight several important characteristics to consider during further development of the project. As the observed teacher-provided grades follow a 5 point (0-4) grade, we treat the problem as a classification task, using metrics of AUC (Hand & Till, 2001) and multi-class Cohen's Kappa, but also want to know how close our predictions are to each grade, including a measure of root mean squared error (RMSE).

Table 1. The performance of our initial models reported in Erickson et al. (under review).

	Avg AUC	Avg Kappa	Avg RMSE
Majority Class	0.500	0.000	0.385
Decision Tree Model	0.581	0.257	0.219
LSTM w/GloVe	0.140	-0.006	0.212
Ensemble	0.594	0.326	0.182

While it is apparent from the results reported in Table 1 that the deep learning model does not perform well likely due to model overfitting (a significant concern when applying such a model to a small dataset), the combination of the deep learning model and simpler decision tree led to improvements in model performance in measures of all three metrics. In this task, the better performance in these metrics suggest that the ensembled model is able to better identify the specific grade given (through the values of AUC and Kappa) and does so with greater confidence (through the value of RMSE) than any of the models individually.

Discussion

The DRIVER-SEAT project is an excellent example of how the collection of data through a platform like ASSISTments can improve our understanding of the benefit of data and can provide insights about how different teachers assess students. For example, a second analysis reported in Erickson et al. (under review), used a bootstrapping method of sampling responses with increasing sample sizes to illustrate the decision tree model performance in regard to RMSE as more data is made available to the model. Figure 2 illustrates an example of how collecting even a small number of teacher-assessed responses can lead to more precise models that can then be used to benefit other teachers.

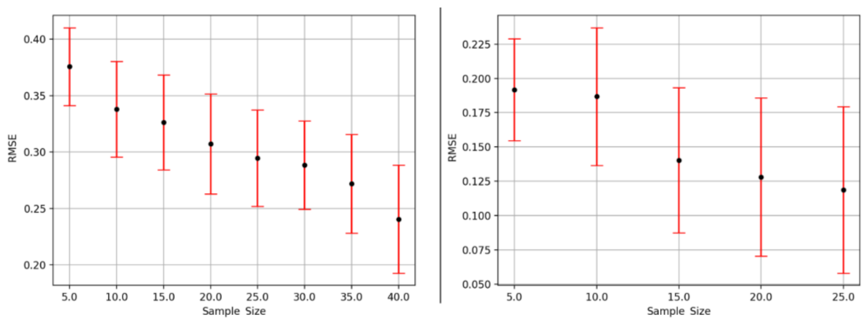


Figure 2. The performance of our initial models with increasing sample sizes as reported in Erickson et al. (under review).

Perhaps one of the most beneficial aspects of utilizing crowd-sourced content to share with and develop tools for teachers is the amount of variation that can exist among that content. In our example of building assessment tools as a part of the DRIVER-SEAT project, a larger variation among teacher perspectives and approaches can benefit the models to support a wider range of teachers. As the project continues in the direction of suggesting more directed feedback for teachers to give to students, larger variations in the tone, focus, and length of feedback messages can only benefit such a tool.

Recommendations and Future Research

The development of educational platforms and tools should consider teacher-sourcing to supplement expert-authored content to scale and fit teachers' needs. Systems that ignore data, feedback, and content created by teachers miss a wealth of information. Of course, not all content generated by teachers is of the highest quality, nor does all such content benefit students equally, so it is important to consider how to identify what works for which sets of students. Conducting randomized controlled trials (RCTs) may help identify beneficial content, but other scalable alternatives such as reinforcement learning (e.g. multi-armed bandit algorithms as is explored in Williams et al., 2016) may also help to identify what works.

Taking advantage of the variation of teacher-created content can also further be leveraged to benefit adaptation and personalization in learning systems. Such variation is needed in developing means of identifying not only what works, but also in identifying for whom (i.e. which groups of students) one set of feedback, instruction, or content may most benefit.

The use of the two methods described here (the decision tree with tf-idf and the LSTM with GloVe embeddings) represent two drastically different complexities of approaches. While ensembling did lead to improvements in performance, much of this performance is likely attributable to the simpler decision tree model. This highlights an important consideration for the development of future systems that aim to implement smarter data-driven tools. There is a need to better understand how the scale of data affects the performance of applied models. It is likely that the more complex methods will eventually outperform the simpler methods, but it is not clear where that point exists. It would therefore be beneficial to develop the means of transitioning between simpler and more complex methods as the scale of data increases to avoid overfitting when data is scarce, and make sufficient use of the data when it becomes available.

Conclusions

While the Results in this chapter detail empirical research being conducted through the development of the DRIVER-SEAT project, such development is only possible through teacher crowdsourced content and feedback. The future designers of educational platform must consider the wealth of content and data currently being generated by teachers in real classrooms who not only often are willing to share, but also in many cases are eager to share and improve their own instructional practices by observing other teachers' created content. In this way, the crowd-sourcing of content and the sharing of such materials either directly or through tools such as those under development for the DRIVER-SEAT project, is positioned to benefit researchers, teachers, and students alike.

References

Botelho, A. F., Baker, R. S., & Heffernan, N. T. (2017, June). Improving sensor-free affect detection using deep learning. In *International Conference on Artificial Intelligence in Education* (pp. 40-51). Springer, Cham.

CCSS-M, (2018). *Common Core State Standards Initiative*. Retrieved from http://www.corestandards.org/

Craig, S.D., Sullins, J., Witherspoon, A., and Gholson, B. (2006). The deep-level-reasoning-question effect: The role of dialogue and deep-level-reasoning questions during vicarious learning. *Cognition and Instruction*, *24*, 565-591.

DePaepe, J. (2015). *CWU Teacher Time Study: How Washington Public School Teachers Spend Their Work Days*. Retrieved from https://www.cwu.edu/sites/default/files/images/teachertimestudy.pdf

Department of Education (2013). The 2013 Teachers' Workload Diary Survey. *The United Kingdom's Department of Education*. Retrieved from https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/ 285941/DFE-RR316.pdf

Doan, A., Ramakrishnan, R., & Halevy, A. Y. (2011). Crowdsourcing systems on the world-wide web. *Communications of the ACM*, *54*(4), 86-96.

Erickson, J., Botelho, A. F., Shirai, S., Varatharaj, A., & Heffernan, N. T. (2019). *Exploring the Automated Grading of Student Open Responses in Mathematics.* Manuscript submitted for publication.

Feng, M., & Heffernan, N. T. (2006). Informing teachers live about student learning: Reporting in the assistment system. *Technology Instruction Cognition and Learning*, *3*(1/2), 63.

Hand, D. J., & Till, R. J. (2001). A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine learning*, *45*(2), 171-186.

Heffernan, N.T., Ostrow, K.S., Kelly, K., Selent, D., Van Inwegen, E.G., Xiong, X., & Williams, J.J. (2016). The Future of Adaptive Learning: Does the Crowd Hold the Key? *International Journal of Artificial Intelligence in Education*. Springer New York. DOI: 10.1007/s40593-016-0094-z

Howe, J. (2006). The rise of crowdsourcing. *Wired magazine*, 14(6), 1-4. Retrieved from http://www.wired.com/2006/06/crowds/

Kannan, A., Kurach, K., Ravi, S., Kaufmann, T., Tomkins, A., Miklos, B., ... & Ramavajjala, V. (2016, August). Smart reply: Automated response suggestion for email. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 955-964). ACM.

King, A. (1992). Comparison of self-questioning, summarizing, and notetaking-review as strategies for learning from lectures. *American Educational Research Journal*, *29*, 303-323.

Kittur, A., Nickerson, J. V., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., ... & Horton, J. (2013, February). The future of crowd work. In Proceedings of the 2013 Conference on Computer Supported Cooperative Work. pp. 1301-1318. ACM.

Malone, T. W., & Bernstein, M. (2015). Collective Intelligence Handbook. MIT Center for Collective Intelligence.

McNamara, D.S. (2004). SERT: Self-explanation reading training. *Discourse Processes*, 38, 1-30

NCES (2018) CCD Quick Facts. *National Center for Educational Statistics*. Retrieved from https://nces.ed.gov/ccd/quickfacts.asp

NCTM (2018). Principles and Standards. *National Council of Teachers of Mathematics*. Retrieved from http://www.nctm.org/standards/

Ostrow, K., & Donnelly, C., & Heffernan, N. (2015). Optimizing Partial Credit Algorithms to Predict Student Performance. In *Proceedings of the 8th International Conference on Educational Data Mining*. Madrid, Spain. June 26-29. pp. 404-407.

Paquette, L., Baker, R. S., de Carvalho, A., & Ocumpaugh, J. (2015, June). Cross-system transfer of machine learned and knowledge engineered models of gaming the system. In *International Conference on User Modeling, Adaptation, and Personalization* (pp. 183-194). Springer, Cham.

Pashler, H., Bain, P., Bottge, B., Graesser, A., Koedinger, K., McDaniel, M., & Metcalfe, J. (2007) Organizing Instruction and Study to Improve Student Learning (NCER 2007-2004). *IES Practice Guide*. Retrieved from https://files.eric.ed.gov/fulltext/ED498555.pdf

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532-1543.

Porcello, D., & Hsi, S. (2013). Crowdsourcing and curating online education resources. *Science*, 341(6143), 240-241.

Pressley, M., Tannebaum, R., McDaniel, M. A., & Wood, E. (1990). What happens when university students try to answer prequestions that accompany textbook materials? *Contemporary Educational Psychology*, 15, 27-35.

Ramachandran, L., Cheng, J., & Foltz, P. (2015). Identifying patterns for short answer scoring using graph-based lexico-semantic text matching. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 97-106).

Rimm-Kaufman, S. E., Baroody, A., Larsen, R., Curby, T. W., & Abry, T. (2015). To what extent do teacher-student interaction quality and student gender contribute to fifth graders' engagement in mathematics learning? *Journal of Educational Psychology*, 107(1), 170-185. doi: http://dx.doi.org/10.1037/a0037252

Riordan, B., Horbach, A., Cahill, A., Zesch, T., & Lee, C. M. (2017). Investigating neural architectures for short answer scoring. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 159-168).

Shute, V. J. (2008) Focus on Formative Feedback. *Review of Educational Research.* Vol. 78, No. 1, pp 153-189

Walonoski, J. A., & Heffernan, N. T. (2006, June). Detection and analysis of off-task gaming behavior in intelligent tutoring systems. In *International Conference on Intelligent Tutoring Systems* (pp. 382-391). Springer, Berlin, Heidelberg.

Wang, Y., Heffernan, N. T., & Heffernan, C. (2015, March). Towards better affect detectors: effect of missing skills, class features and common wrong answers. In *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge* (pp. 31-35). ACM.

Williams, J. J., Kim, J., Rafferty, A., Maldonado, S., Gajos, K. Z., Lasecki, W. S., & Heffernan, N. (2016, April). Axis: Generating explanations at scale with learnersourcing and machine learning. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale* (pp. 379-388). ACM.

Zhao, S., Zhang, Y., Xiong, X., Botelho, A., & Heffernan, N. (2017, April). A memory-augmented neural model for automated grading. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale* (pp. 189-192). ACM.

CHAPTER 12 – TOWARD AUTOMATED SCENARIO GENERATION WITH GIFT

J. T. Folsom-Kovarik[1], Jonathan Rowe[2], Keith Brawner[3], and James Lester[2]
[1]Soar Technology, Inc., [2]North Carolina State University,
[3]U.S. Army Combat Capabilities Development Command – Soldier Center – Simulation and Training Technology Center

Introduction

Adaptive instructional systems such as the Generalized Intelligent Framework for Training (GIFT) can tailor training to meet the learning needs of individuals and teams. A significant cost driver in the design and development of adaptive instructional systems is the manual creation of training scenarios. Delivering personalized instruction to students requires the creation of a broad range of instructional materials. Without effective automation, the tailoring that adaptive instructional systems implement is limited by the small number of instructional variants that a human author can define, as well as a one-size-fits-all approach to training. Further, additional scenarios are useful for enhancing replay through drill-and-practice of specific skills. Generating training scenarios for adaptive instructional systems includes two key components: (1) creating novel scenario content, and (2) devising models that dynamically tailor scenario content to learners.

This chapter discusses two parallel efforts to enhance GIFT through the design, development, and investigation of *automated scenario generation*. First, we describe a scenario variation tool that creates many variants of training scenarios to offer the instructor (or GIFT) increased choices between different combinations of instructional support or challenge. Second, we describe a data-driven framework for dynamic scenario adaptation that models how simulation-based training scenarios can be tailored at run-time to foster optimal learning outcomes. These are two possible approaches to addressing the authoring bottleneck inherent in adaptive instructional systems.

In the first approach, the scenario variation tool makes use of a novelty search algorithm (Lehman & Stanley, 2008, 2011). Genetic algorithms such as novelty search rely on a population of prospective solutions which are modified with 'mutation' or 'crossover' operations to create new prospective solutions in a repeating cycle. Prospective solutions with maximum fitness survive and reproduce in the population from one cycle to another. Novelty search replaces the typical genetic algorithm fitness evaluation with a novelty evaluation (Gomes, Urbano, & Christensen, 2012). In this replacement, genetic variants do not compete to become better, but to become *different*. Novelty search has already been used with success to evolve content similar to training scenarios, such as game levels (Liapis, Yannakakis, & Togelius, 2015). In the present research, training scenario variants attempt to become different as measured by the support or challenge they offer learners. As a result, novelty search is well suited to expand the space of possible training scenarios that GIFT can choose from when it tailors training (Folsom-Kovarik & Brawner, 2018). The scenario variations that result from novelty search provide varying levels of support or difficulty, such as offering a series of increasingly more complex scenarios, varying scenario events while ensuring that complexity is comparable, and offering scenarios that combine more complexity in one learning objective but less complexity in another learning objective that requires support.

The second approach being investigated, the dynamic scenario adaptation framework, DEEPGEN, utilizes reinforcement learning (RL) to induce models for run-time tailoring of training scenarios to achieve instructor-specified learning objectives (Rowe, Smith, Pokorny, Mott, & Lester, 2018). RL refers to a family of machine learning techniques for solving tasks involving sequential decision-making under uncertainty (Sutton & Barto, 2018). Over the past several years, a range of RL techniques have been investigated for run-time personalization of virtual learning environments for K-12 and undergraduate education, including modular RL (Rowe & Lester, 2015), multi-objective RL (Sawyer, Rowe, & Lester, 2017), constraint-based RL (Shen, Mostafavi, Barnes, & Chi, 2018), inverse RL (Rafferty, Jansen, & Griffiths, 2016), and deep RL (Wang, Rowe, Min, Mott, & Lester, 2018). Building upon this foundation, DEEPGEN utilizes RL to induce models for enacting run-time adaptations to military training scenarios, aiming to produce training experiences that optimize learning outcomes or provide effective assessments of target skills. Rather than novelty being the primary selection mechanism of scenario selection, as above, the RL-based dynamic scenario adaption uses a population of simulated students.

This chapter is organized as follows. In the next section, we describe work investigating novelty search and RL, respectively, to automatically generate training scenarios in two military training domains. We then describe efforts to engage subject-matter experts to obtain feedback on how to deliver automated scenario generation capabilities to instructors and scenario developers. Next, we discuss initial findings from the two projects, and conclude by offering recommendations for GIFT and directions of future work.

Scenario Generation Methods

Two complementary demonstrations of the two approaches in two domains of militarily-relevant instruction were carried out to investigate automated generation of training scenarios. First, novelty search was demonstrated in a small unmanned air system (SUAS) training scenario. Second, reinforcement learning-based scenario adaptation was demonstrated in the domain of artillery call for fire (CFF) training.

Novelty Search to Automate SUAS Training

The first demonstration focused on generating many variants of training scenarios in advance of training. Infantry employment of SUASs can be trained with a scenario structured by information delivery and choice points. Trainees work through the process to plan, prepare, and execute an unmanned air system (UAS) mission by making decisions based on information such as mission briefings, UAS observations, and popup events. Optimal and acceptable decisions continue the scenario to the next choice point, while one or more unacceptable decision can cause remediation and restart. In this setting, novelty search can offer GIFT valuable opportunities to change scenarios after a restart or to challenge different aspects during training based on learner choices or GIFT's internal learner model.

Technical demonstrations of novelty search showed the technique can generate hundreds or thousands of scenario variants, and the variants are measurably different or similar enabling fine-grained matching to instructional needs (Figure 1). GIFT could match generated scenarios to learners' needs via 48 measures, scaled continuously between support and challenge, representing the 48 learning objectives (LOs) covered in the original training system. The variations did not control LOs directly, but controlled the number, size, and location of units and areas anywhere on a scenario map. A single variation in one such element might alter the scenario's support for many LOs. For example, moving an enemy armor unit might challenge a recon LO if the unit moved into a wooded area and simultaneously support a dynamic response LO if the tank moved further away from friendly forces. As a

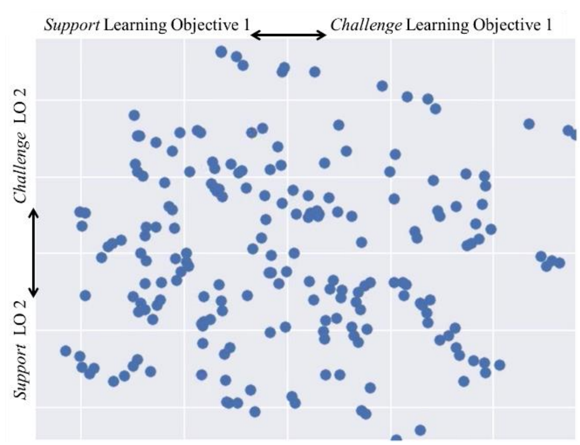


Figure 1. Novelty search creates training variants with many combinations of support and challenge.

result of these interactions, novelty search tended to find scenario variations that supported some LOs and challenged other LOs in combinations that had not previously existed.

The authors explored several methods to present the wealth of variations to nontechnical end users, such as instructors who wish to review the available variants or select one training variant which will best support specific trainees. An initial evaluation by a subject matter expert (SME) resulted in defining a presentation of training variants that military instructors are likely to find valuable. The initial evaluation resulted in user interface recommendations to translate technical variation into a human-usable presentation, and will support an upcoming evaluation by operational users.

Reinforcement Learning-Based Scenario Adaptation in Call for Fire Training

The second demonstration focused on devising RL-based policies for adapting events in example CFF training scenarios. In a CFF mission, an infantry soldier requests indirect fire on a target from supporting artillery (e.g., mortar, field artillery, unmanned aircraft). The soldier, or forward observer, follows a concise communication protocol to identify himself, describe the mission type, describe the target and location, describe the method of engagement, adjust fire as necessary, and conclude with a battle damage assessment. There are a broad range of scenario adaptations that can be enacted to augment the difficulty of a CFF training scenario, such as introduction of obstacles, adjustments to mission type, modifications to enemy behaviors, modifications to weather and time of day, adjustments to type of target and location, and changes to artillery battery response.

Dynamic scenario adaptation involves enacting a series of decisions about how to orchestrate training events at run-time. In DEEPGEN, the full range of possible adaptations is defined in a *Scenario Adaptation Library*, which determines what types of scenario events can be adapted, how they can be adapted, and when they can be adapted. In RL terminology, these correspond to the actions in a Markov decision process, which are enacted at run-time to produce a training experience that meets instructor-specified objectives. The state representation includes both the state of the learner and the history of scenario events to date. Reward is defined in terms of the unfolding scenario's alignment with target instructional objectives. RL provides a systematic process for exploring alternate approaches to dynamic scenario adaptation, gradually improving over time as more trainees interact with the scenario generator.

Figure 2. Virtual Battlespace 3 simulation platform.

To investigate RL-based scenario generation in the domain of CFF training, we utilized example scenarios from Virtual Battlespace 3 (VBS3). Developed by Bohemia Interactive Simulations, VBS3 is a 3D simulation platform that is widely used by the U.S. Army for a range of training purposes, including IED training, surveillance systems, land navigation, route clearance, convoy training, and many other tasks. In this work, we utilize the VBS2Fires plug-in, a third-party tool created by SimCentric Technologies that provides a graphical user interface (GUI) and ballistics simulation engine for training CFF in VBS3 (Figure 2). Automatic scenario generation, which is performed by modifying example VBS3 scenarios provided as input to DEEPGEN, is realized in VBS3 by implementing an automated, or semi-automated, compilation process that produces executable VBS mission files.

As a preliminary investigation of RL-based scenario generation, we implemented a prototype scenario generator that utilizes a multi-armed bandit formalism for inducing policies to generate initial conditions of CFF training scenarios (Rowe et al., 2018). Multi-armed bandits are closely related to RL, but they do not account for the stochastic effects of actions on the state of the task environment, making them a natural starting point for technical demonstration purposes. We utilized a multi-armed bandit approach to induce policies for selecting the weather, time of day, and target movement characteristics in an example CFF training scenario. We considered three possible values of weather (clear, cloudy, rain), 3 possible values for time of day (day, dusk, night), and two possible values of target movement (stationary, moving). To train the multi-armed bandit policies, synthetic data generated from a simple probabilistic simulated learner model was utilized. We ran 50,000 trials of an 18-armed bandit using the UCB1 algorithm to manage exploitation/exploration of different scenario adaptations. Results showed that the scenario generator converged on a stable ranking of alternative training scenarios over time, recommending "easier" scenarios for low competency simulated learners and "harder" scenarios for high competency simulated learners. Although the analysis did not involve modeling sequential decisions about scenario adaptations, it did demonstrate the potential for solving automated scenario generation tasks using RL techniques (Rowe et al., 2018).

Initial Findings

Presentation of Many Variants for Instructor Usability

The first demonstration resulted in several evolutions of presentation for training content like varying scenarios. The underlying novelty search algorithm can vary training in up to eight dimensions per learning objective (not just one, support versus challenge), as described in Dunne, Sivo, and Jones (2015). The dimensions are hypothesized to be domain-independent, so an early idea was to present the dimensions of variation to end users, explaining exactly how each variant differed from the others. Methods that were attempted included arraying many dimensions into visual rows, and selecting two or three dimensions for display in (x,y) space similar to Figure . Dimensions could be selected by their range or variability or combined for display via projection. These early attempts were visually complex and offered details that instructors probably do not need to consider.

A second prototype was created (Figure 3). The key features of this prototype include summarizing all dimensions of variation into just three bins per learning objective (easy, medium, and hard), as well as placing a "top five" scenario list front and center, rather than showing every available variant. The list priority was defined by data captured during usage, and was again designed to be domain-general. Usage data included number of times a variant had been used, average duration, and average pass rate. The parameters were intended to work for multiple instructional domains and forms of instructional delivery, and overall to capture institutional knowledge of which variants were more useful. Each variant also received a random, two-word mnemonic to let instructors remember and search for familiar variations.

One result from engaging with a SME is feedback that will lead to a third iteration of a usable interface. Military instruction in many domains is described by a three-dimensional matrix. The three dimensions are similarly defined in different domains (Sanders & Dargue, 2012): training complexity, mission, and mission conditions for a command staff trainer; weapons platform, target array, and environment for a gunnery trainer; or task complexity, threat level, and environmental factors for an SUAS trainer. The fine-grained dimensions of variation were initially defined to enhance the classic three dimensions, but an important lesson is that the instructors and instructional designers are typically accustomed to working within the three similar dimensions. Therefore, a third prototype should translate the many underlying variations back into just three dimensions, to provide visual shorthand and explanation of how each scenario varies.

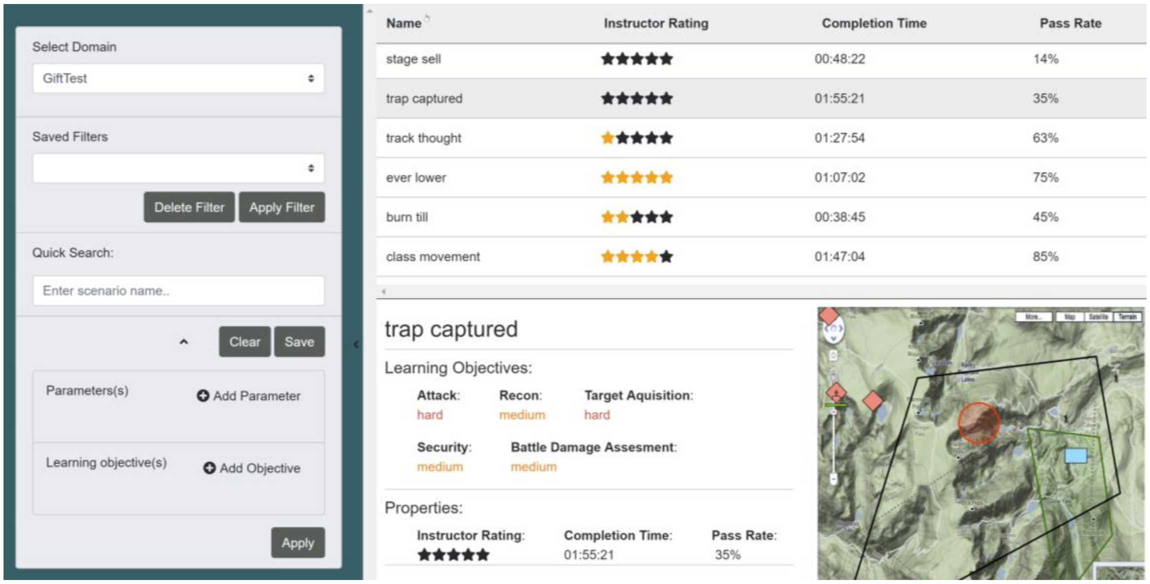


Figure 3: Filtering and ranking variants based on usage parameters and challenge for learning objectives.

Developing Instructor Tools for Dynamic Training Scenario Adaption

Building upon our proof-of-concept demonstration of a multi-armed bandit approach to automated scenario generation, the second demonstration proceeded by investigating two complementary directions: (1) expanding the Scenario Adaptation Library to broaden the space of generatable scenarios while preserving military relevance for real-world CFF training use cases, and (2) designing and developing a prototype DeepGen instructor tool for integrating dynamic scenario adaptation capabilities within adaptive instructional systems, such as GIFT. To ensure project alignment with the requirements of U.S. Army training for CFF, we engaged in iterative cycles of feedback with an Army SME bringing extensive experience in CFF training and adaptive training systems.

First, the Scenario Adaptation Library was expanded to incorporate 13 additional adaptable event sequences beyond the 3 utilized in the multi-armed bandit demonstration. This yielded 16 possible dimensions for dynamic scenario adaptation, each with 2-5 possible levels, corresponding to more than 1,000,000 possible variations that could be generated from a single example training scenario. Several adaptable event sequences could be generalized across multiple example scenarios, such as target type (e.g., wheeled vehicle, tank, bunker) and target behavior (e.g., stationary, on patrol), whereas other adaptable event sequences were tied to particular example scenarios, such as the counter-attack behavior of a specific enemy unit. After developing the expanded Scenario Adaptation Library, we obtained SME feedback on how well the expanded set of adaptable event sequences covered the range of useful CFF training scenarios across difficulty levels and instructional objectives. Further, the SME provided input on scenario elements that lacked realism or required refinement for relevance to Army training purposes. For example, SME input addressed issues such as how terrain and target location can impact scenario difficulty, and common target types of call-for-fire missions.

Next, we began to devise user interface mockups for a DeepGen instructor tool to configure automated scenario generation functionalities in adaptive training systems. The tool was designed for use by military instructors and training content developers, and it was envisioned to support eventual integration with GIFT. Three complementary modes of automated scenario generation were targeted as use cases: (1) offline scenario generation, (2) online scenario generation, and (3) run-time scenario generation. In offline scenario generation, an instructor and/or developer utilizes a tool to produce scenarios *prior to a training exercise*. This has labor-saving benefits, and it also expands the range of possible scenarios that can be created for training. However, offline scenario generation does not support scenario personalization, as it lacks access to an explicit learner model that captures a trainee's state (e.g., knowledge, skills, abilities) or trait information (e.g., prior knowledge, goal orientation). In contrast, online scenario generation produces tailored scenarios *just-in-time during training* by consulting a learner model that reflects the trainee's prior performance and competency levels. Online scenario generation is analogous to the outer loop of an intelligent tutoring system, where pedagogical decisions about problem selection are based upon a student model that is maintained by the system (VanLehn, 2006). The third mode of automated scenario generation, run-time scenario generation, takes this process one step further, enacting scenario adaptation *while the trainee is completing a training exercise*. This is analogous to the inner loop of an intelligent tutoring system, where pedagogical support is delivered to guide the learner through the completion of a problem-solving scenario (VanLehn, 2006). In run-time scenario generation, scenario events are dynamically tailored based upon the learner's performance within the scenario thus far. We distinguish between these three modes because they have distinct implications for the design of instructor-facing tools to support automated scenario-generation use cases, as well as the underlying algorithmic techniques used to implement them.

The purpose of the DEEPGEN instructor tool is to provide instructors and developers with the ability to specify what types of training scenarios they seek to be generated, as well as preview generated scenarios prior to execution in VBS3. The workflow for using the tool is as follows. The first step is to select a training domain.[3] Next, the user (optionally) uploads example VBS3 training missions, expanding the set of base scenarios for RL-based scenario adaptation. Several example VBS3 missions are provided by default. The user can also upload configuration files that specify the current Scenario Adaptation Library and Performance Assessment Logic for the training domain, which are prerequisites for effective RL-based scenario generation.

After completing these configuration steps, the user selects criteria to guide automated scenario generation through a menu-based interface (Figure 4). Initially, two types of scenario generation criteria are offered: Target Skills and Scenario Difficulty. A range of target skills for CFF training are enumerated, including different methods for specifying the coordinates of a target, performing effective adjustments to fire, and providing an accurate battle damage assessment. Difficulty levels include easy, medium, and hard, and these designations are determined based upon input from SMEs. The user can also toggle into advanced mode, which provides more granular control over scenario generation. In advanced mode, the user can provide input on the types of artillery utilized, method of engagement, types of terrain, visibility conditions, and provision of hints and feedback in the CFF training scenario.

In offline scenario generation, the user next presses a "generate" button, having provided a set of input criteria, to obtain a ranked list of automatically generated CFF training scenarios. For each scenario, a card-like view presents summary information about the mission, including usage data, target skills, key scenario properties, and a score metric derived from the expected reward associated with that scenario in RL. The scenarios are ranked according to the score metric, which captures the observed effectiveness of the scenario in meeting the user-specified criteria. These scores are updated over time as learners interact with DEEPGEN, refining the system's model of scenario effectiveness based upon the results of RL. When a user clicks on a scenario card, he/she can view a more detailed summary of the mission, which is presented in a

[3] Currently, the only supported domain for automated scenario generation is CFF training. However, the overall approach to scenario generation that is embodied by DEEPGEN is anticipated to generalize to additional training domains.

standard warning order format, providing information about the situation, mission, execution, task organization, and commander's intent. Generated scenarios can be saved to a library of VBS3 missions for subsequent execution during training.

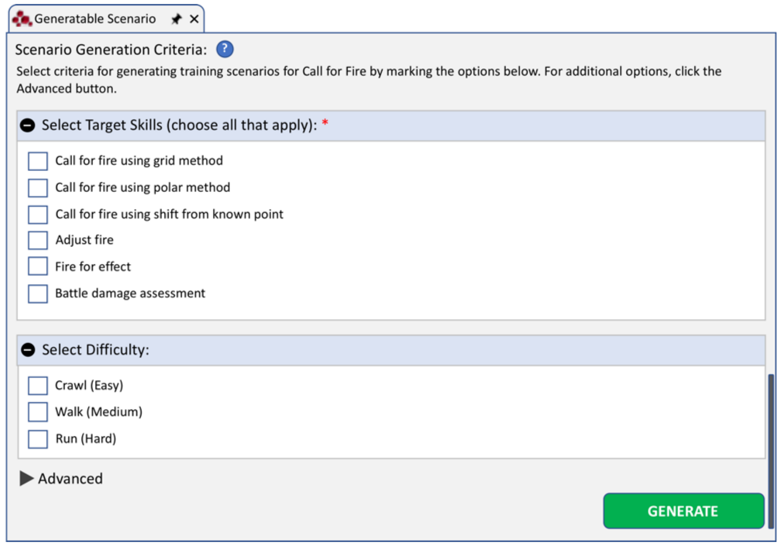


Figure 4. Mockup of DeepGen instructor tool showing default configuration options for automated CFF scenario generation.

The workflow described above is contrasted with anticipated workflows for online scenario generation and run-time scenario generation, respectively. In these latter modes, a "generate" button is unnecessary, because scenarios are generated dynamically during training by tailoring within-scenario events to the individual characteristics of learners. Online scenario generation can be conceptualized as a pedagogical event within a broader instructional sequence, which could also include embedded assessments, direct instruction, and practice with manually-crafted scenarios, for example. Based upon this observation, we have begun to create user interface (UI) mockups of the DeepGen instructor tool that envision automated scenario generation as a course object within the GIFT Course Creator. Devising instructor-facing tools for automated scenario generation that are compatible with existing lesson builders, such as the GIFT Course Creator, will be critical for bringing online scenario-generation use cases into reality.

It should be added that in online and run-time scenario generation, instructors will almost certainly seek the ability to preview how scenario generation systems will operate for different types of learners. Factors such as *transparency* and *explainability* are critical to establishing the trust necessary for human instructors to adopt AI-based technologies, such as automated scenario generation, in their training workflows (Sinha & Swearingen, 2002). Devising methods and tools for visualizing how dynamic scenario adaptation features operate within DeepGen is the subject of continued work.

To guide the iterative design and development of the DeepGen instructor tool, we have engaged SMEs in several rounds of feedback on its user interface design. SME feedback has been instrumental in refining the terminology and criteria featured within the instructor tool, including the CFF skills that are targeted, the types of artillery that are supported, and types of terrain that can be utilized. Specifically, SME feedback led to the addition of configuration options for the method of engagement in CFF (e.g., type of adjustment, trajectory, ammunition, danger close), advised adding terrain options that contain tree cover to enhance scenario difficulty, and suggested refining criteria for time pressure to distinguish between diegetic time pressure (e.g., enemy forces launching an attack) and non-diegetic time pressure (e.g., a time limit).

Discussion

The first demonstration highlights a need in the current state of practice to usefully present many training variations—filtering, finding, and describing the variants presented in a way that aligns with what instructors want to know. Good alignment will allow instructors to start using a sophisticated training system, understand its recommendations, and accept or change them to optimize learning. Poor alignment will increase the barrier to entry for new users and reduce the job effectiveness of any instructors who do not give up on using the training entirely. Because a sufficiently sophisticated computational system may consider hundreds or thousands of factors in recommending an adaptive choice of scenarios or interventions, there is a challenge to identify how these factors, across training domains, should be presented to align with end user needs.

The findings indicate that instructors share a common vocabulary and mental model which are found to appear in multiple sources and training domains. The constructs of task complexity, level of risk or urgency, and environmental conditions should form a template by which more fine-grained variation is understood. Once these three dimensions are accepted as a basis for describing training variants, the dimensions can be applied to different training domains and can be refined by additional measures. If a computational system requires or offers many more dimensions of variation, they may be initially hidden within a summary or rollup metric and available for drill-down when advanced users request more information.

Institutional knowledge, or information about training that is accumulated over time and through repeated use, is likely to provide a secondary or optional entry into the training choices available. In the military use case, achieving the mandated training is paramount and therefore the primary data presentation should support easily identifying mandatory training within a sequence. The three-dimensional model is likely to help filter and find training content for this use case, with qualities such as duration and pass rate available as secondary filters after these.

The findings of the second demonstration illustrate how the intrinsic combinatorics of dynamic scenario adaptation yield a vast range of generatable scenarios for even simple domains, such as CFF training. By integrating additional example scenarios, or devising broader missions that embed CFF within them, the yields of automated scenario generation can be increased further. This observation underscores the importance of devising methods to provide instructors and developers with control over the operation of automated scenario generators. In the DEEPGEN instructor tool, we provide users with general criteria, such as target skills and scenario difficulty, as well as granular criteria, such as domain-specific methods of engagement and adversary behaviors, to guide scenario generation of CFF training missions. Similar to the first demonstration, we find that it is critical to work closely with SMEs to guide the formulation and presentation of control methods to ensure that instructor-facing tools are understandable, usable, and useful. Further, the second demonstration highlights the promise of devising empirically based evaluations of scenario quality that can be leveraged to rank and assess candidate training missions. In RL-based scenario generation, this evaluation mechanism is implemented algorithmically in the form of a reward model that is induced from data on learner interactions with candidate scenarios as well as their performance and training outcomes.

This work also highlights the different ways that automated scenario generation can be integrated into real-world training workflows. Automated scenario generation can be utilized as a labor-saving tool, reducing the costs of developing training scenarios through offline scenario generation processes. Additionally, automated scenario generation can be utilized online, and at run-time, to enhance adaptive training capabilities through dynamic personalization of scenarios in line with the individual characteristics of learners. These complementary modes have significant implications for the design of instructor tools for controlling the operation of scenario generators. In offline scenario generation, an instructor is likely to peruse candidate scenarios, save them to a library of training materials, and deploy them to learners. In online scenario generation, as well as run-time scenario generation, an instructor is likely to seek understanding of how dynamic scenario adaptation will shape learner experiences during a training exercise based upon learners' individual states and traits. Automated scenario generation creates the need for supporting transparency and explainability within instructor tools, which will be critical to establishing the trust necessary for instructors to adopt automated scenario generation technologies in the classroom.

Recommendations and Future Research

Research on automated scenario generation is still in its nascent stages, and there are several promising directions for future research. For the two projects described in this chapter, continued engagement with SMEs, including instructors and training content developers, will be essential for refining the scenario generation tools to support real-world training use cases. In addition, it will be important to investigate how these tools, and their underlying novelty search and RL-based approaches to scenario generation, respectively, generalize to additional training domains. Third, conducting evaluation studies to investigate the prospective labor-saving benefits of automated scenario generation, as well as the training effectiveness of created scenarios, will be critical to develop an evidence base for the benefits of automated scenario generation in adaptive training systems for military domains.

More broadly, there are a myriad of open questions about automatic scenario generation that require further attention. First, it will be important to investigate the relative strengths and weaknesses of alternative computational frameworks that have emerged in recent years, such as generative adversarial networks, for solving automated scenario generation tasks. This calls for methodological progress in the evaluation of automated scenario generation systems, including identification of appropriate instruments, metrics, and research designs that reveal the effectiveness of alternative scenario generation approaches. Second, investigating mixed-initiative systems that enable human instructors and content developers to co-create training scenarios in coordination with automated scenario generation systems has significant potential. Finally, devising examples of how to integrate automated scenario generation functionalities with existing tools for constructing adaptive training systems, such as the GIFT Course Creator, will be critical for taking scenario generation out of the laboratory and into real-world classrooms.

Conclusions and Recommendations for GIFT

The two studies presented in this chapter illustrate recent advances in automated training scenario generation that hold significant promise for real-world training applications. Automated scenario variation before training, and dynamic scenario adaptation during training, are well positioned to help reduce the human effort and cost associated with generating tailored, effective instruction and assessment. Addressing practical considerations in effective deployment and use of such research will help to enrich the training capabilities of adaptive instructional systems such as GIFT, as well as enable the creation of adaptive training systems that continually improve in effectiveness and utility over time. The inputs needed for each of these systems are the instructional objectives, example scenarios that target them, and some amount of student data about experiences with the scenarios. These items are required to generate the scenarios and need to be represented through metadata tags, descriptors, folder structures, or equivalent. For output, the system must either have a link to (1) an instructor interface to select student scenarios, (2) a system interface to automatically assign the scenarios, or (3) both. Adaptive instructional systems must have a way of describing existing content to algorithmic content generators, as well as links to where this content can be placed after its creation.

References

Dunne, R., Sivo, S.A., & Jones, N. (2015). Validating scenario-based training sequencing: The scenario complexity tool. *Proceedings of the the Interservice/Industry Training, Simulation and Education Conference* (No. 15102).

Folsom-Kovarik, J.T., & Brawner, K. (2018). Automating variation in training content for domain-general pedagogical tailoring. *Proceedings of the Sixth Annual GIFT User Symposium* (pp. 75-86). Orlando, FL: U.S. Army Research Laboratory.

Gomes, J., Urbano, P., & Christensen, A. L. (2012). Progressive minimal criteria novelty search. *Proceedings of the Ibero-American Conference on Artificial Intelligence* (pp. 281-290). Springer, Berlin, Heidelberg.

Lehman, J., & Stanley, K.O. (2008). *Exploiting open-endedness to solve problems through the search for novelty. Proceedings of the Eleventh International Conference on Artificial Life* (pp. 329-336). Cambridge, MA: MIT Press.

Lehman, J., & Stanley, K.O. (2011). Novelty search and the problem with objectives. *Genetic Programming Theory and Practice IX* (pp. 37-56). New York, NY: Springer.

Liapis, A., Yannakakis, G.N., & Togelius, J. (2015). Constrained novelty search: A study on game content generation. *Evolutionary Computation, 23*(1), 101-129.

Rafferty, A. N., Jansen, R. A., & Griffiths, T. L. (2016) Using inverse planning for personalized feedback. *Proceedings of the Ninth International Conference on Educational Data Mining* (pp. 472-477).

Rowe, J., & Lester, J. (2015). Improving student problem solving in narrative-centered learning environments: A modular reinforcement learning framework. *Proceedings of the Seventeenth International Conference on Artificial Intelligence in Education* (pp. 419-428). New York, NY: Springer.

Rowe, R., Smith, A., Pokorny, B., Mott, B., & Lester, J. (2018). Toward automated scenario generation with deep reinforcement learning in GIFT. *Proceedings of the Sixth Annual GIFT User Symposium* (pp. 65-74). Orlando, FL: U.S. Army Research Laboratory.

Sanders, J., & Dargue, B. (2012). *A design whose time has finally arrived: Use of adaptive training technology for staff training*. Presented at the 23rd International Training Equipment Conference (ITEC), London, UK.

Sawyer, R., Rowe, J., & Lester, J. (2017). Balancing learning and engagement in game-based learning environments with multi-objective reinforcement learning. *Proceedings of the Eighteenth International Conference on Artificial Intelligence in Education* (pp. 323-334). New York, NY: Springer.

Shen, S., Mostafavi, B., Barnes, T., & Chi, M. (2018). Exploring induced pedagogical strategies through a Markov decision process framework: Lessons learned. *Journal of Educational Data Mining, 10*(3), 27-68.

Sinha, R., & Swearingen, K. (2002). The role of transparency in recommender systems. *CHI'02 Extended Abstracts on Human Factors in Computing Systems* (pp. 830-831). New York, NY: ACM.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction (Second Edition).* MIT Press.

Wang, P., Rowe, J., Min, W., Mott, B., & Lester, J. (2018). High-fidelity simulated players for interactive narrative planning. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, (pp. 3884-3890).

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, *16*(3), 227-265.

Chapter 13 – Self-Improving Components in Conversational Intelligent Tutoring Systems

Zhiqiang Cai[1], Arthur C. Graesser[1], Xiangen Hu[1,2] and Jody L. Cockroft[1]
[1]University of Memphis, [2]China Central Normal University

Introduction

The Generalized Intelligent Framework for Tutoring (GIFT) provides standardized ways to build high performance intelligent tutoring systems (ITSs). The authoring tools in GIFT allow authors to create highly interactive learning content that adapts to learners' individual characteristics (Cai, Graesser, & Hu, 2015). However, developing authoring tools for authors to create systems that can be self-improving is still a challenging problem.

For over four decades, researchers have been trying to develop computer tutoring systems that could be as effective as professional human tutors (Smith & Sherwood, 1976). The effectiveness of tutoring, either human tutoring or computer tutoring, is usually measured by the relative effect size compared to conventional instructional activities without tutoring. The effect size of professional human tutoring was once expected to be 2.0 (Bloom, 1984), whereas untrained human tutoring and computer tutoring is approximately 0.8 (VanLehn, 2011). The tutoring process is a sequence of interactions between a tutor and an individual student. An effective tutor should be able to guide the student to go through a path that is relatively short, if not the shortest, in a learning process. The tutor helps a student in each step of the learning process by recommending learning content, giving immediate feedback and hints, providing and explaining ideal solutions, correcting misconceptions and answering questions (Graesser, Hu, & Sottilare, 2018). Moreover, natural language conversation plays an essential role in human tutoring. The rapid advances in natural language processing and artificial intelligence make it possible to develop incrementally better conversational intelligent tutoring systems. In the last two decades, the Institute for Intelligent Systems at the University of Memphis has developed over a dozen intelligent tutoring systems with conversational agents based on the AutoTutor framework (Graesser, 2016; Graesser et al., 2004; Millis et al., 2011; Nye, Graesser, & Hu, 2014; VanLehn et al., 2007).

AutoTutor uses conversational agents (usually one or two agents) to help students learn. Learning materials in AutoTutor are presented to students as texts, images, videos or content with interactive elements such as buttons, pull-down menus, drag-and-drop objects, etc. AutoTutor responds to students' natural language inputs, as well as events triggered by the interactive elements. An AutoTutor conversation usually starts with a problem. AutoTutor agents help a student to construct a solution to the problem by giving feedback, asking hint questions, correcting misconceptions, answering questions and providing ideal solutions (Graesser, 2016; Nye, Graesser, & Hu, 2014). However, developing conversational ITSs like AutoTutor is expensive. Moreover, the performance of the system relies on iterative improvement based on the data collected from interactions with real students. The improvement process makes the development of an AutoTutor system even more expensive and cumbersome. Therefore, a self-improving mechanism is expected to reduce the development costs and increase the systems' performance.

A self-improving system is a system that can improve its behavior by itself as it evolves while interacting with the world it was designed for. Such a system should have components that are changeable and can be automatically changed without a designer's intervention in the course of acting in the world (Omohundro, 2007). A self-improving intelligent tutoring system should be able to automatically improve using data collected from the interactions with real users. The users include students, teachers and any domain experts. Students provide learning interaction data and learning experience feedback. Teachers provide teaching experience feedback. Domain experts provide human judgments on system performance. In this chapter, we use AutoTutor core components to illustrate possible ways to make intelligent tutoring systems self-improving.

AutoTutor Core Components and Self-improving Possibilities

AutoTutor conversation, often referred to as "Expectation-Misconception Tailored" tutoring, has its special conversation style. It starts with a main question and continues with hint/prompt questions to help students construct a complete answer to a main question. To support such conversations, AutoTutor uses four major components, including question generator (QG), speech act classifier (SAC), answer classifier (AC) and question answerer (QA). In each conversation turn, the QG component creates/selects a question (main/hint/prompt) and waits for a student's answer. The SAC component identifies the type of the input (answer, question, or others). The AC component then identifies the category of the student's answer and gives feedback. The QA component answers a student question if the input is a question.

Question Generation

There are three types of AutoTutor questions: main question, hint question and prompt question. A main question in AutoTutor is a question with a relatively long answer (5 – 10 sentences). The length requirement is for the sake of constructing meaningful conversation that leads to deep learning. The following example shows the main question and the ideal answer of the "Pumpkin Problem" in Newtonian Physics AutoTutor:

> **Main question:** *Suppose a runner is running in a straight line at constant speed, and the runner throws a pumpkin straight up. Where will the pumpkin land? Explain why.*
> **Ideal Answer:** *The person and the object are moving with constant horizontal velocity. When the person throws the object upward only vertical forces are acting on the object. Because only vertical forces are acting on the object, there is no horizontal acceleration. The initial horizontal velocity of the object, which is the same as the person, will not change. The object will travel up and down vertically and move at the same constant horizontal velocity as the person, and as a result the object will land back in the person's hand.*

The ideal answer of a main question is decomposed into multiple statements, called "expectations." The basic rule of such decomposition is that each expectation is an important part of the ideal answer and every part of the ideal answer must be included in an expectation. The expectations in AutoTutor are created for AutoTutor to guide students to construct a complete answer part by part. A hint/prompt question is created to help students construct a specific expectation. The difference between a hint question and prompt question is in the lengths of their answers. The answer of a hint question can be a sentence or a clause, whereas the answer of a prompt question is a single word or phrase. Thus, main question, hint and prompt are questions targeting answers of three different levels: paragraph level, sentence/clause level and word/phrase level. The following example shows an expectation, one of its hints and one of its prompts:

> **Expectation:** *After release, only gravity acts on the pumpkin, and it has no effect on the horizontal velocity.*
> **Hint:** *What can you say about the forces acting on the pumpkin, after release?*
> **Hint answer:** *Gravity is the only force acting on the pumpkin.*
> **Prompt:** *Which component of the pumpkin's velocity does not change after it is released?*
> **Prompt answer:** *the horizontal component.*

AutoTutor questions and answers are authored partly by domain experts and partly by the AutoTutor QG component. Cai et al. (2006) proposed a template-based hint/prompt question generation method that was used in facilitating AutoTutor script authoring. A number of question classification schema have been proposed with specific question generation templates for each question category (Cai et al., 2006; Graesser, Rus, & Cai, 2008; Graesser, Rus, Cai, & Hu, 2012; Olney, Graesser, Person, Piwek, & Boyer, 2012; Rus & Graesser, 2009). Olney, Graesser, Person, Piwek, and Boyer (2012) proposed a technique for generating questions from a concept map. Neural network models (Duan, Tang, Chen, & Zhou, 2017) have also come into the field of question generation. With the help of question generation techniques, the authoring load for authors is greatly reduced. Instead of manually creating questions, a domain expert could create a main question and an ideal answer and let the QG component automatically split the ideal answer into expectations and generate candidate hints and prompts.

AutoTutor questions need iterative improvement. It is difficult for a domain expert and AutoTutor QG component to create an ideal set of questions without the support of student data. There could be missing hint/prompt questions or not very useful questions; there could be questions that are either too easy or too hard. There could be questions of which answers are uncertain or unclear (Cai, Gong, Qiu, Hu, & Graesser, 2016). Student data can help solve such problems. For example, how frequent a question is used in conversations is an indicator of the relevance of the question; the percentage of correct answers from students is an indicator of the difficulty level of the question; the answer uncertainty could be an indicator of question clarity; and the percentage of conversations that leads to successful final solutions could be an indicator of question coverage (whether or not there should be more questions).

In addition to student data, authors may provide data by making judgements on the automatically computed quality indicators, adding new questions, revising bad questions and removing useless questions. Such data can be used to adjust the parameters of the automatic improvement algorithms, such as the threshold of uncertainty, coverage, relevance and difficulty level.

Speech Act Classifier

Speech acts are actions carried by language, including the exterior meaning of the utterance, the intended meaning of the speaker, and the actual effect in conversations. In AutoTutor, speech acts are classified as answers, questions and other expressions. Answers are further classified as, for example, complete answer, partial answer, wrong answer, irrelevant answer, out of domain answer and other customized types. Questions are further classified as definitional question, verification question, concept completion question, etc. Other expressions include greeting, request, meta-communication (e.g., “I didn’t hear it.”), meta-cognition (e.g., “I have no idea about that.”), etc. AutoTutor takes the output of the speech act classifier and determines the next move, such as answering a question, giving feedback to an answer, or responding to other expressions. There has been a substantial body of research in building speech act classifiers for AutoTutor (Moldovan, Li, Rus, & Graesser, 2011; Samei, Li, Keshtkar, Rus, & Graesser, 2014). However, the accuracy is limited in part due to the lack of human annotated training data. For example, in Samei et al. (2014), the precision was 0.53 for answers (labeled as statements in the paper) and 0.63 for questions. The recall was 0.58 for answers and 0.70 for questions. Improved speech act classifiers require a larger corpus of training and test data.

Making speech act classifiers self-improving as more data are collected is a reasonable solution. A self-improving speech act classifier relies on language experts’ feedback. The speech act classifier may present utterances that are hard to classify to language experts for confirmation or correction. The confirmation/correct data then feeds back into the system for model improving.

Answer Classification

A typical AutoTutor interaction is a question from AutoTutor and an answer from a student. Therefore, most of the inputs from students are answers. These answers are usually short. An answer to a main question could be about a paragraph long. An answer to a hint or prompt question usually ranges in length from a word to a sentence long. Evaluating such answers is a research field, called "short answer grading". Burrows et al. (2015) has a good review of the techniques used for short answer grading. Advanced technologies used in short answer grading include LSA (Latent Semantic Analysis), LDA (Latent Dirichlet Allocation) and deep learning neural networks (Basu, Jacobs, & Vanderwende, 2013; Burrows, Gurevych, & Stein, 2015; Cai et al., 2011; Cai, Gong, Qiu, Hu, & Graesser, 2016; Sultan, Salazar, & Sumner, 2016).

There are two typical ways to grade AutoTutor short answers. One way is to compute a score (from 0 to 1) to measure how close the input is to a correct answer. AutoTutor gives a positive feedback (e.g., "Great answer!") for a high score, a neutral feedback (e.g., "Not bad!") for a medium score, and a negative feedback (e.g., "That's not right!") for a low score. Feedback to the grading score is usually short and generic, that is, the feedback does not contain anything specific to the domain problem. An alternative to computing a 0 to 1 score is to categorize the answers. The typical generic categories include "complete answer", "partial answer", "wrong answer", "irrelevant answer", and "out of domain answer". However, any customized categories are possible. Authors may determine what categories to use and associate each category with a specific type of feedback, not necessarily domain independent. The results of both ways are used in selecting next moves.

One of the major challenges in AutoTutor authoring is that it is difficult for authors to imagine what categories of answers may be generated from students without real student data. Therefore, it is difficult for authors to design suitable answer categories in the authoring phase. A self-improving answer evaluation component is therefore necessary for AutoTutor. Figure 1 shows the interface of an AutoTutor self-improving answer classifier (under development). When an existing AutoTutor problem is opened, the tool shows the list of questions (left panel) of the problem. When a question is selected, the answers from all students for the selected question are shown, together with machine identified categories (central panel). An author may change or create an answer's category and associate it with specific feedback. When the classify button is pressed, the tool will rebuild the classifier based on the newly annotated data. Detailed reports about the recall and precision of each category are shown in a popup window (not shown in the Figure).

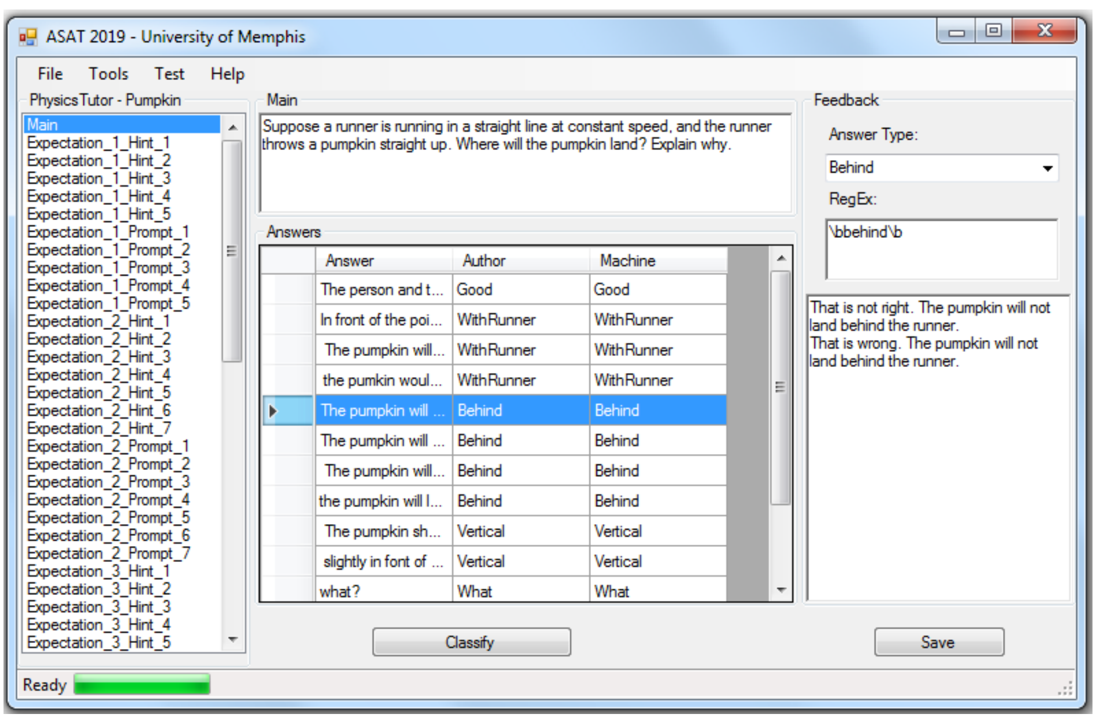


Figure 1. AutoTutor self-improving answer classification interface

Question Answering

There is an extensive body of research on question answering. Mishra and Jain (2016) have provided an excellent summary of this field from 1960's to 2016. In AutoTutor, the QA component is used to answer students' questions. Since students are supposed to answer AutoTutor questions, the proportion of questions in students' inputs is very low. However, when a student question comes up, AutoTutor needs to be able to adequately respond. A response to a student question could be an answer to the question, or, in the case that no answer is available, a statement admitting that AutoTutor does not have an answer. By design, AutoTutor conversations converge to the primary instructional goal of providing the solution to a given problem. However, student questions could lead a conversation to a divergent path. Therefore, AutoTutor is not designed to answer all student questions. Instead, it only answers questions that are highly related to the learning content (Graesser et al., 2012). This is very different from other general QA products that aim at responding to users' divergent questions and do not carry on a deep and coherent conversation (Lally et al., 2012; Mishra & Jain, 2016).

The AutoTutor QA component relies on a relatively small bag of question-answer pairs for each problem. Such bags are initially prepared by content authors. The self-improvable QA component allows authors to review questions that come up from real students and expand the bags to cover most important questions that need to be answered. Furthermore, the QA component has an evaluation model that identifies the question types and matches the question to an existing QA pair, if it exists. The QA component also allows authors to annotate new questions for self-improving the QA classifier.

Discussion and Recommendations

Self-improving systems are supposed to be "automatically" improving without "human's intervention" when they act in the "world" (Omohundro, 2007). In the case of ITSs, such as AutoTutor systems, the "world" is the users, including learners, teachers and content authors. In other words, ITSs act on the human. Therefore, the data that is used for self-improving is from humans. ITS self-improving processes include, (1) collecting data from students, teachers and authors, (2) computing system performance indicators (e.g., question quality, answer evaluation accuracy, etc.), and (3) rebuilding components that can be improved.

ITSs use advanced components, such as the QG, SAC, QC and QA components used in AutoTutor systems. Further advances of the technologies may provide better algorithms to these components. A self-improving ITS may need to replace the existing components with completely new ones, instead of tuning the performance of the existing components. To reduce the development cost, ITSs should make such components easily replaceable. Further, the data collected should be able to support self-improvement of new components. This is possible only when the components are standardized.

As a general framework for ITSs, GIFT should provide standardized ITS components that are replaceable and self-improving. In the current GIFT system, most authoring tools aim at authoring high performance systems without using data after the systems are used. Future GIFT authoring tools should be designed to allow self-improvement. Domain experts should have access to complete learning data and have easy ways to give feedback to the system, such as correcting a speech act or answer category, adding or revising questions and answers, etc. One possible problem is that the continuous involvement of authors may further increase the cost of ITS development. However, this may be compensated by the increase of users due to the improvement of overall system performance.

Acknowledgements

The research was supported by the National Science Foundation (SBR 9720314, REC 0106965, REC 0126265, ITR 0325428, REESE 0633918, ALT-0834847, DRK-12-0918409, 1108845; ACI-1443068), the Institute of Education Sciences (R305H050169, R305B070349, R305A080589, R305A080594, R305G020018, R305C120001), the Army Research Lab (W911INF-12-2-0030), and the Office of Naval Research (N00014-00-1-0600, N00014-12-C-0643; N00014-16-C-3027). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF, IES, or DoD. Examples of AutoTutor can be found at http://www.AutoTutor.org).

References

Basu, S., Jacobs, C., & Vanderwende, L. (2013). Powergrading: A clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, *1*, 391–402.

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, *13*(6), 4-16.

Burrows, S., Gurevych, I., & Stein, B. (2015). The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, *25*, 60-117.

Cai, Z., Gong, Y., Qiu, Q., Hu, X., & Graesser, A. (2016). Making autotutor agents smarter: Autotutor answer clustering and iterative script authoring. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10011 LNAI*, 438–441.

Cai, Z., Graesser, A. C., & Hu, X. (2015). ASAT: AutoTutor Script Authoring Tool. In R. Sottilare, A. C. Graesser, X. Hu, & K. Brawner (Eds.), *Design recommendations for Intelligent Tutoring Systems: Authoring Tools* (pp. 199–210). Orlando: Army Research Laboratory.

Cai, Z., Graesser, A., Forsyth, C., Burkett, C., Millis, K., Wallace, P.,Halper, D., & Butler, H. (2011). Trialog in ARIES: User input assessment in an intelligent tutoring system. in W. Chen & S. Li (Eds.), *Proceedings of the 3rd IEEE International Conference on Intelligent Computing and Intelligent Systems*, (pp429–433). Guangzhou: IEEE Press.

Cai, Z., Rus, V., Kim, H. J. J., Susarla, S. C., Karnam, P., & Graesser, A. C. (2006). NLGML: A Markup Language for Question Generation. In T. Reeves & S. Yamashita (Eds.), *Proceedings of E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2006*, (pp. 2747–2752). Honolulu, Hawaii, USA: Association for the Advancement of Computing in Education (AACE).

Duan, N., Tang, D., Chen, P., & Zhou, M. (2017). Question Generation for Question Answering. In M. Palmer, R. Hwa, & S. Riedel (Eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, (pp. 866–874). Copenhagen, Denmark: Association for Computational Linguistics.

Graesser, A.C. (2016). Conversations with AutoTutor help students learn. *International Journal of Artificial Intelligence in Education*, *26*.124-132.

Graesser, A., Rus, V., & Cai, Z. (2008). Question classification schemes. In *Proc. of the Workshop on Question Generation* (pp. 10-17).

Graesser, A.C., Hu, X., & Sottilare, R. (2018). Intelligent tutoring systems. In F. Fischer, C. E. Hmelo-Silver, S. R. Goldman, and P. Reimann (Eds.), *International handbook of the learning sciences* (pp. 246-255). New York: Routledge.

Graesser, A. C., Lu, S., Jackson, G. T., Mitchell, H. H., Ventura, M., Olney, A., & Louwerse, M. M. (2004). AutoTutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, & Computers*, *36*(2), 180–192.

Graesser, A. C., Rus, V., Cai, Z., & Hu. X. (2012). Question answering and generation. In P. McCarthy and C. Boonthum-Denecke (Eds.), *Applied natural language processing: Identification, investigation, and resolution* (pp. 1-16). Hershey, PA: IGI Global.

Lally, A., Prager, J. M., McCord, M. C., Boguraev, B. K., Patwardhan, S., Fan, J., … Chu-Carroll, J. (2012). Question analysis: How Watson reads a clue. *IBM Journal of Research and Development*, *56*(3.4), 2:1-2:14.

Millis, K., Forsyth, C., Butler, H., Wallace, P., Graesser, A., & Halpern, D. (2011). Operation ARIES!: A Serious Game for Teaching Scientific Inquiry. In M. Ma, A. Oikonomou, & L. C. Jain (Eds.), *Serious Games and Edutainment Applications* (pp. 169–195). London: Springer London.

Mishra, A., & Jain, S. K. (2016). A survey on question answering systems with classification. *Journal of King Saud University - Computer and Information Sciences*, *28*(3), 345–361.

Moldovan, C., Rus, V., & Graesser, A.C. (2011).Automated Speech Act Classification For Online Chat. In S. Visa (Ed.), *Proceedings of the 22nd Midwest Artificial Intelligence and Cognitive Science Conference*, (pp. 23–29).

Nye, B. D., Graesser, A. C., & Hu, X. (2014). AutoTutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education*, *24*(4), 427–469.

Olney, A. M., Graesser, A. C., Person, N. K., Piwek, P., & Boyer, K. E. (2012). Question generation from concept maps. *Dialogue & Discourse*, *3*(2), 75–99.

Omohundro, S. M. (2007). The Nature of Self-Improving Artificial Intelligence. http://selfawaresystems.com/2007/10/05/paper-on-the-nature-of-self-improving-artificial-intelligence.

Rus, V. & Graesser, A.C. (Eds.). (2009). *The question generation shared task and evaluation challenge*. ISBN: 978-0-615-27428-7

Samei, B., Li, H., Keshtkar, F., Rus, V., & Graesser, A. C. (2014, June). Context-based speech act classification in intelligent tutoring systems. In *International Conference on Intelligent Tutoring Systems* (pp. 236-241). Springer, Cham.

Smith, S., & Sherwood, B. (1976). Educational uses of the PLATO computer system. *Science*. https://doi.org/10.1126/science.769165

Sultan, M. A., Salazar, C., & Sumner, T. (2016). Fast and Easy Short Answer Grading with High Accuracy. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (pp. 1070–1075).

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, *46*(4), 197–221.

VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., & Rosé, C. P. (2007). When are tutorial dialogues more effective than reading ? *Cognitive Science*, *31*(1), 3–62.

SECTION IV – SOCIAL PERSPECTIVES AND HUMAN FACTORS

Dr. Arthur C. Graesser, and Dr. Anne M. Sinatra, Eds.

Chapter 14 – Introduction to the Social Perspectives and Human Factors Section

Arthur C. Graesser[1] and Anne M. Sinatra[2]
[1]University of Memphis, [2]U.S. Army Combat Capabilities Development Command - Soldier Center – Simulation and Training Technology Center

Core Ideas

The chapters in this section focus on the different roles that humans play in the development of self-improving adaptive instructional systems (AISs). The underlying assumption is that humans need to be in the loop in a hybrid co-evolution of human and computational machine intelligence. The humans may be learners, instructors, experts, designers, and other stakeholders. The computational system may range from ideal algorithms or procedures to solutions from data-driven machine learning. A hybrid is needed because of the limitations of brute force reinforcement learning and elegant computational theories.

The value of a hybrid self-improving system is confirmed when it improves learning, engagement, system usage, and the learners' impression of the AIS. It is important to conduct studies that perform these evaluations because such data are ultimately the gold standard. Sometimes there is no alternative than to adopt the judgments of humans because the learning environment is extremely complex. Machine learning can then be applied to tune, modify, and improve the performance. Sometimes human judgments are faulty so it is better to adopt intelligent algorithms that have no resemblance to any human judgments. Whenever humans are in the loop, a sensible question to ask is: What humans? Is it a single knowledgeable genius, a panel of relevant experts, or a crowd of novices? Are there times when a crowd of novices generate a solution that is equal to or better than a small number of experts who handcraft a solution? What happens when people disagree? Assessments with suitable gold standards are needed to answer these questions.

Judgments of humans are highly influenced by the visual depiction of the data or system components being analyzed. Learners can improve self-regulated learning activities when they can view their own performance and compare it to their peers. Instructors can take concrete steps to help the students when they can view the students' performance on particular lessons, knowledge, and skills. Instructional designers can improve the content and curriculum when they view performance profiles of the learners. The dashboards that each of these stakeholders view will hopefully be influenced by today's science of human factors and computer-human interaction.

Individual Chapters

The chapter by ***Brusilovsky and Rus*** presents a social navigation approach to self-improving systems that capitalizes on previous learners' search patterns on web facilities, solutions to problems, annotations of interfaces, and other behavior of the crowd of previous users. There are visual depictions of other learners' behavior that are tracked by the system and provide guidance to learners. A particular learner can compare his or her own behavior to the wisdom of the crowd and the crowd's performance on problems of varying difficulty. This social navigation approach has often shown improvements in learning or performance over hand-crafted systems developed by experts.

The chapter by ***Hampton and Baker*** discusses dashboards for instructors and how they can play a role in the evolution of self-improving adaptive instructional systems. Instructors can use this information as feedback to make adjustments to future class lessons, cluster students, help individual learners, and modify content. The actions the instructors take can then be used in educational data mining and machine learning analyses to modify automated components of AISs. In order to implement this approach, there needs to be a systematic analysis of the instructors' activities, many of which are documented in this chapter.

The chapter by ***Gitinabard, Lynch, Long, Meyer, and Woodward*** proposes that a social media framework be integrated with the Generalized Intelligent Framework for Tutoring (GIFT) so that there can be information about communication among students, between students and instructors, and between computer environments and people. Learning communities often introduce fellow learners to new ideas and maintain their engagement over time. Therefore, the various platforms of social media need to somehow be integrated with GIFT, with data logged, tracked, and analyzed for future system improvement. The interface of the learning environment should also expose learners to other stakeholders and allow them to give feedback on their experiences.

The chapter by ***DeFalco and Doty*** raises issues about the morals and ethics of decisions, algorithms, and procedures of systems that are automated in AISs. Sometimes there are conflicts in what path to take, as the authors impactfully point out in conflicts between obeying orders of leaders in the military and following moral or ethical models. Conflicts may occur at the macrolevel, the macrolevel, and aptitude-treatment interactions. The evolution of a self-improving system needs to expose potential conflicts for humans to resolve.

The chapter by ***Graesser, DeFalco, and Cockroft*** raises questions about the quality of human judgments and computational algorithms in guiding self-improving adaptive systems. The chapter documents an alarming lack of compatibility between teachers, data, and the learning sciences, so there needs to be a serious examination of the quality of the decisions that guide automated changes. At the very least, the quality of the different sources of judgments and recommendations need to be continuously evaluated with respect to predicting learning, engagement, use, and learner impressions of an AIS.

Chapter 15 – Social Navigation for Self-Improving Intelligent Educational Systems

Peter Brusilovsky[1] and Vasile Rus[2]
[1]University of Pittsburgh, [2]University of Memphis

Introduction

The idea of self-improving intelligent educational systems is almost as old as the field of intelligent educational systems itself. The origin of this research stream could be traced to the paper of Tim O'Shea "A self-improving quadratic tutor" (O'Shea, 1982), which was published 40 years ago in the famous special issue of *International Journal on the Man-Machine Studies*. This special issue later was re-published in a separate book (Sleeman & Brown, 1982), which arguably launched the very field of intelligent tutoring systems (ITSs) and defined its research agenda for years ahead. O'Shea's paper was somewhat different from the rest of the papers in the special issue. Unlike the majority of researchers who believed that intelligent educational systems should be created by domain experts through knowledge elicitation and engineering, O'Shea argued that an intelligent system should be able to improve itself not just by constant engagement of experts but by using data collected in the process of its practical application. While his self-improving tutor started with an expert-engineered teaching strategy in the form of production rules and assertions, the system also included a pro-active self-improvement cycle. The idea of this cycle was to select an educational objective, make an experimental change in teaching strategy, statistically evaluate the resulting performance over time, and make necessary updates if the change is successful.

While the idea of self-improving ITSs and the original paper produced a stream of follow-up work, for many years this stream was really small and not easily visible in a large body of work on intelligent educational systems. The main obstacle was a rather low level of practical use of these systems. Without a large number or real users working with an intelligent system year by year the idea of automatic experimentation and constant improvement was hard to implement. The situation, however, gradually changed over the years. As the field of ITSs became more mature, some systems, like the famous Algebra Tutor, got exposed to hundreds and thousands of real users year by year (Koedinger, Anderson, Hadley, & Mark, 1997) and the need to constantly improve the knowledge representation and algorithms behind these systems was brought back to the agenda of ITSs researchers. Several research teams demonstrated that learning data routinely collected by ITSs could offer valuable insights on how the systems could be improved and suggested specific approaches to data-driven improvement of ITSs (Martin, Mitrovic, Koedinger, & Mathan, 2011; Pavlik, Cen, Wu, & Koedinger, 2008).

Following this pioneering work, the use of learner data to improve the performance of ITSs and other educational systems (i.e., MOOCs) gradually emerged as one of the most popular topics of research in several research communities including Artificial Intelligence in Education (AI-Ed), Educational Data Mining (EDM), and Learning Analytics and Knowledge (LAK) with dozens of papers published every year. Yet, the absolute majority of research on this topic focuses on just one way of using this data. Whether learner data is used to improve domain models or to adjust parameters of student modeling and personalization approaches, the focus is on enhancing the "machine" intelligence side of ITSs. Yet, every intelligent system could be improved in two different ways. One way, indeed, is to enhance the internal intelligence of the system. The other way, however, is to empower the intelligence of the system user through more advanced and intelligent interfaces. While developers of intelligent systems frequently over-focus on enhancing internal functionality of the systems, the research in the area of intelligent user interfaces demonstrate that augmenting human intelligence through a more powerful, AI-driven interface could remarkably improve the overall efficiency of an intelligent system. In other words, best results could be achieved when human and artificial intelligence work together, not when all efforts are spent on improving the Artificial Intelligence (AI) and the power of human intelligence is wasted due to a primitive interface. Getting back to the problem of improving ITSs using data of past learners, an interesting challenge is how this data could be used to advance the interface side of ITSs so that it could empower human learners, better engage them into interacting with the system, and improve the overall performance. In this chapter, we review one of the approaches, which could efficiently use data of past learners to offer a more efficient interface for future learning: social navigation. In the following sections we introduce the idea of social navigation (Farzan & Brusilovsky, 2018) and review several studies exploring social navigation in different contexts.

Social Navigation

Social navigation is a group of approaches belonging to a broader field of *social information access* (Brusilovsky & He, 2018). Social information access can be formally defined as a stream of research that explores methods for organizing users' past interaction with an information repository in order to provide better access to information to the future users. Various information traces left by past users of interactive systems form highly valuable "community wisdom", which could be harnessed to support various kinds of information access such as search, browsing, and recommendations. Within this stream of research, social navigation approaches (Farzan & Brusilovsky, 2018) focus on using "community wisdom" to assist their users in the process of browsing and navigation, i.e., selecting the most relevant information item or link among many possible options.

The ideas of social navigation are frequently traced back to the pioneer Read Wear and Edit Wear system (Hill, Hollan, Wroblewski, & McCandless, 1992). This system visualized the history of authors' and readers' interactions with a document enabling new users to quickly locate the most viewed or edited parts of the document. Social navigation in the information space as well as the term social navigation was introduced two years later by Dourish and Chalmers as "moving towards cluster of people" or "selecting subjects because others have examined them" (Dourish & Chalmers, 1994). The pioneer systems Juggler (Dieberger, 1997) and Footprints (Wexelblat & Mayes, 1999) used the ideas of social navigation to help users navigate in two kinds of information spaces – a Web site and a text-based virtual environment (known as MUDs and MOOs). Both systems attempted to visualize "wear" traces left by system users in order to guide future users. In addition to this indirect social navigation, Juggler also implemented several types of direct social navigation (for example, allowing users to guide each other directly through chat). This allowed Dieberger (1997) to start the process of generalizing the ideas of social navigation. Further generalization of the field of social navigation was propelled by several workshops, which gathered like-minded researchers, and publications, which stemmed from these workshops (Dieberger, Dourish, Höök, Resnick, & Wexelblat, 2000; Höök, Benyon, & Munro, 2003; Munro, Höök, & Benyon, 1999). As a result of this active ideas exchange, the understanding of what forms the "community wisdom" in social navigation systems was considerably expanded to include a variety of options – from past user "clicks" to rich explicit feedback and resource annotations.

In the context of learning and education, the ideas of social navigation have been introduced in the context of research on Web-based education. Early generation of Web-based education systems (Khan, 1997) supported the learning process by providing learners with access to a variety of educational resources. In this context, it was natural to explore the technology of social navigation, which was known to help users in accessing most appropriate information. First attempts to introduce social navigation in Web-based education has been made by Dron, Boyne, Mitchell, and Siviter (2000) and Kurhila, Miettinen, Nokelainen, and Tirri (2002). The EDUCO system which was built by Kurhila and his colleagues (Kurhila et al., 2002) could be considered as a classic example of exploring the ideas of social navigation in the education context.

EDUCO was a collaborative learning environment which implemented social navigation support to enrich learners' experiences in Web-based learning. EDUCO supported synchronous social navigation by visualizing the presence of others in the learning environment. As users of the system were accessing the educational Web documents, others can view their presence as dots next to the documents in a visualized document space (Figure 1). The color of the documents represented the popularity of the document among the users based on how many times they have been clicked. Furthermore, users can leave comments associated with documents that are visible to others navigating to the document.

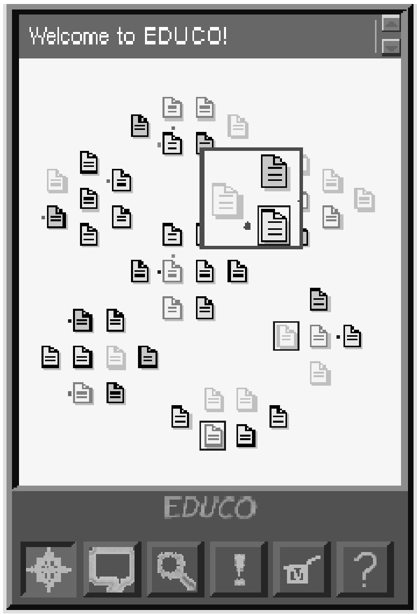


Figure 1. Representation of documents and users within the EDUCO learning environment.

The early examples of educational social navigation and the increased popularity of research on "collective wisdom" and social information access helped to engage several other research teams working on similar topics. In just a few years, the number and the diversity of explored social navigation approaches in educational context increased remarkably (Brusilovsky, Chavan, & Farzan, 2004; Hübscher & Puntambekar, 2004; Mitsuhara, Kanenishi, & Yano, 2004; Tattersall et al., 2004; Vassileva, 2004). Since that time, both the variety and the complexity of research on this topic has been gradually increasing. However, due to the practical focus of this chapter, we do not intend to provide a comprehensive overview of this research stream. Instead, we focus on three well-explored and extensively used systems, which applied different kinds of social navigation to educational processes. We believe that a review of these systems can provide both, a list of useful social navigation techniques and a demonstration how the research on social navigation in the educational context has gradually advanced from simple ideas explored in proof-of-concept systems to more complex designs validated by large-scale field studies.

Knowledge Sea II

Knowledge Sea II (Brusilovsky et al., 2004), originally developed in 2003-2005, provides a good example of how early ideas of "traffic-based" social navigation explored in the pioneer systems Juggler (Dieberger, 1997) and Footprints (Wexelblat & Mayes, 1999) could be applied in an educational context. Knowledge Sea II uses ideas of social navigation to support both browsing and visualization access to information. The visualization-based access is provided through an 8 by 8 cell-based map of the information space. This map is assembled using Kohonen's Self-Organized Map (SOM) technology (Kohonen, 1995) from about 25,000

Web pages devoted to C programming language. Every cell on a resulting map provides access to a subset of these pages. By clicking on a cell, the user can open it and get access to the set of pages located in this cell (Figure 2). An interesting property of SOM technology is that it places similar pages into the same or adjacent cells on the map, so the result presents a reasonably good semantic map of the information space. The cells of the map are marked by keywords, which are most frequently found in the corresponding pages of each cell and by landmark resources located in the cell. The map itself was re-used from the earlier version of the system, Knowledge Sea (Brusilovsky & Rizzo, 2002). In the Knowledge Sea II project, we added a layer of social navigation on top of the map.

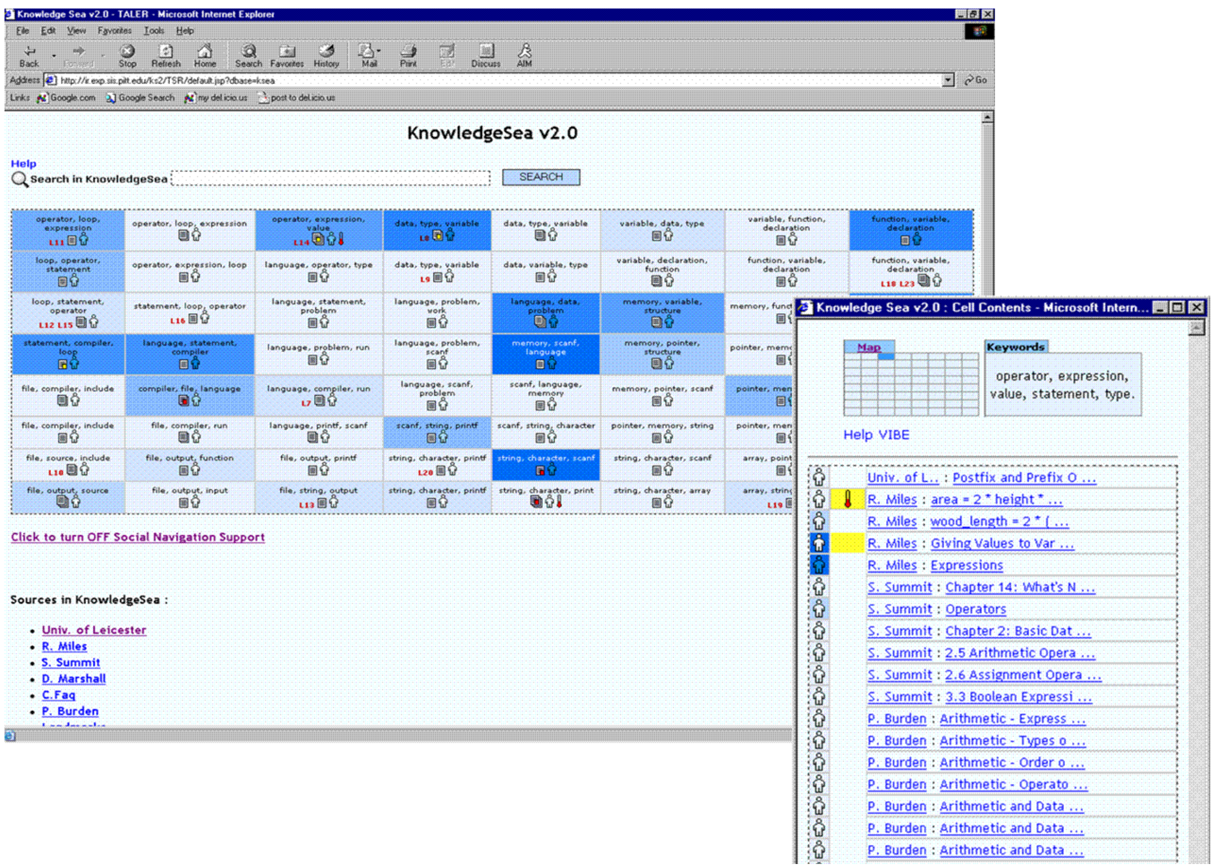


Figure 2. Social navigation support in the Knowledge Sea II system. The knowledge map is shown on the top left and an opened cell is shown on the right. The list of links to the tutorial roots is shown on the bottom left. A darker blue background indicates documents and map cells that have received more attention from users within the same group. Human icons with darker colors indicate documents and cells that have received more attention from the user herself.

The browsing-based access is provided through the hierarchical structure of the C programming tutorials assembled by the system. Each tutorial site is organized as a tree with table of contents, sections, and subsections. The home page of Knowledge Sea II provides access to the root pages of all these tutorials. Starting from that, users can navigate down to the sections or subsections of interest assisted by social navigation visual cues (Figure 3).

The community wisdom in Knowledge Sea II is collected by tracking two kinds of page-centric user information: timed page visits (traffic) and page annotations. This information is used to generate a historyenriched environment with two types of visual cues, which change the appearance of links on the pages and map cells presented to the user (Figure 3). These cues are based on the two kinds of tracked information and are known respectively as traffic- and annotation-based social navigation support. The system generates appropriate cues individually for each user by analyzing past individual activities of the user and other users belonging to the same group.

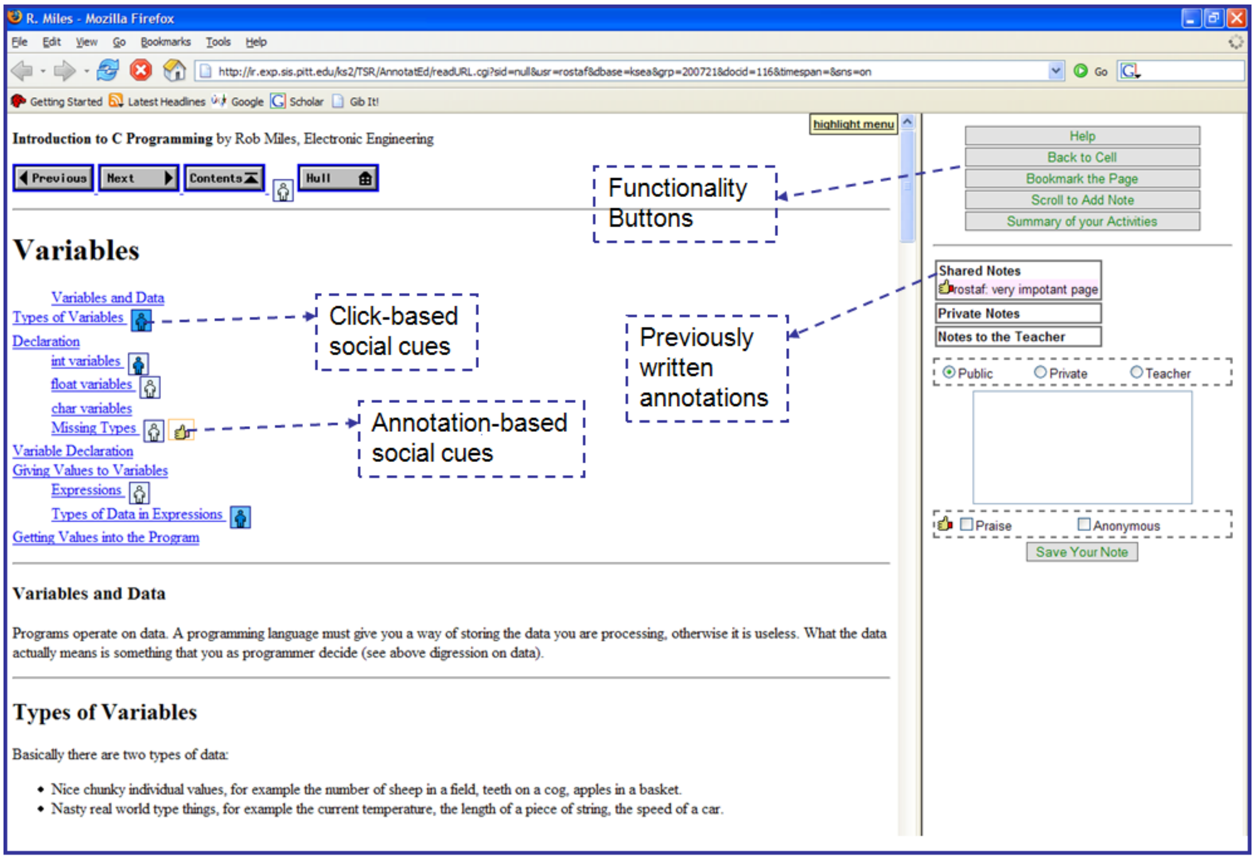


Figure 3. Social navigation in C-programming tutorial pages, from (Farzan & Brusilovsky, 2008).

Traffic-based navigation support attempts to express how much attention the user herself and other users from the same group paid to each of the 25,000 pages that the system monitors. The level of attention for a page is computed by considering both number of visits and time spent on the page and is displayed to the user through an icon that shows a human figure on a blue background. The color saturation of the figure expresses the level of the user's own attention while the background color expresses the average level of group attention. The higher the level of attention is, the darker the color appears to the user. The contrast between colors allows the user to compare her navigation history with the navigation of the entire group. For example, a light figure on a dark background indicates a page that is popular among group members but remains under-explored by the user. The color of the map cell and the human figure shown in the cell is computed by integrating attention parameters of all pages belonging to that cell.

Annotation-based navigation support uses a similar approach to represent the number of page annotations made by the users from the same group. Users can annotate each page in the system. Users can also indicate that a note is praise (i.e., the page is good in some aspect). While users make annotations mainly for themselves, Knowledge Sea II allows all users of the same group to benefit from collective annotation behavior.

The yellow annotation icon shown next to the blue traffic icon shows the density and the "praise temperature" of annotations for each page. The more annotations a page has, the darker the yellow background color appears to the user. The temperature shown on a thermometer icon indicates the percentage of praise annotations.

Both types of social visual cues were provided to guide users to the most relevant and useful pages as implicitly indicated by the past users' activity. Traffic-based social navigation was provided in the very first version of Knowledge Sea II (Brusilovsky et al., 2004) and could be considered as a direct application of the early ideas of social navigation in education contexts. Annotation-based social navigation was added in the second version (Farzan & Brusilovsky, 2005b). This feature was motivated by our experience with the first version. As we found during the first classroom studies, despite its overall effectiveness, traffic-based social navigation was subject to the *avalanche effect*, which has not been well-studied at that time. User clicks and page visits were important, but not reliable signs of user attention and page importance. Frequently, users clicked on a less relevant page by mistake, attracted by a seemingly relevant title. After landing on the page and realizing that it is not helpful, the first visitor backed away. Yet, with traffic-based navigation, every visit left a visible trace: the page link annotation became darker, further increasing a chance to be visited by future users. As we discovered, a simple version of traffic-based social navigation lead to creating some "tar pits", low-value pages with attractive titles, which were falsely indicated as important by social navigation. The addition of more reliable annotation-based social navigation and developing a smarter time-based approach to score user page visits (Farzan & Brusilovsky, 2005a) resolved this problem.

The advanced version of Knowledge Sea II with dual sources of social navigation support has been explored in many classroom studies. In these studies, we were able to discover and confirm several effects of social navigation. We found that a community of students was remarkably good in co-discovering the most important and valuable pages in the context of the course. Note that only a part of the 25,000 pages extracted from multiple tutorials were relevant and useful for our specific C programming course. Even in the classes that started with an empty map, we were able to observe that most relevant pages and their clusters were discovered relatively fast, creating a class-adapted map to guide future users. Moreover, the ability to annotate pages and the visualization of annotations through visual cues could considerably increase a chance for an important page to be noticed. We also found that social visual cues highly influence user navigation behavior. Pages which attracted past attention of the users – as revealed by visual cues – have a significantly higher chance to be re-visited by users who already explored them and visited by new users. In fact, very popular pages visualized by the displayed density of visits and annotations, were more attractive for the users than top results in a ranked search list. As we found in a study of social search in Knowledge Sea II, adding social visual cues to the ranked list of search results shifts user's attention from top-3 results in the list to the most popular pages in the list. We also found that the presence of annotation-based cues doubled a user's chance to follow a specific link. It was clear that the users considered annotation-based cues as more indicative and reliable in finding useful pages. Following our success in using social navigation in Knowledge Sea II, we re-used both explored social navigation approaches in another project (Farzan, Coyle, Freyne, Brusilovsky, & Smyth, 2007.) An extensive report of our findings in both projects is available in Farzan and Brusilovsky (2008).

Progressor

Our experience with social navigation in the Knowledge Sea II project, revealed the importance of the reliability of "social wisdom". Comparing traffic-based traces with annotation-based traces of past behavior, we discovered that actions that require higher-level commitments from the past users are both more reliable in discovering important pages and more influential for the future users. In the Progressor project (Hsiao, Bakalov, Brusilovsky, & König-Ries, 2013), we explored another form of high-commitment traces of behavior: the problem-solving traces of students taking the same course. The work on Progressor followed our past attempts to combine open student modeling (Bull & Kay, 2007) and adaptive navigation support (Brusilovsky, 2007) to help the user in accessing the most relevant problems in a programming course. In our first attempts, we explored traditional knowledge-driven adaptive navigation support where personalized guidance decisions were made on the basis of manually engineered domain models and personalization algorithms (Hsiao, Sosnovsky, & Brusilovsky, 2010). While we found it highly efficient and engaging (Sosnovsky & Brusilovsky, 2015), our concern was that the knowledge-based approach required considerable engagement of domain experts. By replacing traditional knowledge-based navigation support with social navigation support, we hoped that the "community wisdom" could provide an alternative source of knowledge for efficient navigation. On the way to finding the most appropriate way to process and visualize past problem solving behavior in such a way that it could provide efficient help for future users, we explored a sequence of design options (Brusilovsky, Hsiao, & Folajimi, 2011; Hsiao, Bakalov, Brusilovsky, & König-Ries, 2011; Hsiao et al., 2013). The Progressor system reviewed in this section was the last and the most efficient design in this sequence.

The design of Progressor was motivated by the ideas of Open Social Student Modeling (OSSM) and the theories of Social Comparison and Self-Regulated learning. OSSM can be considered a social extension of open student modeling. Open student modeling has been suggested as a way to externalize student models, the key component of any personalized learning system. While in a traditional personalized learning system this model is usually hidden from the student and only used by the personalization engine to provide adaptation effects, systems with an open student model expose this model to the learner and provide an interface for its exploration and possible editing. Open student modeling is known for a number of positive effects. It increases the transparency of personalization, helps raise the students' awareness of their learning performances, and supports meta-cognitive processes (Bull & Kay, 2013). In combination with adaptive navigation support, it can also efficiently guide students to the appropriate content (Sosnovsky & Brusilovsky, 2015). In this context, the idea of OSSM is simply to make the content of individual and student models accessible not only to the target student herself, but to a broader group of students, for example, students in the same class. The most natural way to do it is through social visualization that can visually present the content of multiple student models to the target student in a form that enables comparison of her own knowledge to the knowledge of her peers and the class as a whole.

Research in self-regulated learning examines students' metacognitive strategies for planning, monitoring, and modifying their management and control of their effort on classroom academic tasks (Pintrich & De Groot, 1990). Self-regulated learning involves self-monitoring to optimally interpret feedback from their academic learning process and environment (Zimmerman, 1990). Our work aimed to leverage awareness, motivation, and content organization through social visualizations in the hopes of promoting students' self-regulated learning behavior. Research in social comparison (Festinger, 1954) has demonstrated that people often determine appropriate behavior for themselves by examining the behavior of others, especially similar others (Buunk & Gibbons, 2007). Consequently, it has been shown that individuals tend to behave similarly to their friends and peers (Cialdini, Wosinska, Barrett, Butner, & Gornik-Durose, 1999). Researchers and designers of online systems have used the insights from social comparison research in the study of online social behavior. In the educational domain, the positive impact of social comparison on student performance has been reported in several papers (Light, Littleton, Bale, Joiner, & Messer, 2000). However, the value of social comparison in the context of personalized learning and navigation support has not been studied. Based on the past studies, we hoped that social navigation design that directly engages social comparison could increase its impact and positive value.

Figure 4 shows the Progressor interface. The visualization consists of two panes: the left pane displays the student's own progress and the right one displays the progress of any class peer or the whole class, whichever is selected from a dropdown menu. Each pane visualizes the respective student's progress as a pie chart. The pie chart representation visually conveys the chronological order of lectures while the size of a sector represents the number of problems for each lecture. A lecture may consist of one or several topics, which are represented as angular segments placed within the circular sector of the corresponding lecture. This representation allows the student to easily estimate the amount of work on each individual topic or lecture, while an apparent topical sequence provides a good picture of progress through the course. In addition to that, the ability to view someone else's progress allows the student to quickly find the peers who can help with a difficult topic or quiz. Finally, the ability to view the average progress of the entire class allows the student to relate her progress to that of the whole class and estimate whether she is ahead or behind of the class. In addition to serving as OSSM, the Progressor interface provided direct access to learning content. Clicking on any topic on the student's own model (Figure 4, left) or on a peer or class model (Figure 4, right) opened a list of practice problems available for this topic. Links to problems have also been socially annotated using the same color-coding scheme.

Figure 4. Peers model comparison and social navigation support interfaces in Progressor. The color of course topics indicate a student's own progress with the topic knowledge (left) and class or peer progress (right). A click on a specific topic on either side opens a list of practice problems for the topic.

From a semester-long study cross-compared with previous attempts to organize access to Java problems, we learned that the new design of the OSSM interface was very engaging. Students used Progressor extensively. On average, it achieved the highest system usage across all OSSM interface designs surpassing even the former champion, JavaGuide (Hsiao et al., 2010). Progressor also engaged students to explore more topics and to work on more distinct questions. In addition, the amount of time spent on the system (in terms of the sessions) was doubled. To check whether the boost of usage could be credited to the new design, we examined student interaction with the peer side of the Progressor interface such as re-sorting, scrolling, and accessing the peer list. As before, we found that students interacted with the peer side quite considerably, comparing their progress with the progress of peers and accessing a considerable volume of content from the peer side. Moreover, the more students engaged in interacting with the social features of Progressor, the more likely they were to achieve a higher success rate in answering the self-assessment questions. The findings were consistent with the subjective evaluation outcome, which demonstrated high satisfaction with Progressor (Hsiao et al., 2013).

Mastery Grids

Following the success of Progressor and the discovered value of combining social navigation, open learner modeling, and social comparison ideas within the same design, we attempted to expand these ideas to a more realistic online learning context. One serious limitation of Progressor was its focus on one type of learning content, which in our past studies was one type of programming problems. In a more typical online learning situation, the student has access to multiple types of learning content: readings, worked examples, questions, problems, etc. The first attempt to expand the ideas of Progressor to multiple types of content was done in the Progressor+ system (Hsiao & Brusilovsky, 2017). Following the encouraging results of its evaluation, we developed Mastery Grids, an open-source domain-independent framework for OSSM and social navigation (Loboda, Guerra, Hosseini, & Brusilovsky, 2014).

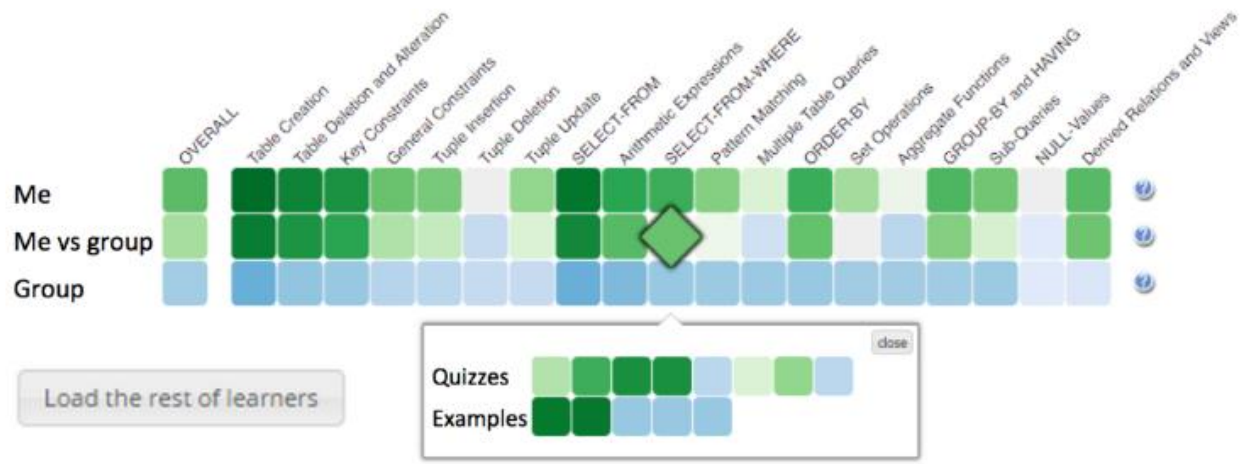


Figure 5. MasteryGrid interface for a Database course.

MasteryGrids uses a grid-based social visualization approach pioneered in Progressor+, which allows easy comparison of the progress of the student against peer students or against the aggregated progress of all students of the class. MasteryGrids uses cells of different color saturation to show knowledge progress of the target student, her reference group, and other students over multiple kinds of educational content organized by topics. Figure 5 shows the “collapsed” version of MasteryGrids’ interface for a database management course. Left to right, the first column of the grid ("OVERALL") shows student average progress, and the remaining columns show student knowledge progress topic by topic starting from the first topic of the database course: "Table Creation". The collapsed version of the OSSM grid includes 3 rows. The first row of the grid (Me) presents the topic-by-topic knowledge progress of the *current student* and uses green colors of different saturation to represent the level of progress (the darker the color, the higher the progress). The third row (Group) shows the aggregated topic-by-topic progress of the reference group (in this case, the whole class) using blue colors of different saturation. The second row (Me vs. Group) presents a topic-by-topic difference between the student progress and the class progress. The cells in the second row are green if the student’s knowledge progress is higher than the class, blue if the class is ahead, and gray when both the student and the rest of the class have the same progress. Higher color saturation indicates a larger difference. MasteryGrids can be configured to disable the OSSM features turning it into a standard Open

Student model, as can be seen in Figure 5. In the Open Student Model version only the first row with the progress of the current student is shown.

By clicking on any topic cell, the student can access learning content associated with the topic. For example, in Figure 5, the student has clicked in a cell of the topic SELECT-FROM-WHERE and the system displays two kinds of learning content available for this topic (quizzes and examples) in two rows of content items represented as colored cells. By clicking in the content cells, the content (problem or example) will be loaded in an overlaid window. The student can access the content by clicking on any of the three rows of the topic (i.e., Me, Me vs. group, or Group). The row clicked defines whether the colors of content cells (Quizzes/Examples) will represent individual progress, comparison between the individual and the group, or the group progress. For example, in Figure 5, the student clicked in the second, differential progress row. Thus, the colors of the content cells also show differential progress (resulting in both green and blue cells.)

The "collapsed" version of the interface is the simplest one available for students. In addition to displaying the overall class progress, MasteryGrids can display and compare progress for each or all types of content. For example, Figure 6 shows an expanded comparison interface for a Java programming course. Here the upper grid (green) shows student's own knowledge progress within each type of content, the bottom (blue) grid shows class progress, and the middle grid allows detailed comparison for each combination of topic and content. The full interface of Mastery Grids allowed the students to choose which resources are visualized and which peer group is used for social comparison. For example, in Figure 6, the student selected "class average" as a basis for comparison, but there are many other options, like top 10 students, upper part of the class, lower part of the class, etc. The interface also provides an option to show the full anonymized ranked grid of individual students with their progress over the course topics. The position of the current student in the list is highlighted to make the overall class standing clearer.

Figure 6. An expanded version of Mastery Grids interface for a Java programming course displaying and comparing progress over different types of content.

Mastery Grids interfaces have been developed for Java (Guerra, Hosseini, Somyurek, & Brusilovsky, 2016), Database (Brusilovsky et al., 2016), and Python (Brusilovsky et al., 2018) courses and extensively studied in these contexts in many classroom studies. To date, the most extensive study has been done in a database course with over a hundred students (Brusilovsky et al., 2016) where the version of Mastery Grids shown in Figure 5 was offered as a non-mandatory practice system to be used during students' study time. Our most valuable discovery from this study is a remarkable ability of the social navigation and comparison interface to engage and retain students, as compared with a more traditional open student model interface without the social component. OSSM motivated students to perform significantly more work with non-mandatory learning content. In addition, social visualization enabled students in the OSSM group to work more efficiently, which could be attributed to the social navigation aspect of our OSSM implementation. Working with OSSM also positively impacted student learning, significantly improving the learning gain of weaker students. This could be attributed to the increased work with the content (as shown by the correlation between the amount of work and exam grade). While it is hardly surprising that more work with learning content resulted in better learning, it is impressive that we were able to achieve this effect with non-mandatory educational content, which the students explore at their own will.

Social Navigation in Dialogue-based Intelligent Tutoring Systems

While the majority of work on social navigation (including the examples reviewed above) focused on social navigation via link augmentation in virtual environments, such as hypertext, MUDs, and WWW, the early promoters of social navigation pointed out that social navigation in the real world frequently happens in the context of a natural language dialogue (Dieberger, 1997; Dieberger et al., 2000). While the research on dialogue-based social navigation received very little attention since the early days (Farrell, Rajput, Das, Danis, & Dhanesha, 2010), it could be very relevant for the area of ITSs due to the increasing popularity of conversational ITSs. ITSs with conversational dialogue form a special category of educational technologies (Rus, D'Mello, & Graesser, 2013). These conversational ITSs are based on explanation-based constructivist theories of learning and the collaborative constructive activities that occur during human tutoring. Conversational ITSs have several advantages over other types of ITSs. They encourage deep learning as students are required to explain their reasoning and reflect on their basic approach to solving a problem. Conceptual reasoning is more challenging and beneficial than mechanical application of mathematical formulas. Furthermore, conversational ITSs have the potential of giving students the opportunity to learn the language of scientists, an important goal in science literacy. A student associated with a more shallow understanding of a science topic uses more informal language as opposed to more scientific accounts (Mohan, Chen, & Anderson, 2009). The impact of conversational ITSs allegedly can be augmented by the use of social elements such as the OSSM as well as dialogue-based social navigation components. For instance, we conjecture that student engagement will increase in conversational ITSs if open learner models and open social student models are added. We are currently working on an NSF-sponsored project that will study the impact of adding open learner models and social navigation elements to the DeepTutor conversational ITS.

Summary, Recommendations, and Future Research

In this chapter, we introduced social navigation technology in the context of online education systems. Social navigation offers an alternative approach for using large volume of past learners' data for developing self-improving intelligent learning systems. While the majority of self-improving ITS work focuses on improving components or the whole system, here we argued that improvements may come from exploiting the "user community wisdom" which results in improved domain models, student modeling, and personalization algorithms. Indeed, the social navigation approach provides an example of using the "wisdom of the crowds" for empowering humans' own intelligence through a more powerful and intelligent interface. A specific goal of social navigation among other interface-focused intelligent interfaces is to help users in finding the most appropriate learning content among multiple options usually available in an online learning system. As the data of our studies shows, the presence of social navigation considerably influences students' navigation behavior, successfully guiding them to the most useful content. In turn, it positively affects student learning performance. By integrating social navigation with OSSM, the value of social interfaces could be further expanded. As the studies show, the most important impact of the OSSM interface with social comparison is an impressive increase of student engagement and retention, which makes OSSM very attractive for contexts where motivation and retention are critical, such as modern MOOCs. The literature on self-regulated learning indicates that the individual and social student models could have an even more significant positive impact on student learning in a self-regulated context. Exploring this direction, we already demonstrated that OSSM interfaces considerably improve student ability to assess their performance in both the absolute and relative sense (Somyurek & Brusilovsky, 2015). However, more extensive longer-term studies are required to assess these effects.

Taken together, our experience and findings provide important insights on the impact of social navigation and OSSM. The positive nature of the observed changes, and the magnitude of this impact demonstrated in several studies, encourages us to recommend social navigation in general and a MasteryGrids-style integration of social navigation and social comparison interfaces to the developers of various kinds of educational systems. In particular, this recommendation is for educational systems focused on more mature learners, self-regulated learning context, and non-mandatory practice learning content. Specifically, focusing on ITSs and the Generalized Intelligent Framework for Tutoring (GIFT; Sottilare, Brawner, Goldberg, & Holden, 2012), we recommend replacing a "hard sequencing" interface in this category of systems, which dictates which problems should be practiced at any moment, with an opportunity to choose the most relevant problem guided by both knowledge-driven intelligent guidance (adapted to students' own knowledge level) and social navigation guidance (adapted to a community of comparable peer learners). This interface will engage both artificial intelligence of the ITS and the natural intelligence of its human users. Multiple studies demonstrated that this kind of navigation support is more efficient and more attractive for mature learners than traditional "sequencing". Moreover, when student engagement or support for self-regulated learning are important, we recommend to use an interface, which integrates social navigation, open student models, and social comparison – as suggested in the Progressor and MasteryGrids interfaces. As our studies show, it could result in a considerable increase of student engagement and better support of self-regulated learning.

In our current and future studies, we plan to further explore the value of social navigation and open student models in different learning contexts. One direction of our research is focused on exploring a value of granularity in open student models and OSSM models. While the projects reviewed in this paper use relatively coarse-grain "topic-level" student models, we are now running a sequence of studies to explore the value of fine-grain "concept-level" models (Barria-Pineda, Guerra-Hollstein, & Brusilovsky, 2018). We also continue a stream of research, which explores the value of these technologies in contexts where learning content is specifically structured in a non-linear form, such as in hierarchical textbooks (Guerra, Parra, & Brusilovsky, 2013). We hope that our future work will bring more insights on the value of "social visdom" for improving online learning and help in developing more efficient systems.

We also plan to explore content dynamics in online learning systems with social navigation capabilities and investigate how content dynamics (adding new learning objects, deleting some, modifying some) impacts its performance. For instance, when you add a new problem, i.e., learning object, it might take a while for students to explore it and therefore accumulate sufficient "community wisdom". In extreme cases, when you add a new item to an established pool of items in a system with social navigation, the new item will likely be obscured by the previous successful items. In other words, unless the platform pushes students somehow to work with new items, the new item will never have a chance to compete with existing items which will be recommended to the users. It is a typical characteristic of a social network to make "the rich richer", i.e., a socially "rich" learning object will get "richer" by the very nature of the social navigation mechanism. The system should have ways to balance exploitation versus exploration of new, recently added objects and give the chance of “newcomers” to become visible in case they are truly valuable for, in this case, learning.

References

Barria-Pineda, J., Guerra-Hollstein, J., & Brusilovsky, P. (2018). A Fine-Grained Open Learner Model for an Introductory Programming Course. Paper presented at the 26th Conference on User Modeling, Adaptation and Personalization, UMAP '18, Singapore.

Brusilovsky, P. (2007). Adaptive navigation support. In P. Brusilovsky, A. Kobsa, & W. Neidl (Eds.), The Adaptive Web: Methods and Strategies of Web Personalization (Vol. 4321, pp. 263-290). Berlin Heidelberg New York: Springer-Verlag.

Brusilovsky, P., Chavan, G., & Farzan, R. (2004). Social adaptive navigation support for open corpus electronic textbooks. Paper presented at the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2004), Eindhoven, the Netherlands.

Brusilovsky, P., & He, D. (2018). Introduction to Social Information Access. In P. Brusilovsky & D. He (Eds.), Social Information Access: Systems and Technologies (pp. 1-18). Cham: Springer.

Brusilovsky, P., Hsiao, I.-H., & Folajimi, Y. (2011). QuizMap: Open Social Student Modeling and Adaptive Navigation Support with TreeMaps. Paper presented at the 6th European Conference on Technology Enhanced Learning (ECTEL 2011), Palermo, Italy.

Brusilovsky, P., Malmi, L., Hosseini, R., Guerra, J., Sirkiä, T., & Pollari-Malmi, K. (2018). An integrated practice system for learning programming in Python: design and evaluation. Research and Practice in Technology Enhanced Learning, 13(18), 18.11-18.40. Retrieved from https://doi.org/10.1186/s41039-018-0085-9. doi:10.1186/s41039-018-0085-9

Brusilovsky, P., & Rizzo, R. (2002). Map-based horizontal navigation in educational hypertext. Journal of Digital Information, 3(1), http://jodi.ecs.soton.ac.uk/Articles/v03/i01/Brusilovsky/. Retrieved from http://jodi.ecs.soton.ac.uk/Articles/v03/i01/Brusilovsky/.

Brusilovsky, P., Somyurek, S., Guerra, J., Hosseini, R., Zadorozhny, V., & Durlach, P. (2016). Open social student modeling for personalized learning. *IEEE Transactions on Emerging Topics in Computing, 4*(3), 450-461. Retrieved from http://doi.ieeecomputersociety.org/10.1109/TETC.2015.2501243.

Bull, S., & Kay, J. (2007). Student models that invite the learner in: The SMILI: Open learner modelling framework. *International Journal of Artificial Intelligence in Education, 17*(2), 89-120. Retrieved from http://portal.acm.org/citation.cfm?id=1435371. doi:citeulike-article-id:12888620

Bull, S., & Kay, J. (2013). Open Learner Models as Drivers for Metacognitive Processes. In R. Azevedo & V. Aleven (Eds.), International Handbook of Metacognition and Learning Technologies (pp. 349-365). Berlin: Springer.

Buunk, A. P., & Gibbons, F. X. (2007). Social comparison: The end of a theory and the emergence of a field. *Organizational Behavior and Human Decision Processes, 102*(1), 3-21. Retrieved from http://www.sciencedirect.com/science/article/pii/S074959780600094X.

Cialdini, R. B., Wosinska, W., Barrett, D. W., Butner, J., & Gornik-Durose, M. (1999). Compliance with a request in two cultures: The differential influence of social proof and commitment/consistency on collectivists and individualists. *Personality and Social Psychology Bulletin, 25*(10), 1242-1253.

Dieberger, A. (1997). Supporting social navigation on the World Wide Web. *International Journal of Human-Computer Interaction, 46*, 805-825.

Dieberger, A., Dourish, P., Höök, K., Resnick, P., & Wexelblat, A. (2000). Social navigation: Techniques for building more usable systems. *Interactions, 7*(6), 36-45.

Dourish, P., & Chalmers, M. (1994, August 1994). Running out of space: Models of information navigation. Paper presented at the People and computers IX : Proceedings of the Ninth Conference of the British Computer Society Human Computer Interaction Specialist Group, HCI'94, Glasgow.

Dron, J., Boyne, C., Mitchell, R., & Siviter, P. (2000). CoFIND: steps towards a self-organising learning environment. Paper presented at the WebNet'2000, World Conference of the WWW and Internet, San Antonio, TX.

Farrell, R., Rajput, N., Das, R., Danis, C., & Dhanesha, K. (2010). Social Navigation for the Spoken Web. Paper presented at the 2010 ACM conference on Recommender systems, RecSys '10, Barcelona, Spain.

Farzan, R., & Brusilovsky, P. (2005a). Social navigation support in E-Learning: What are real footprints. Paper presented at the IJCAI'05 Workshop on Intelligent Techniques for Web Personalization, Edinburgh, U.K.

Farzan, R., & Brusilovsky, P. (2005b). Social navigation support through annotation-based group modeling. Paper presented at the 10th International User Modeling Conference, Berlin.

Farzan, R., & Brusilovsky, P. (2008). AnnotatEd: A social navigation and annotation service for web-based educational resources. New Review in Hypermedia and Multimedia, 14(1), 3-32. Retrieved from http://dx.doi.org/10.1080/13614560802357172

Farzan, R., & Brusilovsky, P. (2018). Social Navigation. In P. Brusilovsky & D. He (Eds.), Social Information Access: Systems and Technologies (pp. 142-180). Cham: Springer.

Farzan, R., Coyle, M., Freyne, J., Brusilovsky, P., & Smyth, B. (2007). ASSIST: Adaptive social support for information space traversal. Paper presented at the 18th conference on Hypertext and hypermedia, HT '07, Manchester, UK.

Festinger, L. (1954). A theory of social comparison processes. *Human Relations, 7*(2), 117-140.

Guerra, J., Hosseini, R., Somyurek, S., & Brusilovsky, P. (2016). An Intelligent Interface for Learning Content: Combining an Open Learner Model and Social Comparison to Support Self-Regulated Learning and Engagement. Paper presented at the the 21st International Conference on Intelligent User Interfaces (IUI '16), Sonoma, California, USA.

Guerra, J., Parra, D., & Brusilovsky, P. (2013). Encouraging Online Student Reading with Social Visualization. Paper presented at the 2nd Workshop on Intelligent Support for Learning in Groups at the 16th Annual Conference on Artificial Intelligence in Education, AIED 2013, Memphis, TN, USA.

Hill, W. C., Hollan, J. D., Wroblewski, D., & McCandless, T. (1992). Edit wear and read wear. Paper presented at the SIGCHI Conference on Human Factors in Computing Systems, CHI'92.

Höök, K., Benyon, D., & Munro, A. J. (Eds.). (2003). Designing Information Spaces: The Social Navigation Approach. Berlin: Springer

Hsiao, I.-H., Bakalov, F., Brusilovsky, P., & König-Ries, B. (2011). Open Social Student Modeling: Visualizing Student Models with Parallel Introspective Views. Paper presented at the 19th International Conference on User Modeling, Adaptation, and Personalization, UMAP 2011, Girona, Spain.

Hsiao, I.-H., Bakalov, F., Brusilovsky, P., & König-Ries, B. (2013). Progressor: social navigation support through open social student modeling. New Review of Hypermedia and Multimedia, 19(2), 112-131. Retrieved from http://dx.doi.org/10.1080/13614568.2013.806960. doi:10.1080/13614568.2013.806960

Hsiao, I.-H., & Brusilovsky, P. (2017). Guiding and motivating students through open social student modeling: Lessons learned. *Teachers College Record, 119*(3). Retrieved from http://www.tcrecord.org/Content.asp?ContentId=21773.

Hsiao, I.-H., Sosnovsky, S., & Brusilovsky, P. (2010). Guiding students to the right questions: adaptive navigation support in an E-Learning system for Java programming. *Journal of Computer Assisted Learning, 26*(4), 270-283. Retrieved from http://dx.doi.org/10.1111/j.1365-2729.2010.00365.x doi:10.1111/j.1365-2729.2010.00365.x

Hübscher, R., & Puntambekar, S. (2004). Modeling learners as individuals and as groups. Paper presented at the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2004), Eindhoven, the Netherlands.

Khan, B. H. (Ed.) (1997). Web Based Instruction. Englewood Cliffs, New Jersey: Educational Technology Publications.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education, 8*, 30-43.

Kohonen, T. (1995). Self-organizing maps. Berlin: Springer Verlag.

Kurhila, J., Miettinen, M., Nokelainen, P., & Tirri, H. (2002). EDUCO - A collaborative learning environment based on social navigation. Paper presented at the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002), Málaga, Spain.

Light, P., Littleton, K., Bale, S., Joiner, R., & Messer, D. (2000). Gender and social comparison effects in computer-based problem solving. *Learning and Instruction, 10*(6), 483-496. Retrieved from https://www.sciencedirect.com/science/article/pii/S0959475200000104. doi:https://doi.org/10.1016/S0959-4752(00)00010-4

Loboda, T., Guerra, J., Hosseini, R., & Brusilovsky, P. (2014). Mastery Grids: An Open Source Social Educational Progress Visualization. Paper presented at the 9th European Conference on Technology Enhanced Learning (EC-TEL 2014), Graz, Austria.

Martin, B., Mitrovic, A., Koedinger, K., & Mathan, S. (2011). Evaluating and improving adaptive educational systems with learning curves. *User Modeling and User-Adapted Interaction, 21*(3), 249-283. Retrieved from http://dx.doi.org/10.1007/s11257-010-9084-2.

Mitsuhara, H., Kanenishi, K., & Yano, Y. (2004). Learning Process Sharing for Educational Modification of the Web. Paper presented at the ED-MEDIA'2004 - World Conference on Educational Multimedia, Hypermedia and Telecommunications, Lugano, Switzerland.

Mohan, L., Chen, J., & Anderson, C. W. (2009). Developing a multi-year learning progression for carbon cycling in socio-ecological systems. *Journal of Research in Science Teaching, 46*, 675–698.

Munro, A. J., Höök, K., & Benyon, D. (Eds.). (1999). Social Navigation of Information Space. Berlin: Springer

O'Shea, T. (1982). A self-improving quadratic tutor. *Int. Journal on the Man-Machine Studies, 11*, 97-124.

Pavlik, P. I., Jr., Cen, H., Wu, L., & Koedinger, K. R. (2008). Using item-type performance covariance to improve the skill model of an existing tutor. Paper presented at the the 1st International Conference on Educational Data Mining, Montreal, Canada.

Pintrich, P. R., & De Groot, E. V. (1990). Motivational and self-regulated learning components of classroom academic performance. *Journal of Educational Psychology, 82*(1), 33-40.

Rus, V., D'Mello, S., & Graesser, A. (2013). Recent advances in conversational intelligent tutoring systems. *AI Magazine, 34*(3), 42-54.

Sleeman, D., & Brown, J. S. (1982). Intelligent tutoring systems. In D. H. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 1-12). London: Academic Press.

Somyurek, S., & Brusilovsky, P. (2015). Impact of Open Social Student Modeling on Self-Assessment of Performance. Paper presented at the E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, Kona, HI, USA.

Sosnovsky, S., & Brusilovsky, P. (2015). Evaluation of topic-based adaptation and student modeling in QuizGuide. *User Modeling and User-Adapted Interaction, 25*(4), 371-424. Retrieved from http://link.springer.com/article/10.1007/s11257-015-9164-4 doi:10.1007/s11257-015-9164-4

Sottilare, R.A., Brawner, K.W., Goldberg, B.S., & Holden, H.K. (2012). *The Generalized Intelligent Framework for Tutoring (GIFT)*. Downloaded from www.gifttutoring.org on November 30, 2012.

Tattersall, C., Berg, B. v. d., Es, R. v., Janssen, J., Manderveldt, J., & Koper, R. (2004). Swarm-based adaptation: Wayfinding support for lifelong learners. Paper presented at the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH 2004, Berlin.

Vassileva, J. (2004). Harnessing P2P Power in the Classroom. In J. C. Lester, R. M. Vicari, & F. Paraguaçu (Eds.), Intelligent Tutoring Systems: 7th International Conference, ITS 2004 (pp. 305-314). Berlin, Heidelberg: Springer.

Wexelblat, A., & Mayes, P. (1999). Footprints: History-rich tools for information foraging. Paper presented at the ACM Conference on Human-Computer Interaction (CHI'99), Pittsburgh, PA.

Zimmerman, B. J. (1990). Self-Regulated learning and academic achievement: An overview. *Educational Psychologist, 25*(1), 3-17. Retrieved from https://doi.org/10.1207/s15326985ep2501_2

Chapter 16 – Reports To Facilitate Improvements of Adaptive Instructional Systems

Andrew J. Hampton[1] and Ryan S. Baker[2]
[1]University of Memphis, [2]University of Pennsylvania

Introduction

Many technologically groundbreaking methods have been proposed for creating self-improving adaptive instructional systems (AISs). Where these proposals primarily embrace innovative uses of advanced technical capabilities, we emphasize maintaining contact with a base from which systems should improve themselves: consideration of how humans perform similar tasks. In many of these systems, developers have processes in place to receive meaningful and interpretable data on various aspects of learner performance in order to understand system quality and learner outcomes. Providing accurate, relevant system information in a comprehensible, usable fashion constitutes a necessary precondition to any reliably and demonstrably self-improving (or in fact, manually improving) AISs. In this chapter, we consider how instructional systems are currently adapted, and propose some ideas for how analogous improvement approaches can be implemented within self-improving systems.

The concept of a self-improving system presupposes that (a) quality is measurable and (b) measurements of quality can inform changes meant to improve those measurements. AIS practitioners continue to explore creative ways to produce autonomously controlled enhancements based on these two pillars, but we must carefully verify that each of these two preconditions is met. That is, a self-improving system may generate improvements by identifying and adjusting sub-optimal interactions with learners, but without an effective way to measure learner outcomes, the changes may be arbitrary or even counterproductive. This could happen for several reasons. Changes may be ineffective because of poor measurement. Changes may be effective but the system is unable to convince an outside practitioner that this is the case. Systems may become trapped in local maxima, unable to reason out of a condition in which parameters of interest (but not actual performance) appear optimally tuned. The system may recognize a problem but not have the operators to address it. In other words, a self-improving system may not always be able to improve itself. Sometimes the only way to solve a problem may be to present relevant, digestible information to humans and enlist them to fix the flaw.

While simple in principle, selection and presentation of information provide an array of challenges. An AIS, by definition, has complex interaction protocols. Learner input changes the system output, using complex formulations of performance to produce truly unique learner profiles. This expansive array of possible paths impedes a priori categorization of learners into discrete data categories. Further, systems typically contain metrics for performance on multiple facets of a topic rather than a holistic score. Layers of multiplicative complexity stem from unstandardized inclusions of psychological variables, coordination with external instruction, or consideration of learners collectively (e.g., how a class performs, and a single learner in relation to that class). Identifying the critical variables requires careful consideration and often sophisticated data analysis. Representing the resulting data may vary drastically depending on their type (individual versus collective, qualitative versus quantitative, etc.) and rate of change. Exceptionally complicated concepts and relationships need to be conveyed efficiently, with primary metrics easily perceptible and the option to dig deeper readily available.

Dashboards provide the primary method of conveying these types of information to practitioners. Understanding the ways in which humans use the dashboard data fundamentally impacts dashboard design. Learning analytics researchers have established methods for developing dashboards for use by instructors and system designers. A careful review of these research traditions with an eye toward cultivating, organizing, and presenting AIS variables to practitioners will yield actionable recommendations for designers.

Designing for Users

“Know thy user”, a common mantra among system designers, reminds us that there exists a typical set of use cases, and that those use cases should be made as easy and efficient as possible. In the context of an AIS, the most critical variables typically consist of metrics related to learner performance and quality of instructional content. These metrics then form the basis of changes made to the system or curricula to improve learner outcomes. “Know thy user” thus suggests that designers present information in such a way as to match the existing mental models of instructors and practitioners. With this match established, common evaluations or procedures proceed as seamlessly as possible.

However, it can in some cases be challenging to develop reporting and dashboards that support teachers in conducting data-driven decision-making (Marsh & Farrell, 2015). As a result, many dashboards are used unevenly across teachers (Hawn, 2019). This relates broadly to the challenge of providing professional development training to teachers that encourages individualized instruction (Al Otaiba, Connor, Folsom, Greulich, Meadows, & Li, 2011). Principled dashboard design is unlikely to fix this problem on its own, but certainly constitutes a step in the direction of broader adoption, both of dashboards and of the effective instructional practices that they can promote.

The conventional human–computer interaction approach of matching the presentation of information to existing mental models assumes appropriate mental models for users (Flach & Voorhorst, 2016) who may have limited experience in pedagogical theory or educational technology. The challenge then becomes to present information in such a way as to *encourage* a mental model in line with the affordances and constraints of the system, with respect to the ultimate goal of improving learning outcomes. In this way, information can directly suggest action by encouraging an intuitive understanding of the relationship between intervention and outcome, between intentional cause and desired effect.

For example, data representations of individual instructional items should immediately suggest to instructors which items fail to challenge learners (ceiling effect) or serve only to frustrate and confuse (floor effect). Areas ripe for improvement should be accentuated, and methods of intervention intuitively suggested. Dashboard design provides a direct opportunity to influence mental models toward appropriate action.

Applications of Dashboards in Improvement of Instruction

Learning analytics dashboards provide potentially useful resources for many stakeholders attempting to improve the performance of AISs. This includes developers working *on* the systems, teachers working *in* the systems, and school personnel working *with* the systems as part of their broader instructional approach. Many AISs have been designed for and used by school and university personnel, including teachers, instructors, academic advisors, school counselors, and higher-level administrators. Though most dashboards are presented through traditional computer screen displays, recent research has considered if other modalities may be better for providing reports to classroom teachers, including wall-based ambient displays (Alphen & Bakker, 2016) or augmented reality headsets (Holstein, McLaren, & Aleven, 2018).

Perhaps the most common application for dashboards in the learning analytics space is in communicating data on learner student performance to instructors. These types of dashboards have become commonplace within K–12 adaptive learning systems and were present from the first large-scale usage of AISs. However, their inclusion was initially poorly documented. For example, a "teacher's toolkit" with reports on student performance has been part of the *Cognitive Tutor* system and used in schools since the mid-1990s but is not discussed in detail in publications regarding that system. Teacher reports were considerably better documented in the later *ASSISTments* system, with an article discussing the range of reports offered by the system as well as their use by teachers and school leaders (Feng & Heffernan, 2006). Both *Cognitive Tutor* and *ASSISTments* are relatively straightforward mathematics problem-solving environments. This type of dashboard has been incorporated in recent years by other types of learning systems such as exploratory learning environments (Mavrikis, Gutierrez-Santos, & Poulovassilis, 2016) and group learning environments (Martinez-Maldonado, Clayphan, Yacef, & Kay, 2015).

Another popular use for learning analytics dashboards is in risk prediction—typically dropout or course failure. Perhaps the most widely used platform for predicting which students are at-risk in higher education is the *Civitas* platform, used by universities worldwide and providing a range of reports about students, leveraging data on course-taking, grades, admissions data, and questionnaires administered to students (Milliron, Malcolm, & Kil, 2014). The *Course Signals* platform, perhaps the first widely used platform in this space, integrates dashboard reports with recommendations and scaffolding for email-based interventions by instructors (Arnold & Pistilli, 2012); when an instructor clicks on an at-risk student within the dashboard, the platform goes straight to a recommended action and scaffolded intervention tailored to that specific learner, which the instructor has the option to modify before sending to the learner. At the K–12 level, *BrightBytes*, used by dozens of school districts nationwide, offers reports on the at-risk status of students (Singh, 2018). These are only a few examples of the increasing number of platforms and vendors now offering at-risk prediction dashboards to universities and school districts. There have also been efforts to create dashboards for risk prediction within the context of MOOCS (e.g. Chen et al., 2016), though adoption lags the use of dashboards in for-credit online university programs. Many other uses exist for dashboards in learning, from presenting course recommendations to students (Denley, 2013), to providing visualizations to students of their own activity and progress (Kim, Jo, & Park, 2016), to visualizing group work (Kay et al., 2006).

However, there has not yet been sufficient attention in the published literature to dashboards for the enhancement of learning content. Existing dashboards used primarily for teachers to monitor student progress can effectively serve a secondary function of evaluating content (e.g. Feng & Heffernan, 2006) or the effectiveness of the learning platform. While the design and enhancement of dashboards is often seen as a secondary goal by the developers of adaptive learning systems, we would argue that dashboards are all but indispensable to the creation of high-quality computer learning environments. Though there are efforts to create dashboards for course designers, this work has often not been published, despite its pivotal role in enhancing the instructional quality of adaptive learning systems.

How Dashboards are Used

Dashboards are used in a variety of ways that potentially impact the practice of teachers (Miller et al., 2015; Xhakaj et al., 2017) and academic advisors (Arnold & Pistilli, 2012; Lonn, Aguilar, & Teasley, 2015). Teachers frequently access dashboards during class to drive pedagogical decisions, consulting dashboards as frequently as eight times per class session (Molenaar & Knoop-van Campen, 2017), and using this information to inform pedagogical strategies. For example, many teachers use proactive remediation, where teachers speak to a student struggling with specific material (Miller et al., 2015). Other teachers use dashboards to identify students who have recently succeeded and provide them with encouragement (Molenaar

& Knoop-van Campen, 2017). Molenaar and Knoop-van Campen (2017) note that teachers using dashboards shift from providing general learning support to the most struggling students to providing targeted support to a broader range of students.

Not all dashboards are used in real-time. Indeed, within the *ASSISTments* system, one of the most common uses of dashboards is by teachers to identify content that students struggled with during homework and redesign the next class session's lesson to target specific errors common across students (Kelly, Heffernan, D'Mello, Namias, & Strain, 2013). Teachers also use *ASSISTments* dashboards to split students into groups, putting students in groups together if they make similar patterns of errors, and giving advanced content to groups of students who made no errors (Heffernan & Heffernan, 2014).

The use of dashboards connected to adaptive (or non-adaptive) instructional systems is part of a broader move towards using data in instruction, what is referred to as data-based decision making or data-driven decision making (Halverson, 2014). Teachers now commonly receive a broad range of types of data within dashboards, including data from student information systems, formative assessment systems, disciplinary data, attendance data, and teacher observation data (Mandinach & Jackson, 2012). While there has been skepticism about the effectiveness of data-based decision-making involving coarser-grained data (Mandinach & Jackson, 2012), teachers' use of dashboards involving finer-grained data (such as is generated by AISs) is associated with better student outcomes, both in K–12 (Xhakaj et al., 2017) and in higher education (Arnold & Pistilli, 2012). It is not yet entirely clear which of these uses produces the benefits associated with dashboards—but this is exactly the sort of question that could be answered by a self-improving AIS. By trying the pedagogical strategies associated with the use of data in dashboards, one-by-one, an AIS could determine which approach to data-driven instruction works for which students in which contexts.

Recommendations and Future Research

These past projects suggest some steps for taking AISs forward. First, the types of decisions captured in instructor responses to dashboards may be directly applicable to training AISs to replicate instructor behaviors. Take, for example, the *Course Signals* system discussed above. This system presents recommended interventions to instructors, but instructors do not have to follow that recommendation—they can ignore it or modify the intervention. By tracking the situations in which instructors choose to intervene and how they choose to modify interventions, an AIS may be able to more closely replicate instructor behavior.

One could envision that this could be accomplished via linking between the Generalized Intelligent Framework for Tutoring (GIFT) architecture and an external system that provides recommendations to an instructor, via the Learning Management System Module. The recommendations could be generated from the Pedagogical Module, using information from the Learner Module. In turn, the Learning Management System could be designed to track whether and how instructors follow its recommendations, and this information could be passed back to the Pedagogical Module to update its algorithm for when and how to recommend intervention.

More directly, it may be possible to create a repertoire of AIS behaviors based on the behaviors in which teachers engage with dashboards, such as the proactive remediation behaviors identified by Miller and his colleagues (2015) and the encouragement behaviors identified by Molenaar and Knoop-van Campen (2017). Again, these behaviors could be embedded into the Pedagogical Module and triggered by the Learner Module. In this case, the behaviors would actually occur within the Tutor Module rather than the Learning Management System Module. An AIS could then use reinforcement learning (running within the Pedagogical Module) to determine which situations are the most beneficial for applying these strategies.

Further study of teachers' behaviors with dashboards would likely be beneficial to research and development along these lines. For example, it may be possible to create activities similar to those used by Kelly et al. (2013), where performance within *ASSISTments* is used to drive the selection of problems to work through, providing a practice experience to one student and a worked-out erroneous example (Adams et al., 2014) to other students. Instruction modeling analysis (Khachatryan, in press) can be conducted on the practices of teachers such as Dr. Kelly to provide a foundation for creating an AIS that can replicate observed pedagogical strategies, in the case of GIFT within the Pedagogical Module.

The future of AISs is strong. However, in developing AISs that can improve themselves, it is worth considering the example of how humans already enhance existing instructional systems. By doing so, we may be able to speed the enhancement of these systems and understand which enhancements work to the ultimate benefit of the learners.

References

Adams, D. M., McLaren, B. M., Durkin, K., Mayer, R. E., Rittle-Johnson, B., Isotani, S., & Van Velsen, M. (2014). Using erroneous examples to improve mathematics learning with a web-based tutoring system. *Computers in Human Behavior*, *36*, 401–411.

Al Otaiba, S., Connor, C. M., Folsom, J. S., Greulich, L., Meadows, J., & Li, Z. (2011). Assessment data-informed guidance to individualize kindergarten reading instruction: Findings from a cluster-randomized control field trial. *The Elementary School Journal, 111*(4), 535–560.

Alphen, E. V., & Bakker, S. (2016, May). Lernanto: Using an ambient display during differentiated instruction. In J. Kaye, & A. Druin (Eds.), *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 2334–2340). San Jose, CA: ACM.

Arnold, K. E., & Pistilli, M. D. (2012). Course signals at Purdue: Using learning analytics to increase student success. In S. B. Shum, D. Gasevic, & R. Ferguson (Eds.), *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (pp. 267–270). Vancouver, Canada: ACM.

Chen, Y., Chen, Q., Zhao, M., Boyer, S., Veeramachaneni, K., & Qu, H. (2016, October). DropoutSeer: Visualizing learning patterns in Massive Open Online Courses for dropout reasoning and prediction. In G. Andrienko, S. Liu, & J. Stask0 (Eds.), *IEEE Conference on Visual Analytics Science and Technology* (pp. 111–120). New Brunswick: IEEE.

Denley, T. (2013). Degree compass: A course recommendation system. *Educause Review Online*.

Feng, M., & Heffernan, N. T. (2006). Informing teachers live about student learning: Reporting in the ASSISTment system. *Technology Instruction Cognition and Learning*, *3*(1/2), 63.

Flach, J. M., & Voorhorst, F. (2016). *What matters*. Dayton, OH: Wright State University Library.

Halverson, R. (2014). Data-driven leadership for learning in the age of accountability. In A. J. Bowers, A. R. Shoho, & B. G. Barnett (Eds.), *Using Data in Schools to Inform Leadership and Decision Making* (pp. 255–266). Charlotte, NC: Information Age Publishing Incorporated.

Hawn, M.A. (2019) *Data-wary, value-driven: Teacher attitudes, efficacy, and online access for data-based decision making* (Unpublished doctoral dissertation). Teachers College, Columbia University, New York.

Heffernan, N. T., & Heffernan, C. L. (2014). The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education, 24*(4), 470–497.

Holstein, K., McLaren, B. M., & Aleven, V. (2018). Student learning benefits of a mixed-reality teacher awareness tool in AI-enhanced classrooms. In C. P. Rosé, R. Martínez-Maldonado, H. U. Hoppe, R. Luckin, M. Mavrikis, K, Porayska-Pomsta, B. McLaren, & B. du Boulay (Eds.), *Proceedings of the International Conference on Artificial Intelligence in Education* (pp. 154–168). London, England, UK.

Kay, J., Maisonneuve, N., Yacef, K., & Reimann, P. (2006). The big five and visualisations of team work activity. In M. Ikeda, K. D. Ashley, & T. W. Chan (Eds.), *International Conference on Intelligent Tutoring Systems* (pp. 197–206). Berlin, Heidelberg: Springer.

Kelly, K., Heffernan, N., D'Mello, S., Namias, J., & Strain, A. (2013). Adding teacher-created motivational video to an ITS. In C. Boonthum-Denecke, & G. M. Youngblood (Eds.), *Proceedings of the Twenty-Sixth Florida Artificial Intelligence Research Society Conference*. pp. 503–508. St. Pete Beach, FL: AAAI Press.

Khachatryan, G. (in press) *Instruction modeling and blended learning.* Cambridge, MA: Oxford University Press.

Kim, J., Jo, I. H., & Park, Y. (2016). Effects of learning analytics dashboard: analyzing the relations among dashboard utilization, satisfaction, and learning achievement. *Asia Pacific Education Review*, *17*(1), 13–24.

Lonn, S., Aguilar, S. J., & Teasley, S. D. (2015). Investigating student motivation in the context of a learning analytics intervention during a summer bridge program. *Computers in Human Behavior*, *47*, 90–97.

Mandinach, E. B., & Jackson, S. S. (2012). *Transforming teaching and learning through datadriven decision making*. Thousand Oaks, CA: Corwin Press.

Marsh, J. A., & Farrell, C. C. (2015). How leaders can support teachers with data-driven decision making: A framework for understanding capacity building. *Educational Management Administration & Leadership*, *43*(2), 269–289.

Martinez-Maldonado, R., Clayphan, A., Yacef, K., & Kay, J. (2015). MTFeedback: Providing notifications to enhance teacher awareness of small group work in the classroom. *IEEE Transactions on Learning Technology, 8*, 2. pp. 187–200.

Mavrikis, M., Gutierrez-Santos, S., & Poulovassilis, A. (2016). Design and evaluation of teacher assistance tools for exploratory learning environments. In D. Gašević, & G. Lynch (Eds.), *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge* (pp. 168–172). Edinburgh, Scotland: ACM.

Miller, W.L., Baker, R., Labrum, M., Petsche, K., Liu, Y-H., Wagner, A. (2015) Automated detection of proactive remediation by teachers in reasoning mind classrooms. In J. Baron, G. Lynch, & N. Maziarz (Eds.), *Proceedings of the 5th International Learning Analytics and Knowledge Conference*, 290–294. Poughkeepsie, New York: ACM.

Milliron, M. D., Malcolm, L., & Kil, D. (2014). Insight and action analytics: Three case studies to consider. *Research & Practice in Assessment*, *9*, 70–89.

Molenaar, I., & Knoop-van Campen, C. (2017). Teacher dashboards in practice: Usage and impact. In *European Conference on Technology-Enhanced Learning (EC-TEL)* (pp. 125–138). Tallinn, Estonia: Springer.

Singh, R. P. (2018). Learning Analytics: Potential, Protection, and Privacy in the Educational System. In M. K. Singh, Z. Zerihun, & N. Singh (Eds.), *Impact of Learning Analytics on Curriculum Design and Student Performance* (pp. 1–18). Hershey, PA: IGI Global.

Xhakaj, F., Aleven, V., & McLaren, B. M. (2017). Effects of a teacher dashboard for an intelligent tutoring system on teacher knowledge, lesson planning, lessons and student learning. In E. Lavoué, H. Drachsler, K. Verbert, J. Broisin, & M. Pérez-Sanagustín (Eds.), *European Conference on Technology-Enhanced Learning* (pp. 315–329). Tallinn, Estonia: Springer.

Chapter 17 – Social Media Frameworks and Improved Learning Engagement in GIFT

Niki Gitinabard[1], Collin F. Lynch[1], Rodney Long[2], Christopher Meyer[3] & Lucy Woodman[3]
[1]North Carolina State University, [2]U.S. Army Combat Capabilities Development Command (CCDC) - Soldier Center – Simulation and Training Technology Center (STTC), [3]Synaptic Sparks, Inc.

Introduction

Learning is a social activity. Students move through educational programs in cohorts of like-minded or similarly-aged peers. Within classes they develop formal and informal study groups, make friends, and wrestle with competition. Students will also leverage these social connections for class and career advice, informal tutoring, collaborative learning, and cheating. As prior research has shown, the presence or absence of a trusted peer can make the difference between passing and failing, or between pressing on and dropping out (Chuateco, Dennis, & Phinney, 2005). Despite this, many online learning platforms are designed primarily for individuals and make few attempts to incorporate key social relationships. As a consequence, they often miss key mechanisms for providing student support and critical information that can be used to guide and to retain future learners. In this chapter we will survey the use of explicit social network systems in educational platforms and the analysis of social network data to support students' learning. We will also highlight how third-party tools and social features have been integrated into online learning platforms and the impacts, if any, that they have had. We will conclude with recommendations for the tools in the Generalized Intelligent Framework for Tutoring (GIFT) software suite.

As Lave and Wenger (1991) argued in their influential work, learning is about social participation. Organizational groups of all types from professional societies to academic organizations, to families, form *communities of practice* (Wenger, 1998). *Communities of practice* are groups of individual learners and practitioners that are united by a shared identity as students, researchers, dentists, or claims adjusters. These communities act as repositories of organizational knowledge from best practices, to knowing which class is easy. They help to train new members by engaging them in appropriate activities through *legitimate peripheral participation* (Lave & Wenger, 1991), by providing opportunities for peer support and for learning via the co-construction of new knowledge (Heo, Kim, & Lim, 2010), as well as providing exemplars for new learners to emulate (Dabbish, Kraut, & Seering, 2017).

The importance of peer groups has been established by a number of empirical researchers. Chuateco et al. (2005), for example, showed that peer support in college can help students adjust better and lack of peer support is a predictor of lower GPA. Similarly, better social connections have also been linked to better performance (Dawson, 2010) and to early participation in linked learning communities (Settle & Steinbach, 2016) where students simultaneously enroll in classes to form what we term *conscious cohorts*. Settle and Steinbach (2016) and Carlson et al. (2014) also showed that these communities reduce students' feelings of isolation and thus improve their course retention. Similar conclusions have also been advanced by Blom et al., (2014) as well as Eckles and Stradley (2012) who found that individual isolation, or limitation to an isolated subgroup can increase the likelihood of dropouts.

The development of good community structures is thus essential to education, and is all the more challenging in online settings. In face-to-face classrooms, students are furnished with an immediate, and accessible peer group. While they may not like their classmates, it is clear whom those classmates are, and how they may be reached. In online settings, students have no set of obvious compatriots. While their classmates may share personal backgrounds, professional goals, and educational challenges, they are often separated by distances, time zones, and even cultures and may be limited to communicating with one-another through asynchronous forums, or limited chats. Thus, even in a class of thousands, a student may be very much alone.

Social Platforms and Social Networks

As online learning has become more commonplace, and as traditional class sizes have grown to 30, 50 or even hundreds, the need for alternative methods to foster and maintain social connections among students has grown as well. As Burns, Light, Light, and Nesbitt (2010, pp. 85) noted, "Many tutors now have to deal with large tutorial groups and are coming to use computer-based resources as both an aid and supplement to face-to-face meetings. Tutors are placing course materials on the worldwide web and establishing e-mail and conferencing systems as a basis of communication between themselves and their students. Some tutors are also setting up online course discussions to run alongside face-to-face tutorials."

One avenue that many researchers and educators have taken is to incorporate platforms such as Facebook which support explicit and implicit social networks. Online social networks, while previously uncommon have become a basic means of communication for many. As noted even back in 2010, Web 2.0 technologies have enabled students, both in secondary and university level education, to connect and socialize with regards to learning in undeniably impactful ways (Eklund & VanDoorn, 2013). Commercial platforms like Facebook have made communication and collaboration fast and easy; this creates a shared sense of "social connectedness" (Eklund & VanDoorn, 2013) that can in turn reflect and sustain both physical communities with shared personal connections (such as family), or virtual communities of like-minded individuals with shared interests. The value added from the incorporation of social media tools into instructional tutoring software suites such as GIFT is very attractive, especially in online-only courses. Increasing users' ability to collaborate is not just limited to online courses. Implementing certain elements of social media into traditional, face-to-face courses has also proven to be quite helpful. Ball, Desbrow, Irwin, and Leveritt (2012), for example, studied classroom use of Facebook and noted that "The [Facebook] page enhanced communication and interaction between students and the course instructors, interaction with the Facebook page was easy as students were commonly using Facebook for social networking, students were able to receive updates and information which may have been missed via other communication means," and "…[the] response to questions and facilitation of discussions were faster than relying on email and discussion boards" (Ball et al., 2012, pp. 1227). Bohley et al. (2009) also previously put forward a conceptual framework for the use of tools such as Facebook to support the formation and maintenance of communities of practice.

The topics of social platforms and social networks were of particular interest to the GIFT team in 2017 due to the potential applications in improving intelligent tutoring systems. The team performed a trade study of available and open-source social platform software suites, and then began to perform feasibility studies on potential integrations with the GIFT software suite. At the conclusion of the studies, it was determined that integrating GIFT with the social networking engine software framework called Elgg (located on the world wide web at https://elgg.org) would best serve the community and the GIFT research team.

When seeking specific capabilities that add the most value to online learning platforms like GIFT, the research and development teams analyzed subsets of popular platforms such as Facebook and Blackboard to see which capabilities were highest in value and most-feasible to implement in the GIFT software suite for future experiments. Following the advice of researchers like Lee and McLoughlin, "Social software tools can be effectively integrated into both face-to-face and online environments; the most promising settings for a pedagogy that capitalizes on the capabilities of these tools are fully online or blended so that students can engage with peers, instructors, and the community in creating and sharing ideas" (Lee, McLoughlin, 2008, pp. 2). Initial capabilities deemed to have high value-to-feasibility ratings were tools such as User-to-User Online Chat, Discussion Boards, File Sharing, Course Groups, and concepts of Curriculum Ownership to automatically create networked groups of students and instructors based on courses that users owned and shared in GIFT. All of these capabilities also happened to be included with Elgg or reasonably easy to implement on a software level.

Allowing instructors to quickly, efficiently, accurately, and sometimes automatically grade learners' performance metrics was a final area of postulated improvement. A software package such as Blackboard aimed to "expand access to education" (Blackboard, 2019), and accomplishes this by allowing instructors to "type or upload their course materials into Blackboard" (Lewis, MacEntee, & Youngs-Maher, 2002, pp. 921) and then allowing them to manage and grade students' work. According to one study, a significant benefit of Blackboard is that the quizzes save both student and instructor time, thanks to their electronic nature and self-correcting feature (Fritz, 2003). This style of automated assessment is largely present in GIFT, however, new proposed tools may be integrated with learning management systems, instructor dashboards, and/or social networking software to great effect.

In addition to scaffolding students' engagement and course management the use of social networking platforms also provides new opportunities for data analysis. Educational researchers have long studied students' social networks. For example, Elovici, Fire, Katz, Rokach, and Shapira (2012) and Barnes and colleagues (Barnes, Gitinabard, Heckman, Lynch, & Xue, 2017a; Gitinabard, Khoshnevisan, Lynch, & Wang, 2018) showed direct correlations between students' social connections and their retention in both traditional face-to-face and online courses respectively. Other researchers have drawn similar conclusions by analyzing social networks created from students' online forum use to evaluate their grades (e.g. Oleksandra & Shane, 2016; Dawson et al., 2018), feelings of social connection (e.g. Dawson et al., 2018), and completion rate (Choi, Jablokow, Pursel, Velegol, & Zhang, 2016; Andres et al., 2018; Chen & Zhang 2017; Carlson et al., 2014; Adamson, Rosé, Sinha, & Yang, 2013). Prior researchers have also identified the formation of distinct subcommunities within courses (Dawson, 2010; Albert et al., 2015; Barnes et al., 2017a) and the impact of those communities on students' grades.

This analysis has historically been based either upon the analysis of *explicit* social relationships which are gathered from students via surveys or other direct questionnaires (e.g. Elovici et al., 2012; Farmer & Rodkin,1996; De La Fuente, Dimitriadis, Gómez, Martınez, & Rubia, 2003). Or the analysis of *implicit* social relationships that are estimated from students' participation activities in online forums (e.g. Carlson et al. 2014; Barnes et al., 2017b), or in rarer cases the collection of phone calls and text messages (Lei, Sinha, & Wang, 2015). In the latter case there is often little ground truth for analysis. In one rare study, Capelleri, Peserico, and Samory (2017) analyzed a setting where both explicit and implicit networks were available and did find a high degree of duplication between them. Once collected, this data can be analyzed through standard tools such as Meerkat-ED (Rabbany, Takaffoli, Zaïane, 2011).

Thus by incorporating a social learning platform into the GIFT system, we can vastly enhance its utility for students and instructors by providing avenues for the development and maintenance of communities of practice among the learners. We can also provide opportunities for direct peer support and intervention. Additionally, we can collect social relationship data that can be analyzed to identify students who need support, identify potential peers or problems, and to enhance and even tailor instruction to meet students' needs through data-driven recommender systems. That said, the resulting social networks will also contain a great deal of personal information that, if abused, could allow for excess monitoring of students' social relationships either within or across classes or allow information about individual skills or training to be unintentionally revealed. Therefore, rigid controls and security measures would need to be put in place to secure this student data, and strong procedural guarantees would need to be made to prevent abuse.

Evaluation & Results

In light of this strong research, the GIFT team believed that there is a high value to be provided to GIFT users from combining social media with pre-existing educational GIFT courses and experiments based off of modern instructional management tools. A private GIFT server was setup for the testing of new social media capabilities added to the GIFT software suite with the Elgg framework. Primary capabilities were specifically implemented for social media framework experimentation purposes that are not currently present in the deployed public version of GIFT. These include:

Automated Login to GIFT Social Media Framework Database

After a user had created an account at GIFTtutoring.org and logged in to the social media framework version of GIFT, the user was also added to an isolated social media framework database. This allowed for full experimentation with the user's data to occur without impacting any other online version of GIFT they may use in the future.

Course Association Groups and Ownership Rights

Every user in the GIFT Social Media Framework is able to create their own (and multiple) curriculums, courses, and groups. As such, when a course was created and owned in GIFT, a corresponding social media group was created with the GIFT author as exclusive owner. The author was then able to share, invite, and group other GIFT Social Media Framework users as they saw fit into "class groups" for discussions, file sharing, and chatting.

Friend Invites, Private Messages, and Feeds

Additionally, every GIFT social media user was able to ask other GIFT users for their ID, invite them to be friends, and upon confirmation share in each other's public account information. This information included profile info, current status, what users liked/disliked, commented on, and group memberships. Once a handshake of friend confirmation occurred, each user was also able to message the other and see an increased activity feed based on customizable filters. For instance, a friend may elect to see other courses that are newly-created by other authors, but no other information on activities.

A third-party entity performed two phases of testing, using internal GIFT team members to sign in and fully exercise the social media capabilities listed above. Results of initial tests demonstrated that the system is robust (crash-free, but not bug-free), and the value expected from the system was tending to follow assumptions as, for the first time, GIFT users were able to collaborate together using traditional social media methods.

The duration of the tests described above was short, roughly one formal week per each of the two test periods. The system was exercised against the entire online GIFT database with roughly 400 users created and 1000+ total courses. A one-to-many mapping of users to their courses was verified, and as the internal GIFT team members logged in, they were able to perform all of the functions described above satisfactorily. The Elgg framework, once integrated with the GIFT system, provided capabilities beyond those which were expected. As a result, some bugs were discovered, such as not having an automated email system available to reset passwords setup on the test server.

In general, team feedback was positive using "Dislike, Neutral, or Like" scales for each capability. Some test users (< 20% of the 20 participants) were overwhelmed by the available new functionality as the Social Media Framework added an entire new layer of GIFT interactions. The other subset of team members focused solely on the existence of bugs, leading to "Dislike" perceptions that would need to be addressed in a follow-on version. Additionally, the third-party team setup a logging system to allow for any experimental data to be captured, and in 2018 reached a point where further research and direction is required to determine the highest value next courses of action.

Discussion

Students are learning and collaborating at ever-increasing numbers on modern web software tools, thanks to platforms such as Blackboard, Facebook, and GIFT. For this reason, the analysis of gaps between GIFT+Elgg, having been joined through the integration effort with shared user accounts, and other online software suites that serve similar purposes, is a current area of study. The results of that ongoing gap analysis, led the GIFT team to discuss the following items of note. As social media continues to evolve, it can only be assumed that the fusion of social networking and education will evolve as well. In fact, some argue that traditional education methods will be left behind entirely as technology continues to influence learning. "Eventually, teachers and administrators will have difficulty defending traditional pedagogies from the challenge of new perspectives toward learning" (Lee and McLoughlin, 2008, pp. 5). Viewpoints such as those Lee and McLoughlin (2008) presented above, are what we consider to be mainstream hypotheses at the time of this writing. Nearly no contrary evidence is available in public forums that disagrees with the statement, "Social media frameworks provide undeniable value to the learning process." The main discussion in public forums and papers center around which capabilities add the most value, and the main discussion in software/engineering groups is which capabilities add the most value for the least cost of implementation.

Therefore, from the perspective of GIFT research and development, the team focused on the cross-section of perceived value-to-cost of implementation and experimentation concerning social media framework capabilities. Debates and measures of value will continue to be held, and it is estimated that the following items provide the most value to GIFT experimentation:

1) Fixing all bugs with existing capabilities

2) Removing "stubbed" functionality links, or removing all "dead links,"

3) Allowing for additional user customization

4) Investigating additional options for adaptive support and scaffolding with the data

Conclusions and Recommendations for Future Research

The integration of social media in education is here to stay. Learning management systems and online learning platforms are now ubiquitous at all grade levels. Activities in "traditional" classrooms increasingly take place online or even "outside of class." This paradigm shift has made student's online social interactions a core part of the learning process and it provides educators and researchers the opportunity to leverage these tools to enhance students' learning processes. Our goal in this work is to build on the recommendations of Jain and colleagues who proposed creating Learning Management Systems that included social networking and visualizations to better understand their students (Jain, Jethwa, Patil, Poddar, & Somani, 2018). Such social technologies and social analytics have been beneficial for students in many other domains. The next frontier is applying this immersive learning technology to military training where the support can help students to connect with one another, to continue in their coursework, and even to form the community of trainees that is essential to military training. Once existing features are stabilized in the GIFT-Elgg implementation, adding features that allow instructors to better manage their classes and visualize information may therefore be of the highest value. Similarly, as Ali and Qazi (2018, pp. 295) state, "...the usage of SNS-FB for formal learning helped in transforming the students' perception of the assignment. The students perceive that such activities help them not only to complete their class assignment, but it also enhances their learning, improves interaction with the instructor, helps to engage more with peers and most importantly promotes critical thinking".

There is still much more research that needs to be done. Many recent studies suggest that more quantitative assessments should be performed in future experiments. Visualization tools such as instructor dashboards and analytics that display current system usage for further experimentation are likely next in line to experiment with. Additionally, more work should be done to evaluate how the students' online social relationships would relate to their direct study habits, support needs, and goals. While prior research has shown that these social networks can be analyzed to predict performance and relationships, more is necessary to make those predictions into productive guidance.

With the integration of GIFT and the social media framework, we will have increased opportunities for experimentation, intervention, and analysis. According to Ozonur and Tomak (2018), "Researchers should investigate whether motivational and social presence levels increase based on instructional design. These additional studies would contribute further to the literature and the efforts of distance education practitioners to provide better opportunities for learners" (pp. 12). By adding tools to better manage and visualize instructor activities and student performance, we will be able to provide the highest value to the GIFT community. As a result of these improvements, engagement between students and instructors could then be better quantitatively measured. As Abidi, Hussain, Zhang, and Zhu (2018) suggest, student engagement is complex and dependent on several factors such as teaching experience, course design, teaching style, and course concepts. Experimenting with, extending, and integrating tools mentioned in this work will help us meet the experimental goals suggested in prior work. Further, it will help us enhance the learning opportunities for students and instructors and it will move the needle on military training by bringing it into the ever more online and social future.

References

Abidi, S.M.R., Hussain, M., Zhang, W., & Zhu, W. (2018). Student engagement predictions in an elearning system and their impact on student course assessment scores. *Computational Intelligence and Neuroscience,* 2018. Retrieved from https://www.hindawi.com/journals/cin/2018/6347186/cta/.

Adamson, D., Rosé, C. P. Sinha, T., & Yang, D. (2013, December). Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In Proceedings of the 2013 NIPS Data-driven education workshop, 11, 14. Conference Series: Materials Science and Engineering, 336, 1 - 10.

Albert, J., Baker, R., Barnes, T., Bergner, Y., Brown, R., Eagle, M., Lynch, C. F., McNamara, D., & Wang, Y. (2015). Communities of Performance & Communities of Preference. In Proceedings of education data mining (Workshops), Graph Based Educational Data Mining Workshop.

Ali, F. & Qazi, A. (2018). Are social networking sites suitable for formal learning among business research students? A mixed method experimental approach are social networking sites suitable for formal learning among BRS. *Bulletin of Education and Research,* 40, 281-299.

Andres, J. M. L., Baker, R. S., Crossley, S. A., Gašević, D., Joksimović, S., & Siemens, G. March). Studying MOOC completion at scale using the MOOC replication framework. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, 71-88. ACM.

Ball, L., Desbrow, B., Irwin, C., & Leveritt, M. (2012). Students' perceptions of using Facebook as an interactive learning resource at university. *Australasian Journal of Educational Technology, 28*, 1221-1232.

Barnes, T., Gitinabard, N., Heckman, S., Lynch, C. F., & Xue, L. (2017a). A social network analysis on blended courses. In Proceedings of educational data mining (Workshops), Graph-based Educational Data Mining Workshop.

Barnes, T., Gitinabard, N., Heckman, S., & Lynch, C. F. (2017b). Identifying Student Communities in Blended Courses. In Proceedings of educational data mining conference, 378-379.

Blackboard. (2019). Retrieved from https://www.blackboard.com/index.html.

Blom, J., Dillenbourg, P., Li, N., Skevi, A., Verma. & Zufferey. G., (2014) Watching MOOCs Together: investigating co-located MOOC study groups. *Distance Education* 35(2), 217-233

Bohley, M., Gunawardena, C. N., Hermans, M. B., Richmond, C., Sanchez, D., & Tuttle, R. (2009). A theoretical framework for building online communities of practice with social networking tools. Educational Media International, 46(1), 3-16.

Burns, J.R., Light, P., Light, V., & Nesbitt, E. (2010). 'Let's You and Me Have a Little Discussion': Computer mediated communication in support of campus-based university courses. *Studies in Higher Education, 25*, 85-96.

Capelleri, V.M., Peserico, E., & Samory, M. (2017, February). Quotes Reveal Community Structure and Interaction Dynamics. In Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, 322-335. ACM.

Carlson, R., Goldman, Resnick, L.P., Rosé, C. P., Sherer, J., Wen, M., & Yang, D.(2014, March). Social factors that contribute to attrition in MOOCs. In Proceedings of the first ACM conference on Learning@ scale conference, 197-198. ACM.

Chen, Y., & Zhang, M. (2017, May). MOOC student dropout: pattern and prevention. In *Proceedings of the ACM Turing 50th Celebration Conference-China*, 4. ACM.

Choi, G.W., Jablokow, K. W., Pursel, B. K., Velegol, D., & Zhang, L. (2016). Understanding MOOC students: motivations and behaviours indicative of MOOC completion. *Journal of Computer Assisted Learning*, 32(3), 202-217.

Chuateco, L. I., Dennis, J. M., & Phinney, J. S., (2005). The role of motivation, parental support, and peer support in the academic success of ethnic minority first-generation college students. Journal of college student development. *Journal of College Student Development*, 46(3), 223-236.

Dabbish, L., Kraut, R., & Seering, J. (2017, February). Shaping pro and anti-social behavior on twitch through moderation and example-setting. In Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, 111-125. ACM.

Dawson, S. (2010). 'Seeing' the learning community: An exploration of the development of a resource for monitoring online student networking. *British Journal of Educational Technology*, 5, 736-752.

Dawson, S., Gašević, D., Hennis, T., Joksimović, S., Kovanović, V., Poquet, O., de Vries, P. (2018). Social presence in massive open online courses. *International Review of Research in Open and Distributed Learning*, 19(3), 42-68.

De La Fuente, P., Dimitriadis, Y., Gómez, E., Martınez, A., & Rubia, B. (2003). Combining qualitative evaluation and social network analysis for the study of classroom social interactions. *Computers & Education*, 41(4), 353-368.

Eckles, J. E., & Stradley, E. G. (2012). A social network analysis of student retention using archival data. *Social Psychology of Education*, 15(2), 165-180.

Eklund, A.A. & VanDoorn, G. (2013). Face to Facebook: Social media and the learning and teaching potential of symmetrical, synchronous communication. *Journal of University Teaching & Learning Practice*, 10, 1-14.

Elovici, Y., Fire, M., Katz, G., Rokach, L., & Shapira, B. (2012, December). Predicting student exam's scores by analyzing social network data. In *International Conference on Active Media Technology*, 584-595. Springer, Berlin, Heidelberg.

Farmer, T. W., & Rodkin, P. C. (1996). Antisocial and prosocial correlates of classroom social positions: The social network centrality perspective. *Social Development*, *5*(2), 174-188.

Fritz, K.M. (2003). Using Blackboard 5 to deliver both traditional and multimedia quizzes on-line for the foreign language class. ERIC Document Reproduction No. ED482584.

Gitinabard, N., Khoshnevisan, F., Lynch, C. F., & Wang, E. Y. (2018). Your Actions or Your Associates? Predicting Certification and Dropout in MOOCs with Behavioral and Social Features. In *Proceedings of educational data mining*, 404-410.

Heo, H., Kim, Y., & Lim, K. Y. (2010). Exploratory study on the patterns of online interaction and knowledge co-construction in project-based learning. *Computers & Education*, *55*(3), 1383-1392.

Jain, N., Jethwa, B., Patil, A., Poddar, P. & Somani, S. (2018). E-learning portal using social networking features. *International Journal on Recent and Innovation Trends in Computing and Communication, 6*, 161-165.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge university press.

Lee, M.J.W. & McLoughlin, C. (2008). Mapping the digital terrain: New media and social software as catalysts for pedagogical change. *ASCILITE*. Retrieved from http://www.ascilite.org.au/conferences/melbourne08/procs/mcloughlin.pdf.

Lei, X., Sinha, T., & Wang, W. (2015). Teasing Apart Behavioral Protocols in Longitudinal Self-reported Friendship Networks. *NIPS Workshop on Networks in Social and Informational Sciences (NIPS 2015), Montreal, Canada, December 12, 2015*. Stanford University, School of Engineering, Management Science & Engineering (MS&E). Retrieved from http://hdl.handle.net/20.500.11850/195548

Lewis, B., MacEntee, V., & Youngs-Maher, P. (2002). Software compare and contrast: Blackboard, Web, and Lotus Notes a panel discussion, presented at Informing Science & IT Education Conference, Cork, 2002. Santa Rosa, CA: Informing Science Institute.

Oleksandra, P., & Shane, D. (2016, April). Untangling MOOC learner networks. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*. 208-212. ACM.

Ozonur, M. & Tokmak, H.S. (2018). Social presence and motivation in online environments: Second Life versus the Enocta Learning Management System/Adobe Connect. *Australasian Journal of Educational Technology, 34*, 1 - 14.

Rabbany, R., Takaffoli, M., & Zaïane, O. R. (2011). Analyzing participation of students in online courses using social network analysis techniques. In *Proceedings of educational data mining conference*, 21-30.

Settle, A., & Steinbach, T. (2016, September). Improving Retention and Reducing Isolation via a Linked-courses Learning Community. In *Proceedings of the 17th Annual Conference on Information Technology Education* 34-39. ACM.

Wenger, E. (1998). *Communities of Practice: Learning, Meaning, and Identity.* Cambridge University Press.

Chapter 18 – Considerations for Developing Self-Improving Systems to Support Phronesis: Moral and Ethical Thinking and Reasoning in Military Populations

Jeanine A. DeFalco[1,2] and Joseph Doty [3]
[1]U.S. Army Combat Capabilities Development Command – Soldier Center – Simulation and Training Technology Center, [2]Oak Ridge Associated Universities, [3]Duke University

Introduction

Informing self-improving systems to tune their models at runtime and take possible effects into account includes designing systems that can re-arrange their overall structure and thereby self-improve performance (Bellman, Tomforde, & Würtz, 2014). In the field of military education and training, to successfully design a self-improving system still requires basic research into what are the most salient traits and competencies that are relevant in human-computer interaction interactions from which a system can self-improve. One area that is ripe to examine this interaction of traits and contexts is in the area of moral and ethical decision making, or perhaps more appropriately, the Ancient Greek notion of *phronesis,* an area directly implicated in the goal of developing military leaders that can disobey orders "smartly"—one of three main considerations Army Chief of Staff General Milley has identified as necessary in preparing future US Army leaders (Barno & Bensahel, 2017).

Self-Improving Systems

Bellman, Tomforde, and Würtz (2014) have provided an insightful analysis of how self-improving systems incorporate several aspects of mutual influence between components, including how heterogeneous elements interact, the indirect influence of observable behaviors, and the context of the system. These elements depend on valid and timely knowledge of how these components interweave to address dynamically changing goals and priorities, the availability of resources, and details of operational contexts. Mastery of these systems requires self-organization elements such as adaptivity and flexibility to achieve a more organic, life-like functionality. It could be argued that the complex challenges in designing effective modelling, analysis, and infrastructure for a self-improving system that supports successfully interwoven systems could be aided in setting boundary conditions on objective functions that guide behavior.

Bellman et al. (2014) propose that one such possible solution includes developing a model in a domain that is inherently uncertain and dynamic that still allows for a constrained set of goals and priorities that would allow related systems to reflect, evaluate, and respond about the possible effects of interactions. With this thoughtful analysis in mind, the authors of this chapter suggest the domain of moral and ethical decision making for military decision-makers. However, we want to contextualize our notions of moral and ethical decision making in a religious neutral framework. When we discuss moral and ethical decision making, our orientation is more along the lines of the notion of *phronesis. Phronesis* is Greek for "practical wisdom," derived from learning and evidence of practical things. Thinking, creativity, and discriminate intelligence are the products of *phronesis*, enabling individuals to apply discernment in human activity, make good judgements, and determine the right thing to do in practical everyday situations. It is within this context that we define the conversational landscape of moral and ethical decision-making.

Ethical Decision Making in Military Actions

The complexity of decision-making for military leadership includes encompassing the ethical implications of military actions (Reed et al., 2016). In 2006, a mental health advisory team (MHAT), working for the Office of the Surgeon General, published a report that addressed the mental health of combat soldiers deployed in Operation Iraqi Freedom, but also addressed the topic of battlefield ethics (MHAT, 2006). In this report, the following findings were reported:

- Whereas 85% of Soldiers and Marines reported receiving training in treating non-combatants, 33% of Marines and 29% of Soldiers did not agree that their commanding officers were explicit in terms of prohibiting mistreatment of non-combatants.

- The data revealed that 47% of Soldiers and 38% of Marines agreed that non-combatants should be treated with dignity and respect, whereas 17% agreed that all non-combatants should be treated as insurgents.

- Over a third of Soldiers and Marines reported that torture should be allowed in circumstances where it was deemed necessary to save the life of a fellow Soldier or Marine or to obtain important information about insurgents.

- The data revealed that 28% of Soldiers and 31% of Marines reported facing ethical situations in which they did not know how to respond.

Of particular relevance to military education and training is the last reported statistic: the uncertainty of 28% of Soldiers and 31% of Marines who reported facing ethical situations in which they did not know how to respond. It is in this gap, the gap to provide education and training to support ethical considerations in decision-making, where self-improving, machine ethics systems are needed to address not only top-level military leadership, but all military personnel.

Litz et al. (2009) suggest that combat experiences that violate deeply held moral beliefs and expectations may lead to moral injury and spiritual distress. Wortmann et al. (2017) maintain that helping morally injured war veterans falls within the scope of mental health clinicians, whereas the MHAT report recommended that soldiers be given battlefield ethics training (MHAT, 2006).

There have been some investigations into the plausibility of replacing or supplementing human soldiers with autonomous robotic warriors to improve ethical outcomes in combat situations, as well as providing human soldiers with support tools for ethical-decision making (Reed & Jones, 2013). However, developing a system that combines human behavioral and goal inputs with the capabilities of a self-improving system is arguably a better model than those that rely solely on mathematical models (Reed & Jones, 2013) or case-based reasoning technology (Althuizen & Wierenga, 2014).

There is an emerging body of research in machine ethics that involves developing machines, either as hardware or as mathematical/logical models, with codified ethics principles, parameters, and procedures that allow decision-makers to consider the ethical implication of potential actions (Reed et al., 2016). This field involves the design and development of models that use rules, attributes, consequences, or principles associated with actions to suggest an ethically sound action or guide the user through an ethical decision-making process (Reed et al., 2016). Given the complexity and constraints in this nascent field of machine ethics, this work confirms our belief that the moral and ethical decision-making domain is a viable place to examine and test optimization of designs for self-improving systems.

Importance of Ethical Decision-Making in Military Contexts

Ethical decision-making can best be described as a process of rational analysis oriented towards resolving an ethical dilemma (Betan, 1997). To date, there have been a number of different ethical decision-making models that offer linear instructions for navigating through dilemmas (Cottone, & Claus, 2000), yet there is no standard model to address this issue as ethical decision making has, since the time of the Ancient Greeks, been acknowledged as an inherently complex task (Neukrug, et al., 1996). Yet this complexity is not a viable excuse for negligence in this area.

Contrary to popular opinion, military culture is not one that insists on unquestioning and blind obedience to leadership (Mastroianni, Kimmelman, Doty, & Thomas, 2011). Rather, there is a standing legal obligation for service members to disobey orders under circumstances (Mastroianni, Kimmelman, Doty, & Thomas, 2011) and more recently, for military leadership to learn how to disobey orders "smartly," as noted by Chief of Staff of the Army General Mark Milley in his address to attendees at the 4 May 2017 Commanders Series Event (Barno & Bensahel, 2017).

The notion that soldiers are required to challenge and disobey unlawful orders harkens back to 1799 when the US Supreme Court held Navy commanders "act at their own peril" when obeying illegal presidential orders (Powers, 2018). Famously, the My Lai Massacre of 16 March 1968 rejected First Lieutenant William Calleys' defense of "I was only following orders" when he was sentenced to life in prison after his conviction of premeditated murder.

In 2004, the mistreatment of prisoners in Iraq again underscored the criminality of this offense under both international law and the Uniform Code of Military Justice (Powers, 2018). However, military personnel can still be held accountable for crimes committed even when obeying orders. And while there is no requirement for obeying orders that are unlawful, disobeying orders is done at great peril by service members. This is because the determination of an illegal or unlawful order rests on the determination of military superiors and the military court system (Powers, 2018).

This appeal to higher authority to resolve disagreements means that military service men and women are at great risk if they choose to disobey an order, even though at the same time they are legally bound to a standard that requires them to act ethically. This is clearly a contradiction of ethical directives. Compounding this murkey space is the fact that irrespective of legal consequences, the psychological consequences that arise from the guilt or doubt of executing or failing to execute an order can have long term negative consequences (Mastroianni, Kimmelman, Doty, & Thomas, 2011), including contributing to PTSD (Litz et al., 2009; Bryan, Ray-Sannerud, Morrow, & Etienne, 2013). Given the gravity of both these potential outcomes, the necessity to develop systems that can address the development of moral and ethical decision-making, or *phronesis,* is not only self-evident but has been recognized by Army leadership as an imperative for the future success of the US Army (Barno & Bensahel, 2017). These efforts should begin with developing a self-improving system that can adaptively guide and support learners in moral and ethical thinking and reasoning training.

Informing Designs of Self-Improving Systems Derived from AIS Practices

To date, there have been a number of different moral/ethical decision-making models that offer linear instructions for navigating through dilemmas (Cottone, & Claus, 2000), yet there is no standard adaptive instructional model to address this issue. Given the complexity of the domain in combination with the inherent unique traits and abilities of learners, adaptive instructional systems (AISs) may be the best model to devise and deploy replicable, effective moral/ethical decision-making training.

AISs are computer-based systems that guide learning experiences by adapting instruction and recommendations based on the goals, needs, or preferences of the learner as well as intelligent computational procedures (Sottilare, Barr, Robson, & Graesser, 2018). While there is a lack of consensus on the exact nature of an AIS, one viable description identifies AISs as systems that include one or more of the following three broad approaches to support learning objectives for the learner: macrolevel adaptions, aptitude-treatment interactions, microlevel adaptions (Park & Lee, 2003).

Macrolevel adaption includes instructional alternatives selected on the basis of student's instructional goals, ability, and achievements in a curriculum structure (Park & Lee, 2008). Aptitude-treatment interactions include adaptions relevant to specific instructional procedures and strategies specific to learner characteristics (Park & Lee, 2008). Lastly, microlevel adaptions include adaptations of instruction by way of assessment of a learner's specific learning needs during instruction and subsequent responsive instructional prescription in response to those needs (Park & Lee, 2008).

Macrolevel Adaption

Devising a self-improving system informed by best practices in AISs includes determining the parameters of the macrolevel adaption for supporting moral and ethical thinking and reasoning skills. For the purposes of this chapter, our initial parameters include addressing the needs of the US Army and their service members. Building within that parameter space, instructional goals need to be aligned by the US Army's leadership on key principles on ethical and moral decision-making in combat contexts. This would include an instructional design that would support the development of expert problem solving within the domain of ethical and moral decision making in a military context. Possible iterations of an instructional design could include game-based learning, but a preferable model would feature dialogic, interactive activities (Chi, 2009) facilitated by a human-virtual agent to support the development of a learner's critical thinking and perspective of role-taking abilities.

It is important to note here that there is a robust body of literature that has provided compelling evidence that instructional and joint interactive discourse activities are central to effective learning (Chi, 2009; Chi & Wylie, 2014; Reznitskaya & Gregory, 2013). Chi (2009) notes that these kinds of interactive activities include self-construction, guided-construction, sequential-construction, and co-construction of knowledge. These activities represent spaces where the learner has the benefit of contributions from a dialogue partner, including receiving additional information, new perspectives, corrective feedback, or -- what is particularly important for moral and ethical thinking and reasoning development -- opportunities to pursue new paths or lines of reasoning.

But perhaps even more relevant to this discussion is the empirical evidence that indicates the potential of interactive discourse activities as effective methods to develop higher order thinking and deeper understanding of subject-matter knowledge for learners (Chi & Wylie, 2014). The importance of this provides parameters that can be used to evaluate the affordances of employing game-based learning or a human-virtual agent in developing moral/ethical thinking and reasoning abilities in the learner. Namely, the effectiveness of both platforms lay less in their novelty or ability to motivate or engage learners. Rather, game-based environments and human-virtual agents can potentially provide superior replicable and trackable learning environments when oriented towards supporting interactive discourse activities that facilitate co-construction of knowledge – a central element in developing moral and ethical thinking and reasoning.

Aptitude Treatment Interactions

Initial exploratory work should also include identifying aptitude-treatment interactions that are adaptive to the learner as derived from specific learner characteristics. In this instance, examining the learner characteristic of moral sensitivity would be instrumental in this effort. Moral sensitivity is the ability of a person to recognize when an ethical issue even exists (MacIntyre, Doty, & Xu, 2016). While thinking morally or ethically may not be an inherent characteristic skill or trait, the literature suggests that this capacity may be developed (Jordan, 2007; MacIntyre, Doty, & Xu, 2016; Rest, 1986; Reynolds, 2008; Sadler, 2004; Sparks & Hunt, 1998).

Moral Sensitivity

A central tenant to thinking morally or ethically is the notion that moral sensitivity includes the awareness of how one's actions effect other people, and the ability to critically reflect on possible scenarios of cause-and-effect consequences of events in the real world (MacIntyre, Doty, & Xu, 2016). To successfully accomplish this requires the ability to be perceptive and sensitive in a given situation, an ability to achieve mindfulness in order to recognize moral issues, or in other words, the ability to be empathetic and engage in perspective or role-taking (MacIntyre, Doty, & Xu, 2016).

As a comprehensive approach to understanding ethical sensitivity, MacIntyre, Doty, and Xu (2016) have developed a global ethical sensitivity instrument, the *Life Events Survey*, that is domain-general and can be used to measure the "Degree of Mindfulness,"—mindfulness having been identified by these same authors as a capacity centrally implicated in moral sensitivity. Ideally, this instrument could be used as a central, self-organizing element upon which adaptivity and flexibility would drive instruction, interventions, and feedback.

Microlevel Adaptions

Arguably, addressing microlevel adaption is the area that would be the most challenging in developing a self-improving system to support ethical and moral thinking and reasoning. The microlevel represents the meeting place where the macrolevel and aptitude treatment interactions interweave to address dynamically changing goals and priorities, assess the availability of resources, monitor details of operational contexts by way of assessment of a learner's specific learning needs during instruction and their responses to instructional prescriptions, or ideally interactive discourse activities. This would require the integration of sensors that can provide data on facial and gestural cues, as well as the integration of natural language processing capabilities that could detect, decipher, and respond to discourse activities. These activities would dynamically adapt as the learner progresses through activities that would support self-awareness, self-regulation, and opportunities to think, reflect, and practice skills that can be called upon in future chaotic and morally ambiguous combat circumstances.

In essence, employing instructional mediums such as game-based or human-virtual agents in a self-improving system would allow for a framework of standardization of instruction that is still adaptable, flexible, and responsive to the learner. Further, this approach would allow for an assessment function that could be monitored by instructors, enabling a comprehensive collection of data to track learner's progress and use as a forward feed for future self-improving iterations of the learning experience. For example, employing the Generalized Intelligent Framework for Tutoring (GIFT) (Sottilare, Brawner, Sinatra, & Johnston, 2017) as the framework within which to build this moral and ethical training system would be an optimal platform to both design and deliver an adaptive course in ethical and moral decision making.

GIFT's functionality allows for instructional designers to integrate game-based and/or human-virtual agent mediums into a singular course. Additionally, trait assessment surveys and pre- and post-tests can be built into the same course that could be used to adapt the sequencing and complexity of content for each individual learner. GIFT also provides a comprehensive data output on not only survey and assessment responses, but also actions taken while an individual is engaged with the game-based and/or human virtual mediums, allowing for a comprehensive fine-grained analysis on the learner's engagement and learning outcomes. In this way, designing a self-improving system to support the moral and ethical decision making capabilities of emerging military leaders can be standardized at a macro-level, yet responsive to individual traits and competencies at the microlevel. Using GIFT as the framework for this self-improving system would also allow for cost-efficient replication and ease of delivery when capitalizing on GIFT's cloud-based functionality (Sottilare, Brawner, Sinatra, & Johnston, 2017).

Final Thoughts

Design considerations for a self-improving system includes an examination of the elements necessary to realize an interwoven, adaptive, and life-like system. This includes creating a system that is driven by domain-specific goals that contain intelligent parameters, and implements components that capitalize on relevant theories of human behavior and cognition. A self-improving system oriented towards supporting the ethical and moral decision-making capabilities (as aligned with the notions of *phronesis*) of combat military personnel not only is an appropriately dynamic and complex domain but addresses a critical need for the present and future US Army. Building such a system in the GIFT platform would not only provide a replicable course that can adapt instruction and sequencing of courses for individual learners, but it would provide an ease of delivery for emerging military leadership around the world.

References

Althuizen, N., & Wierenga, B. (2014). Supporting creative problem solving with a case-based reasoning system. *Journal of Management Information Systems, 31*(1), 309-340.

Barno, D., & Bensahel, N. (2017). Three things the Army Chief of Staff wants you to know. *War on the Rocks*. Retrieved online on 17 January 2019 at https://warontherocks.com/2017/05/three-things-the-army-chief-of-staff- wants-you-to-know/.

Bellman, K., Tomforde, S., & Würtz, R. P. (2014). Interwoven systems: Self-improving systems integration. In *Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2014 IEEE Eighth International Conference on* (pp. 123-127). IEEE.

Betan, E. J. (1997). Toward a hermeneutic model of ethical decision making in clinical practice. *Ethics & Behavior, 7*(4), 347-365. doi: http://dx.doi.org/10.1207/s15327019eb0704_6

Bryan, C. J., Ray-Sannerud, B., Morrow, C. E., & Etienne, N. (2013). Guilt is more strongly associated with suicidal ideation among military personnel with direct combat exposure. Journal of affective disorders, 148(1), 37-41.

Chi, M. T. (2009). Active-constructive-interactive: A conceptual framework for differentiating learning activities. *Topics in Cognitive Science, 1*(1), 73-105.

Chi, M. T., & Wylie, R. (2014). The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational Psychologist, 49*(4), 219-243.

Cottone, R. R., & Claus, R. E. (2000). Ethical decision-making models: A review of the literature. *Journal of Counseling & Development, 78*, 275-283. doi: http://dx.doi.org/10.1002/j.1556- 6676.2000.tb01908.x

Jordan, J. (2007). Taking the First Step toward a Moral Action: A Review of Moral Sensitivity Measurement across Domains." *The Journal of Genetic Psychology, 168*(3), 323–359.

Litz, B. T., Stein, N., Delaney, E., Lebowitz, L., Nash, W. P., Silva, C., & Maguen, S. (2009). Moral injury and moral repair in war veterans: A preliminary model and intervention strategy. *Clinical Psychology Review, 29,* 695–706. http://dx.doi.org/10.1016/j.cpr.2009.07.003

MacIntyre, A., Doty, J., and Xu, D. (2016). Ethical sensitivity during military operations: Without mindfulness there is no sensitivity. In: S. Belanger and D. Lagace-Roy (Eds.), *Military ethics and well-being*. Montreal, Canada: McGill-

Queen's University Press.
Mastroianni, G., Kimmelman, S., Doty, J., & Thomas, J. (2011).Obedience and Personal Responsibility. In (Eds.) P. Sweeney, M. Matthews, & P. Lester, *Leadership in Dangerous Situations : A Handbook for the Armed Forces, Emergency Services, and First Responders*. Annapolis: Naval Institute Press.
Mental Health Advisory Team (MHAT) IV, Office of the Surgeon General (2006) Final report: operation Iraqi freedom 05-07
Neukrug, E., Lovell, C., & Parker, R. J. (1996). Employing ethical codes and decision-making models: A developmental process. *Counseling and Values*, *40*(2), 98-106. doi: http://dx.doi.org/10.1002/j.2161-007X.1996.tb00843.x
Park, O. C., & Lee, J. (2003). Adaptive instructional systems. *Educational Technology Research and Development*, 25, 651-684.
Powers, R. (2018). What to know about obeying an unlawful military order. US Military Careers: Laws and Legislation. Retrieved at Caution-https://www.thebalancecareers.com/military-orders-3332819 < Caution-https://www.thebalancecareers.com/military-orders-3332819 >
Reed, G. S., & Jones, N. (2013). Toward modeling and automating ethical decision making: design, implementation, limitations, and responsibilities. *Topoi*, *32*(2), 237-250.
Reed, G. S., Petty, M. D., Jones, N. J., Morris, A. W., Ballenger, J. P., & Delugach, H. S. (2016). A principles-based model of ethical considerations in military decision making. *The Journal of Defense Modeling and Simulation*, *13*(2), 195-211.
Rest, J.R. (1986). *Moral Development: Advances in Research and Theory*. Westport, CT: Praeger.
Reynolds, S. (2008). "Moral Attentiveness: Who Pays Attention to the Moral Aspects of Life?" *Journal of Applied Psychology* 91(5): 1027–41.
Reznitskaya, A., & Gregory, M. (2013). Student thought and classroom language: Examining the mechanisms of change in dialogic teaching. *Educational Psychologist*, *48*(2), 114-133.
Sadler, T. D. (2004). Moral sensitivity and its contribution to the resolution of socio-scientific issues. *Journal of Moral Education*, *33*(3), 339-358.
Sottilare, R., Barr, A., Robson, R., Hu, X., & Graesser, A. (2018). Exploring the Opportunities and Benefits of Standards for Adaptive Instructional Systems (AISs) In Proceedings from *Intelligent Tutoring Systems Conference AIS Standards Workshop*, Montreal, Quebec, Canada Retrieved at https://www.researchgate.net/publication/328296318_Exploring_the_Opportunities_and_Benefits_of_Standards_for_Adaptive_Instructional_Systems_AISs
Sottilare, R. A., Brawner, K. W., Sinatra, A. M., & Johnston, J. H. (2017). An updated concept for a Generalized Intelligent Framework for Tutoring (GIFT). *GIFTtutoring. org*.
Park, O. C., & Lee, J. (2003). Adaptive instructional systems. *Educational Technology Research and Development*, 25, 651-684.
Sparks, J.R., & Hunt, S.D. (1998). "Marketing Researcher Ethical Sensitivity: Conceptualization, Measurement and Exploratory Investigation," *The Journal of Marketing*. 62(2): 92–109.
Wortmann, J. H., Eisen, E., Hundert, C., Jordan, A. H., Smith, M. W., Nash, W. P., & Litz, B. T. (2017). Spiritual features of war-related moral injury: A primer for clinicians. *Spirituality in Clinical Practice*, *4*(4), 249.

Acknowledgement

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number **W911NF-17-2-0152**. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Chapter 19 – Simple Humans, Evolving Computation, Smart Intelligent Tutoring Systems

Arthur C. Graesser[1], Jeanine A. DeFalco[2,3] and Jody L. Cockroft[1]
[1]University of Memphis, [2]U.S. Army Combat Capabilities Development Command - Soldier Center – Simulation and Training Technology Center, [3]Oak Ridge Associated Universities

Introduction

A fundamental approach to developing self-improving intelligent tutoring systems (ITSs) is to have different sources of assessment on the quality of the decisions, solutions and algorithms. The sources of assessment can be categorized in various ways, but we identify the major categories as being theoretical models, computational algorithms, human judgments, and empirical performance. Quality is high on the *theoretical model* front when an implementation (i.e., decision, solution, algorithm, or entire system) is compatible with the features of a theory, model, or hypotheses that is widely accepted in scientific research and practice. The quality of a *computational algorithm* increases to the extent that implementation is fast, efficient, complete, general, interoperable, and so on. There are different stakeholders who can provide *human judgments* of an implementation: learners, instructors, learning engineers, researchers, and supervisors. *Empirical performance* includes learning gains, engagement, and other psychological measures that can be objectively measured and that validate the success of an implementation.

The main point of this chapter is to make the case that all sources of assessment have limitations that designers of self-improving intelligent tutoring systems (SI-ITSs) need to consider. There is no perfect gold standard. Designers bicker endlessly on which theoretical model to adopt and on the adequacy of a computational algorithm. Humans disagree in their judgments and typically rely on simple cognitive heuristics when making judgments and decisions. Many would agree that empirical performance should be high in the priority of considerations, but performance data are often difficult to obtain, and alternative measures of performance sometimes reflect trade-offs rather than consistency. Given there is no perfect gold standard for the quality of an implementation, judgments will need to be made in setting priorities and weights on the alternative sources of assessment in SI-ITS design.

It is widely acknowledged that the existing ITSs are far from perfect and that the various blemishes run the risk of frustrating learners or having a minimal impact on learning gains. That is a major part of the inspiration for building SI-ITSs: The ITS will hopefully improve over time as data are collected and computational algorithms are modified through machine learning. In contrast, it is too often not acknowledged that human judgments are typically based on simple heuristics and that humans often disagree. The simplicity, errors, and other limitations of human judgments need to be better understood in order to make wiser decisions on when they should be trusted. This chapter addresses the limitations and implications of human judgments in SI-ITS design.

Related Research

The design of many ITSs rely on judgments of humans. Curriculum experts generate the lessons, subject matter experts identify important content, language experts decide whether natural language expressions are similar in meaning, "markers" grade the quality of essays, students judge the quality of the learning experience, and instructors judge the pedagogical quality of the ITS. These judgements are necessary for the generation, modification, and tuning of ITS content and mechanisms.

ASSISTments (Heffernan & Heffernan, 2014, https://www.assistments.org/) and *BrainTrust* (Olney, 2018) are two exemplary ITSs that illustrate how human judgments are part of the ongoing process of improving an ITS. ASSISTments is a system that allows teachers to create materials for mathematics (as well as other topics) to see how well students perform, and to interact with researchers on possible improvements based on the science of learning. Authoring tools are available to guide the instructors in creating the materials. The *Builder* guides the curriculum designer or teacher in creating lessons, whereas the *Teacher* view shows performance of particular students on particular lessons, and the *Student* view guides the students in completing tasks and viewing feedback on their performance. These three perspectives are extremely important for scaling up a system because it accommodates the points of view of curriculum designers, instructors, and students. In 2015, ASSISTments was used by over 600 teachers in 43 states and 12 countries, with students completing over 10 million mathematics problems. Learning gains are well-documented and explain the success in the system being scaled up for widespread use. Rochelle, Feng, Murphy, and Mason (2016) reported that ASSISTments improved mathematics scores reliably and larger than normal expectations of growth.

In *BrainTrust* (Olney & Cade, 2015) students read and work with a virtual student on a variety of educational tasks related to the reading. These educational tasks are designed to both improve reading comprehension and contribute to the creation of an ITS based on the material they read. After a human student reads a passage, they work with the virtual student to summarize, generate concept maps, reflect on the reading, and answer questions. The virtual student's performance on these tasks is a mixture of previous student answers and answers dynamically generated using Artificial Intelligence (AI) and natural language processing techniques. As the human teaches and corrects the virtual student, they in effect improve the answers from previous sessions, author tutorial dialogues, and improve domain models underlying the ITS. BrainTrust creates content by letting novices and AI do the authoring but then letting other novices check the work to ensure quality. As the human teaches and corrects the virtual student, they in effect improve the answers from previous sessions and author a domain model for the underlying ITS.

ASSISTments and BrainTrust are examples of SI-ITSs that are hybrids between human judgments and computational algorithms as sources of quality. There are a number of other systems that follow a similar hybrid foundation, such as *SimStudent* (Matsuda et al., 2013). In addition to the original content and tutoring strategies being guided by theory, they can evolve in a data driven fashion through crowd sourcing and machine learning on the growing corpus. Learning gains have also been assessed in these systems as measures of performance. Consequently, all four sources of quality assessment are to some extent involved in the development of these SI-ITSs.

It is important to acknowledge, however, that many if not most of the judgments of humans are imperfect. When humans make judgments and decisions, they routinely use simple mental shortcuts, called *heuristics*, to handle complexity and uncertainty. This results in their considering one or a few aspects of a complex problem but ignoring others. The simple heuristics work under many circumstances, but they also sometimes systematically deviate from rational theory and accuracy. Tversky and Kahneman (1974) documented many of these heuristics in their landmark *Science* paper. For example, according to the *availability* heuristic, humans often select options that are easier to access from memory. According to the *base rate fallacy*, people overestimate a very rare property of a category and underestimate a very common property. Human judgments are constrained by available information, time constraints, and cognitive limitations during the reasoning process.

Sometimes human judgments can be quite reliable in the sense that the individuals agree. For example, trained judges have a reasonable level of agreement when judging whether two language expressions are similar in meaning (Cai et al., 2011; Rus, Olney, Foltz, & Hu, 2017; Rus, Lintean, Banjade, Niraula, & Stefanescu, 2013) and the quality of an essay (Foltz, Streeter, Lochbaum, & Landauer, 2013). The agreement is assessed with a variety of indices, such as correlations, kappa scores that control for baserate distributions, F-measures in computational linguistics, and so on. In contrast, trained judges show low to moderate agreement on judgments of syntactic complexity (Graesser, Wiemer-Hastings, Kreuz, Wiemer-Hastings & Marques, 2000) and emotions of learners during learning (D'Mello, Craig, & Graesser, 2009). Novices show high agreement on the extent to which word concepts evoke a vivid mental image but disagree substantially on how interesting they consider the word concepts. The value of human judgments obviously decreases with the reliability of their judgments. Designers of SI-ITSs need to systematically keep track of which human judgments to trust and which to discard.

Added to this complexity is the difficulty inherent in devising assessments that do not merely align with content but measure the intent of the learning objectives. Black (2004) maintained that any assessment of learning should be aimed to promote students' learning. Importantly, Black (2004) distinguished between assessment activities that serve the purposes of accountability or ranking, and that of certifying competence. Further, when assessments provide feedback that influences modification of teaching and learning activities, assessments are formative in the learning experience. The formative assessments provide evidence for teachers to adapt their teaching activities to better meet the learning needs of their students (Black, 2004). Accordingly, without thoughtful consideration as to the why or to what end content is taught, assessment measures will not capture whether learning actually occurred and whether it will support further learning.

Perhaps accomplished teachers and pedagogical theories can come to the rescue and help sort out which human judgments to trust. Unfortunately, the theoretical foundations of pedagogy in colleges of education and classroom practice are *not* closely aligned with the science of learning. For example, Nathan and Petrosino (2003) documented that "expert blind spots" of teachers are often incompatible with algebra performance. The teachers' professional development leads them to conclude that a symbolic algebra problem should precede a contextualized word problem, whereas the performance data suggest the opposite. Nathan and Petrosino (2003) identify a number of expert blind spots that are alarmingly frequent across mathematics, science, and the language arts.

A study commissioned by the National Council on Teacher Quality (Pomerance, Greenberg, & Walsh, 2016) investigated whether some well-documented scientific principles of learning were included in the professional development programs in colleges of education. These principles included (1) space learning over time, (2) interleave worked example solutions with problem solving exercises, (3) combine graphics with verbal descriptions, (4) connect and integrate abstract and concrete representations of concepts, (5) use quizzing to promote learning, and (6) ask deep explanatory questions (Pashler et al., 2007). Pomerance et al. (2016) conducted an analysis of 48 textbooks and 48 teacher education programs in the United States by tracking the occurrence of the above 6 principles of learning. According to the report, there was not a single textbook that covered the 6 principles and no book covered more than 2 principles. Regarding the coursework, most programs prepared candidates for only a single principle and one-third none at all.

The science of learning, which has often been integrated with modern ITSs, has not had much of an impact on teacher education and practice. There are many potential reasons why the scientific principles of learning and instruction are not integrated with the mainstream educational settings. The teachers may not be aware of the principles or the principles may be too difficult to integrate in a classroom setting. The training may be primarily focused on some of the major summative assessments demanded by the state or by Federal government, such as *No Child Left Behind, Race to the Top,* or the *Common Core*. Perhaps the teachers use more sophisticated pedagogical methods that are superior to the principles of learning. If so, the superior principles need to be articulated, tested, validated, and compared with the existing scientific principles.

The process of one-on-one human tutoring also falls prey to inaccurate human judgments (Chi, Siler, Yamauchi, Jeong, & Hausmann, 2001; Graesser & Person, 1994). Graesser, D'Mello, and Person (2009) analyzed the metacognitive judgments of tutors and students in human-to-human tutoring sessions. They identified five major tutoring illusions that stem from the large gap in knowledge between the tutor and students:

(1) *Illusion of grounding*. The unwarranted assumption that the tutor and student have shared knowledge about a word, referent, or idea being discussed in the tutoring session. Given the low common ground between tutor and tutee, this assumption is false.
(2) *Illusion of feedback accuracy*. The unwarranted assumption that the feedback that the student or tutor gives to each other is accurate. For example, tutors incorrectly believe the students' answers to their comprehension gauging questions (e.g., "Do you understand?"). The more knowledgeable students have a higher likelihood of saying they do *not* understand; there is a significant negative relationship between their understanding and saying yes, they understand (Chi et al., 2001; Graesser & Person, 1994). On the flip side, tutors have a higher likelihood of giving positive than negative short feedback to erroneous or vague student contributions (Graesser & Person, 1994).
(3) *Illusion of discourse alignment*. The unwarranted assumption that the student understands the discourse function, intention, and meaning of the tutor's dialogue contributions. For example, tutors sometimes give hints, but the students do not realize they are hints.
(4) *Illusion of student mastery*. The unwarranted assumption that the student has mastered much more than the student has really mastered. The student believes they have mastered a complex concept when the student expresses a word or phrase rather than a more complete and precisely articulated answer to a tutor question. Tutors often make the same conclusion about a student's mastery.
(5) *Illusion of knowledge transfer*. The tutor's unwarranted assumption that the student understands whatever the tutor says and thereby knowledge is accurately transferred. In actuality, the students typically understand very little, as revealed by their responses to follow up questions and requests for actions or summarization.

One important consequence of these illusions is that tutoring process data can sometimes have misleading results in simple data mining and machine learning procedures. For example, consider a tutoring pattern in which a tutor asks a comprehension gauging question ("Do you understand?"), followed by a positive response by the student (e.g., "Yes", head nod). That might be viewed as a positive signal of success with respect to the previous tutoring exchange in a simple reinforcement learning algorithm. As discussed, however, we know from human tutoring research that it is the less knowledgeable students who tend to answer yes. As another example in the arena of human emotions, the more knowledgeable students tend to experience confusion as they attempt to solve difficult problems that elicit cognitive disequilibrium (D'Mello, Lehman, Pekrun, & Graesser, 2014). Therefore, confusion may be a signal of success rather than failure. The mechanisms of learning and motivation are quite complex, sometimes with counterintuitive tradeoffs, so simple machine learning analyses will run the risk of being misleading.

Discussion and Recommendations

This chapter has identified some of the limitations of the major sources of assessing the quality of an SI-ITS and its components: theoretical models, computational algorithms, human judgments, and empirical performance. So how do designers minimize these problems? We offer three recommendations.

The first recommendation is to engage in an ongoing validation of each source. Theoretical models are consistently being tested in the normal evolution of the scientific method. The chief challenge lies in identifying the precise scope of each theoretical model with respect to the knowledge domain, depth of acquisition, population of learners, and sociocultural context of application (National Academy of Sciences, Engineering, and Medicine, 2018). An ITS for memorizing facts is very different than an ITS for team collaborative problem solving, for example. Computational algorithms are routinely validated with respect to specific data sets, corpora, and applications, but run the risk of having minimal generality to the design of new SI-ITSs. Individual human judges need to be validated with respect to (a) their consistency over time,

(b) agreement with other judges, and (c) alignment with objective performance data; judges need to be removed from consideration if they do not meet these three criteria. And finally, empirical performance measures need to be validated by (a) verifying their similarity to other measures that target the same theoretical construct and (b) considering complex interactions between learning, motivation, and emotions.

The second recommendation is for the field, including the Generalized Intelligent Framework for Tutoring (GIFT), to identify problematic sources of quality assessment. A good example is learning styles (e.g., "I learn best visually rather than in language") that are measured by self-report ratings and judgments (as opposed to process and performance data). Self-reported learning styles are known to be bogus, even though they are popular in the public. Another example is machine learning solutions that optimize a single measure, such as learning efficiency (amount of learning per unit time) that does not consider the difficulty of the material. The wisdom of high quality versus low quality measures needs to be accumulated, preserved, and applied in the GIFT community.

The third recommendation is integrating the quality assessments from all four sources in the self-improvement process. Some features and feature configurations should be necessary or highly weighted, others should be optional with some weight value, and yet others should be zero or weighted negatively. These assessment weights need to be incorporated into the guts of the machine learning algorithms in the evolving SI-ITS. Without this step, the designers are unlikely to develop a smart ITS that improves learning and motivation.

References

Black, P. (2004). *The nature and value of formative assessment for learning*. Unpublished manuscript, Educational Testing Service, Princeton, NJ. Retrieved from http://74.125.155.132/scholar?q=cache:RzppkZG-WasJ:scholar.google.com/+The+Naure+and+value+of+formative+assessment+for+learning+author:black&hl=en&as_sdt=2000

Cai, Z., Graesser, A.C., Forsyth, C., Burkett, C., Millis, K., Wallace, P., Halpern, D. & Butler, H. (2011). Trialog in ARIES: User input assessment in an intelligent tutoring system. In W. Chen & S. Li (Eds.), *Proceedings of the 3rd IEEE International Conference on Intelligent Computing and Intelligent Systems* (pp.429-433). Guangzhou: IEEE Press.

Chi, M.T.H., Siler, S., Yamauchi, T., Jeong, H. & Hausmann, R. (2001). Learning from human tutoring. *Cognitive Science*, *25*, 471- 534.

D'Mello, S.K., Craig, S.D., & Graesser, A.C. (2009). Multi-method assessment of affective experience and expression during deep learning. *International Journal of Learning Technology, 4*, 165-187.

D'Mello, S., Lehman, B., Pekrun, R., & Graesser, A.C. (2014). Confusion can be beneficial for learning. *Learning and Instruction.* 29, 153-170.

Foltz, P. W., Streeter, L. A., Lochbaum, K. E., & Landauer, T. K (2013). Implementation and applications of the Intelligent Essay Assessor. Handbook of Automated Essay Evaluation, M. Shermis & J. Burstein, (Eds.). pp. 68-88. Routledge, NY. NY.

Graesser, A. C., D'Mello, S., & Person, N. K. (2009). Metaknowledge in tutoring. In D. Hacker, J. Donlosky, & A. C. Graesser (Eds.), *Handbook of metacognition in education* (pp. 361-382). New York: Taylor & Francis.

Graesser, A. C., & Person, N. K. (1994). Question asking during tutoring. *American Educational Research Journal*, *31*, 104-137.

Graesser, A. C., Wiemer-Hastings, K., Kreuz, R., Wiemer-Hastings, P., & Marques, K. (2000). QUAID: A questionnaire evaluation aid for survey methodologists. *Behavior Research Methods*, *Instruments, and Computers*, *32*, 254–262.

Heffernan, N. & Heffernan, C. (2014). The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education. 24*, 470-497.

Matsuda, N., Yarzebinski, E., Keiser, V., Raizada, R., William, W. C., Stylianides, G. J., & Koedinger, K. R. (2013). Cognitive anatomy of tutor learning: Lessons learned with SimStudent. *Journal of Educational Psychology*, *105*, 1152-1163.

Nathan, M. J. & Petrosino, A. J. (2003). Expert blind spot among preservice teachers. *American Educational Research Journal. 40*(4), 905-928.
National Academy of Sciences, Engineering, and Medicon (2018). *How People Learn II: Learners, contexts, and cultures.* Washington, D.C.: National Academies Press.
Olney, A. M. (2018). Using novices to scale up intelligent tutoring systems. In Interservice/Industry training, Simulation, and Education Conference (I/ITSEC) 2018.
Olney, A. M., & Cade, W. L. (2015). Authoring intelligent tutoring systems using human computation: Designing for intrinsic motivation. In D. D. Schmorrow & C. M. Fidopiastis (Eds.), *Foundations of Augmented Cognition* (Vol. 9183, pp. 628–639). Springer International Publishing.
Pashler, H., Rohrer, D., Cepeda, N. J., & Carpenter, S. K. (2007). Enhancing learning and retarding forgetting: choices and consequences. *Psychonomic Bulletin & Review*, *14*(2), 187–193. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/17694899
Pomerance, L., Greenberg, J., & Walsh, K. (2016). Learning about Learning: What Every New Teacher Needs to Know. *National Council on Teacher Quality*. Retrieved from http://files.eric.ed.gov/fulltext/ED570861.pdf
Rochelle, J., Feng, M., Murphy, R. & Mason, C. (2016). Online mathematics homework increases student achievement. *AERA OPEN*, *2*, 1–12.
Rus, V., Olney, A. M., Foltz, P. W., & Hu, X. (2017). Automated assessment of learner-generated natural language responses, In R. Sottilare, A. Graesser, X. Hu, & G. Goodwin (Eds.), *Design Recommendations for Intelligent Tutoring Systems: Volume 5 - Assessment.* Orlando, FL: U.S. Army Research Labor*atory.*
Rus, V., Lintean, M. C., Banjade, R., Niraula, N. B., & Stefanescu, D. (2013). SEMILAR: The Semantic Similarity Toolkit. In ACL (Conference System Demonstrations) (pp. 163-168).
Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science*, *185*, 1124-1131.

Acknowledgement

The research on was supported by the National Science Foundation (ACI-1443068), the Institute of Education Sciences (R305C120001), Army Research Lab (W911INF-12-2-0030, W911NF-17-2-015), and the Office of Naval Research (N00014-12-C-0643; N00014-16-C-3027). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the NSF, IES, ONR, ARL or the U.S. Government.

BIOGRAPHIES

Editors

Dr. Keith Brawner is a senior researcher for the U. S. Army Combat Capability Development Command Soldier Center at the Simulation and Training Technology Center (CCDC-SC-STTC), and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT). He has 13 years of experience within U.S. Army and Navy acquisition, development, and research agencies. He holds a Masters and PhD degree in Computer Engineering with a focus on Intelligent Systems and Machine Learning from the University of Central Florida. His current efforts are on artificial intelligence for the Synthetic Training Environment Simulation and Network Compression. He manages research in adaptive training, semi/fully automated user tools for adaptive training content, and architectural programs towards next-generation training.

Dr. Art Graesser is a professor in the Department of Psychology and the Institute of Intelligent Systems at the University of Memphis and is a Senior Research Fellow in the Department of Education at the University of Oxford. He received his Ph.D. in psychology from the University of California at San Diego. Dr. Graesser's primary research interests are in cognitive science, discourse processing, and the learning sciences. More specific interests include knowledge representation, question asking and answering, tutoring, text comprehension, inference generation, conversation, reading, education, memory, emotions, computational linguistics, artificial intelligence, human-computer interaction, and learning technologies with animated conversational agents. Dr. Graesser served as editor of the journal Discourse Processes (1996–2005) and Journal of Educational Psychology (2009-2014) and as president of the Empirical Studies of Literature, Art, and Media (1989-1992), the Society for Text and Discourse (2007-2010), International Society for Artificial Intelligence in Education (2007-2009), and the FABBS Foundation (2012-13). He has published over 500 articles in journals, books, and conference proceedings. Dr. Graesser and his colleagues have designed, developed, and tested software that integrates psychological sciences with learning, language, and discourse technologies, including AutoTutor, AutoTutor-Lite, MetaTutor, GuruTutor, DeepTutor, HURA Advisor, SEEK Web Tutor, Operation ARIES!, iSTART, Writing-Pal, AutoCommunicator, Point & Query, Question Understanding Aid (QUAID), QUEST, & Coh-Metrix.

In 2010, Dr. Graesser received the Distinguished Scientific Contribution Award (Society for Text and Discourse) and in 2011 he received the Distinguished Contributions of Applications of Psychology to Education and Training Award (American Psychological Association). In 2012, Dr. Graesser received the first Presidential Award for Lifetime Achievement in Research from the University of Memphis. This award is the University's highest level of research recognition given to its faculty. It was established as part of the University's Centennial fundraising campaign in order to recognize the vital role and impact of research at the University of Memphis. He served as Chair of the Framework group in PISA Collaborative Problem Solving 2015. In 2018 he received the Harold W. McGraw, Jr. Prize in Education.

Dr. Xiangen Hu is a professor in the Department of Psychology, Department of Electrical and Computer Engineering and Computer Science Department at The University of Memphis (UofM) and senior researcher at the Institute for Intelligent Systems (IIS) at the UofM and is professor and Dean of the School of Psychology at Central China Normal University (CCNU). Dr. Hu received his MS in applied mathematics from Huazhong University of Science and Technology, MA in social sciences and Ph.D. in Cognitive Sciences from the University of California, Irvine. Dr. Hu is the Director of Advanced Distributed Learning (ADL) Partnership Laboratory at the UofM, and is a senior researcher in the Chinese Ministry of Education's Key Laboratory of Adolescent Cyberpsychology and Behavior.

Dr. Hu's primary research areas include Mathematical Psychology, Research Design and Statistics, and Cognitive Psychology. More specific research interests include General Processing Tree (GPT) models, categorical data analysis, knowledge representation, computerized tutoring, and advanced distributed learning. Dr. Hu has received funding for the above research from the US National Science Foundation (NSF), US Institute of Education Sciences (IES), ADL of the US Department of Defense (DoD), US Army Medical Research Acquisition Activity (USAMRAA), US Army Research Laboratories (ARL), US Office of Naval Research (ONR), UofM, and CCNU.

Dr. Vasile Rus is a Professor in the Department of Computer Science at The University of Memphis with a joint appointment in the Institute for Intelligent Systems. He also serves as the Director of The Data Science Research Initiative at The University of Memphis. Dr. Rus is also a Systems Testing Research Fellow of the FedEx Institute of Technology, an honor received for his pioneering work in the area of software systems testing. His research interests lie at the intersection of artificial intelligence, machine learning, and computational linguistics with an emphasis on developing interactive intelligent systems based on strong theoretical findings in order to solve critical challenges that would change the educational and human computer interaction landscape. Dr. Rus has been involved in research and development projects in the areas of computational linguistics and information retrieval for more than 20 years and in open-ended student answer assessment and intelligent tutoring systems for more than 15 years. He has been involved in the development of the following intelligent tutoring systems: DeepTutor (PI), Writing Pal (co-PI), MetaTutor (co-PI), and AutoMentor (co-PI). Dr. Rus has served in various roles on research projects funded by the National Science Foundation, Department of Defense, and Department of Education, and private companies; has won the first two Question Answering competition organized by the National Institute for Science and Technology (NIST); recently his team won the English Semantic Similarity challenge organized by the leading forum on semantic evaluations – SemEval; has received 4 Best Paper Awards; produced more than 150 peer-reviewed publications; and serves in various roles for major Computer Science and EdTech conferences such as AAAI, ACL, and AIED.

Dr. Anne M. Sinatra is a Research Psychologist in the Learning in Intelligent Tutoring Environments (LITE) Lab within the U.S. Army CCDC Soldier Center Simulation and Training Technology Center. The focus of her research is in cognitive psychology, human factors psychology, and adaptive team tutoring. She has specific interest in how information relating to the self and about those that one is familiar with can aid in memory, recall, and tutoring. Her dissertation research evaluated the impact of using degraded speech and a familiar story on attention/recall in a dichotic listening task. Her post-doctoral work examined the self-reference effect and personalization in the context of computer-based tutoring. Her work has been published in journals including the Journal of Artificial Intelligence in Education, Computers in Human Behavior, and Interaction Studies. Her work has also been published in conference proceedings including the Human Factors and Ergonomics Society conference, and the Human Computer Interaction International conference. She additionally has served as an editor on three books, and chaired two team tutoring workshops during the Artificial Intelligence in Education conferences in 2018 and 2019. Dr. Sinatra received her Ph.D. and M.A. in Applied Experimental and Human Factors Psychology, as well as her B.S. in Psychology from the University of Central Florida.

Authors

Daniel Auerbach is a Research Programmer at the University of Southern California Institute for Creative Technologies.

Dr. Ryan Shawn Baker is a tenured Associate Professor in the Graduate School of Education at the University of Pennsylvania where among other things he directs the Penn Center for Learning Analytics. His research is at the intersection of Educational Data Mining and Human-Computer Interaction. He develops and uses methods for mining the data that comes out of the interactions between students and educational software, in order to better understand how students respond to educational software, and how these responses impact their learning. He studies these issues within intelligent tutors, simulations, MOOCs/online courses, and educational games. He studies these issues in the context of K-12 formal and informal learning, higher education, the military, and lifelong learning.

Dr. Gautam Biswas holds an Endowed Chair as Cornelius Vanderbilt Professor of Computer Science and Computer Engineering, in the EECS Department and a Senior Research Scientist at the Institute for Software Integrated Systems (ISIS) at Vanderbilt University. Biswas conducts research in Intelligent Systems with primary interests in analysis of complex embedded systems, and their applications to diagnosis, prognosis, and fault-adaptive control. He is also involved in developing simulation-based environments for learning and instruction. This includes the Teachable Agents project, where students learn science by building causal models of natural processes. More recently, he has exploited the synergy between computational thinking ideas and STEM learning to develop systems that help students learn science and math concepts by building simulation models. He has also developed innovative educational data mining techniques for studying students' learning behaviors and linking them to metacognitive strategies. In all of his projects, there has been a strong emphasis on scaffolding students' learning of metacognitive strategies and preparing them for future learning. His work in machine learning and data mining is broadly applicable to a large number of projects that may require exploratory data analysis methods. His research has been supported by funding from the Army Research Labs, NASA, NSF, DARPA, and the US Department of Education.

Dr. Anthony Botelho received his PhD in Learning Sciences and Technologies from Worcester Polytechnic Institute (WPI) and is now a research scientist in the computer science department at WPI working with the ASSISTments online learning platform. Much of Dr. Botelho's research has focused on improving student learning by developing tools designed to help teachers in providing feedback that address gaps in student knowledge and increase engagement. Additionally, Dr. Botelho's work has focused on studying student knowledge, affect, and behavior to aid teachers in making better data-driven decisions in the classroom and support other researchers in designing interventions that promote positive learning strategies.

Dr. Peter Brusilovsky received his BS (1983) and PhD (1987) degrees from Moscow "Lomonosov" State University and his postdoctoral training at University of Sussex, University of Trier, and Carnegie Mellon University. He also holds a Doctor honoris causa degree from the Slovak University of Technology in Bratislava. Currently, he is a Professor of Information Science and Intelligent Systems at the University of Pittsburgh, where he directs Personalized Adaptive Web Systems (PAWS) lab. Peter Brusilovsky has been working in the field of adaptive educational systems, user modeling, and intelligent user interfaces for more than 30 years. He published numerous papers and edited several books on adaptive hypermedia, adaptive educational systems, user modeling, and the adaptive Web. Peter is the past Editor-in-Chief of IEEE Transactions on Learning Technologies and a board member of several journals including User Modeling and User Adapted Interaction, Journal on AI and Education, and ACM Transactions on Social Computing.

Zhiqiang Cai is a Research Assistant Professor with the Institute for Intelligent Systems at the University of Memphis. He has a M.S. degree in mathematics received in 1985 from Huazhong University of Science and Technology, P. R. China. After 15 years of teaching mathematics in colleges, he has worked in the field of natural language processing and intelligent systems. He is the chief software designer and developer of Coh-Metrix, OperationAries, CSAL AutoTutor and many other text analysis tools and conversational tutoring systems. He has co-authored over 100 publications**.**

Dr. Min Chi is an Assistant Professor in the Department of Computer Science at NC State University. She joined the department in August 2013 as a Chancellor's Faculty Excellence Program cluster hire in the Digital Transformation of Education. Dr. Chi earned her Ph.D. and M.S. in the Intelligent System Program from the University of Pittsburgh, and B.E. in the Information Science and Technology from Xi'an Jiaotong University, China. She was a Postdoctoral Fellow in the Machine Learning Department at Carnegie Mellon University, and Human Sciences and Technologies Advanced Research Institute at Stanford University. She received the Best Paper Award at the Intelligent Tutoring Systems Conference and Best Student Paper Award at the User Modeling, Adaptation, and Personalization Conference in 2010, and Best Student Paper Award at the Intelligent Tutoring Systems Conference in 2008. She also received Best Poster Award at the Educational Data Mining Conference in 2008. Her specialty is applied machine learning and data mining, and her research lies at the intersection of educational data mining and human-computer interaction. Dr. Chi research primarily focuses on applying machine learning and data mining methods to improve human learning and exploring new machine learning and data mining challenges posed by learning and social science.

Jody L. Cockroft, AA, BS, CCRP is a Research Specialist at the University of Memphis (UoM) in the Psychology Department with the Institute for Intelligent Systems where she has been for the past five-plus years. She has over thirty years of experience in scientific research and has worked on both the bench and on various clinical studies. She has been an integral part of the Army Research Laboratory Cooperative agreement, the Advanced Distributed Learning Initiative ADL-A project and the Advanced Learning Theories, Technologies and Impacts (ALTTAI) Consortium for the past several years while at the UoM. She has authored or co-authored over twenty articles in peer-reviewed journals as well as countless abstracts and posters. She is the treasurer of the Adaptive Instructional Systems (AIS) Working Group IEEE P2247.1.

Dr. Jeanine A. DeFalco, is an Adaptive Training Research Scientist and Post-Doctoral Research Fellow with the Army Research Laboratory, Human Research & Engineering Directorate Training Technology Office (RDRL-HRR) Orlando, Florida, working out of the United States Military Academy at West Point, NY. She received her PhD in Psychology from Columbia University, specializing in Human Development/Cognitive Studies in Education with a concentration in Intelligent Technologies. Jeanine's current research includes developing and testing pedagogical models for the Generalized Intelligent Framework for Tutoring to determine the relationship of creative and analogical reasoning in accelerated expert problem-solving in critical care medical education. Jeanine is also serving as a dissertation coach for the University of Pennsylvania's Graduate School of Education's PennCLO Executive Doctoral Program. Jeanine holds a Master's in Educational Theatre from New York University, and a Masters in Drama Studies from Johns Hopkins University.

Dr. Joseph Doty is the Executive Director of the Dr. John Feagin Leadership Program and Associate Director of the Leadership Education and Development (LEAD) curriculum at the Duke University School of Medicine. Prior to his work at Duke he was an ethics and leadership advisor in Singapore. Joe is a graduate of the U.S. Military Academy, West Point NY, and served in both educational and leadership positions during his 28 years in the Army. An avid reader and writer, Joe enjoys thinking, learning and helping others think and learn. He is an avid sports and current affairs fan, enjoys yard work, fitness and a good cup of coffee. Joe received Masters Degrees from Indiana University and the Naval War College and earned his PhD from the University of Northern Colorado. He has a passion for leadership, ethics, character development, mindfulness, and teaching. Some of the courses he taught at West Point include Leadership and Ethics in Sports.

Dr. Jeremiah T. Folsom-Kovarik is a Senior Scientist at Soar Technology, Inc. Dr. Folsom-Kovarik earned his Ph.D. in computer science from the University of Central Florida in 2012, where he added a training overlay to a call-for-fire simulation in use by the U.S. Marine Corps which demonstrated planning under uncertainty, tractable machine learning for policies at real-world scale, and improved outcomes in multiple measures. Dr. Folsom-Kovarik has contributed to the science of intelligent systems in adaptive user interfaces, cognitive ergonomics, human-machine teaming, and other machine learning challenges. One theme of this research has been making machine learning effective in challenging settings which lack large training data or require control interactions with nontechnical personnel. After the advent of deep learning and big data approaches, cognitive and semantic knowledge provides a powerful approach to directing and focusing machine learning algorithms and creating useful and usable artificial intelligence under real-world data constraints.

Niki Gitinabard is a Ph.D. Student at North Carolina State University, working with Dr. Collin F. Lynch. Her research is focused on modeling different student activities on online platforms associated with classes. Currently, she is working on extracting useful behavioral patterns from students' online activities and training prediction models on their performance through social network analysis, sequence mining, and the analysis of their teamwork habits.

Dr. Benjamin Goldberg is a member of the Learning in Intelligent Tutoring Environments (LITE) Lab at the U.S. Army CCDC Soldier Center in Orlando, FL. He has been conducting research in the Modeling & Simulation community for the past eight years with a focus on adaptive learning in simulation-based environments and how to leverage Artificial Intelligence tools and methods to create personalized learning experiences. Currently, he is the LITE Lab's lead scientist on instructional management research within adaptive training environments and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT). Dr. Goldberg is a Ph.D. graduate from the University of Central Florida in the program of Modeling & Simulation. His work has been published across several well-known conferences, with recent contributions to the Human Factors and Ergonomics Society (HFES), Artificial Intelligence in Education and Intelligent Tutoring Systems (ITS) proceedings. Dr. Goldberg has also recently contributed to the journal Computers in Human Behavior and the Journal of Cognitive Technology.

Dr. Andrew J. Hampton studied at the University of Central Florida and the Burnett Honors College on a full National Merit scholarship. After graduating with a B.S. in psychology and a minor in cognitive science, he entered the Wright State Human Factors/Industrial Organizational doctoral program in the fall of 2011 where he worked as a graduate research and teaching assistant, while also co-founding the Professional Development Group, a graduate student organization developed to encourage independent initiatives. While at Wright State, he earned his Master's degree in 2013, the program's Graduate Student Excellence Award in 2014, and his PhD in 2018. In 2016, he accepted a position as a researcher under Art Graesser at the Institute for Intelligent Systems, within the University of Memphis. There, as a Research Scientist Assistant Professor, he coordinated a major grant with the Office of Naval Research and functions as project manager for ElectronixTutor, a platform integrating several intelligent tutoring systems in a common interface. His work won the 2016 Human Factors Prize for Excellence in Human Factors/Ergonomics Research, focusing on Big Data analytics. Andrew's research interests include technologically mediated communication, psycholinguistics, intelligent tutoring systems, artificial intelligence, and political psychology.

Dr. Neil Heffernan is Professor of Computer Science and Director of the Learning Sciences and Technologies program at Worcester Polytechnic Institute. Before entering academia, Neil taught middle school math and science in the Teach for America program in Baltimore, where he met his wife Cristina. While completing his Ph.D. in Computer Science at Carnegie Mellon University, Neil decided to incorporate his passion for education and study EdTech. In 1997, Neil had a seizure and was told he had brain cancer and two years to live. This traumatic event helped Neil and Cristina learn what was important to them: making the world a better place. This helped motivate Dr. Heffernan to make this platform a free public service. Neil and Cristina created the ASSISTments platform as a free service that is used by 50,000 across the United States for daily classwork and nightly homework. In October, 2016 Dr. Heffernan was asked to present at the White House on the reproducibility crisis in educational research and the need for pre-registration and open-data. In December 2016, the Heffernans presented at the White House for a second time on the SRI evaluation that found ASSISTments doubled student learning). He has received national press from U.S. News, Scientific American, The New York Times, The Boston Globe and NPR. Dr. Heffernan has written 60+ papers on learning analytics and over two dozen papers on the results of randomized controlled trials.

Dr. Jong W. Kim was a postdoctoral researcher and adaptive tutoring scientist US Army Research Laboratory, Orlando, FL. Kim received his PhD in Industrial Engineering from Pennsylvania State University, and MS in Industrial Engineering from University of Central Florida. His research interests lie in the area of Cognitive Science and Adaptive Instructional Science. Particularly, Kim is interested in testing cognitive theories of learning (and forgetting) for the development of adaptive instructional systems. Previously, Kim has developed a theory of skill learning and forgetting (D2P: Declarative to Procedural) that is currently being applied to implement a series of intelligent tutoring systems for the Navy. After joining the GIFT team at ARL, Kim started to investigate an adaptive instructional system in a psychomotor domain that can be run beyond the desktop environment, pursuing maximizing training effectiveness in a simulated and range-based training.

Dr. James C. Lester is Distinguished University Professor of Computer Science and director of the Center for Educational Informatics at North Carolina State University. His research investigates AI technologies to support learning in education, training, and healthcare. His current work spans intelligent game-based learning environments, multimodal learning analytics, affective computing, and computer-supported collaborative learning. The adaptive learning environments he and his colleagues create have been used by thousands of students in K-12 classrooms around the globe. He is the author of more than two hundred publications, primarily in the area of AI technologies for education. His research program is supported by the National Science Foundation (NSF), the National Institutes of Health (NIH), the National Institute of Standards and Technologies (NIST), and the Army Research Laboratory (ARL). He has served as Editor-in-Chief of the International Journal of Artificial Intelligence in Education. He is the recipient of an NSF CAREER Award and the Best Paper Awards at the International Conference on Artificial Intelligence in Education, the ACM International Conference on Intelligent User Interfaces, the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, and the International Conference on User Modeling, Adaptation, and Personalization. His foundational work on pedagogical agents has been recognized with the IFAAMAS Influential Paper Award by the International Federation for Autonomous Agents and Multiagent Systems. He is a Fellow of the Association for the Advancement of Artificial Intelligence (AAAI).

Dr. Weiwen Leung is a postdoctoral fellow at the University of Toronto, where he moved after completing his PhD in Economics and MS in Computer Science at the University of Minnesota. His research agenda is to use behavioral economics to increase contributions to online communities.

Rodney Long is a Science and Technology Manager at the Army's Soldier Center, Simulation and Training Technology Center (STTC) in Orlando, Florida. He is currently conducting research in adaptive training technologies, including intelligent tutoring systems, architectures, authoring tools, etc. Mr. Long has a wide range of simulation and training experience that spans 32 years in the Department of Defense and has a Bachelor's Degree in Computer Engineering from the University of South Carolina and a Master's degree in Industrial Engineering from the University of Central Florida.

Dr. Collin F. Lynch is an Assistant Professor of Computer Science at North Carolina State University. His research is focused on the development of robust adaptive educational systems for Ill-Defined domains such as scientific writing, law, and software development. He also conducts research on social network analysis, graph mining, evolutionary computation, and augmented graph grammars. He completed his doctoral research in Intelligent Systems at the University of Pittsburgh where he was affiliated with the Learning Research and Development Center. His doctoral dissertation work focused on the use of argument diagrams to teach and diagnose student-produced arguments.

Dr. Noboru Matsuda is an Associate Professor in the Department of Computer Science at North Carolina State University. Noboru's research interests include need-based innovation of cutting-edge technologies to enhance learning as well as to advance theories of how people learn. He is particularly interested in artificial intelligence technologies enabling students to learn, teachers to teach, and researchers to understand how people learn and fail to learn. Dr. Matsuda's scholarly expertise spans computer science, cognitive science, learning science, and education. In recent years, Noboru has launched a new NSF funded research project on learning engineering where he studies how to efficiently build adaptive online courses and how to develop research on the sciences of learning using the adaptive online courses. This research has been extended to another NSF funded research project where the research team develops an infrastructure to build an adaptive online course that has multiple intelligent tutoring systems embedded in online courseware. The SimStudent technology is used to facilitate authoring intelligent tutors on the web-browser.

Christopher Meyer is the President and CEO of Synaptic Sparks. Prior to the creation of Synaptic Sparks, Chris worked hand-in-hand with representatives from the Departments of Defense, Health and Human Services, Energy, and Education to assist in the creation of solutions to solve challenges at a national level. He received his Bachelor and Master of Science degrees in computer science from Kansas State University, in addition to minors in modern languages and economics. Christopher brings international experience and his own engineering knowledge to Synaptic Sparks to inspire local, city, state, and federal involvement in the enhancement of STEM-based education for the next generation of America.

Dr. Benjamin Nye is the Director of Learning Science at the University of Southern California, Institute of Creative Technologies (USC-ICT). Dr. Nye's research has been recognized for excellence in intelligent tutoring systems (1rst Place ONR ITS STEM Grand Challenge; Nye, Windsor, et al., 2015; Nye et al., 2014), cognitive agents (BRIMS 2012 best paper; Nye & Silverman, 2013; Nye, 2012), and realistic behavior in training simulations (Federal Virtual Worlds Challenge; Silverman et al., 2012). His research is on scalable learning technologies (Nye et al., 2014) and design principles that promote learning (Nye, Graesser, & Hu, 2014; Nye, 2014; Nye, Morrison, & Samei, 2015). This research has led to 20 peer-reviewed papers, 11 book chapters, and 5 open-source projects.

Dr. Steven Ritter, also known as Steve, PhD co-founded Carnegie Learning, Inc. in 1998 and serves as its Chief Product Architect. Dr. Ritter served as Chief Scientist and Senior Cognitive Scientist of Carnegie Learning. Dr. Ritter has been developing and evaluating educational systems for over 10 years. Dr. Ritter is leading several research projects related to understanding and improving the way students learn mathematics, in partnership with the U.S. Department of Education, The Heinz Endowments, RAND Corporation, the National Science Foundation, Carnegie Mellon University, the University of Pittsburgh, and Southern University. Dr. Ritter serves as Director of Education Division Board for Software & Information Industry Association. Dr. Ritter is the co-author of Carnegie Learning's Algebra I curricula, one of only two math curricula recognized by the U.S. Department of Education as scientifically proven to have significant, positive effects on student learning. He is the author of numerous papers on the design, architecture and evaluation of educational technology and served as chairman of the IEEE Learning Technology Standards Committee working group on tool/agent communication. He is a recognized researcher and thought-leader in intelligent tutoring systems (ITSs) and data mining to improve education effectiveness. He is a postdoctoral associate and research scientist at Carnegie Mellon. He serves as a Member of the Advisory Board at Learning Objects, Inc. Dr. Ritter earned a B.Sc. in Cognitive Science from Brown University in 1985 and a Ph.D. in Cognitive Psychology at Carnegie Mellon University in 1992.

Dr. Jonathan Rowe is a Research Scientist in the Center for Educational Informatics at North Carolina State University and an Adjunct Assistant Professor in the Department of Computer Science. He received the Ph.D. and M.S. degrees in Computer Science from North Carolina State University, and the B.S. degree in Computer Science from Lafayette College. His research is in the areas of artificial intelligence and human-computer interaction for advanced learning technologies, with an emphasis on game-based learning environments, intelligent tutoring systems, user modeling, educational data mining, and computational models of interactive narrative generation.

Dr. Shitian Shen obtained his Ph.D. degree from Department of Computer Science at North Carolina State University. His research mainly focuses on the decision making and the pedagogical strategy induction in the Intelligent Tutoring System, and statistical data analysis on students' learning performance and behaviors. He is a member of the program committee of Educational Data Mining 2018, 2019.

Machi Shimmei is a PhD student in the Department of Computer Science at North Carolina State University. Her research focus is on the innovation and application of cutting-edge technologies to facilitate research on education and the science of learning.

Dr. Robert A. Sottilare is the Science Director for Intelligent Training at Soar Technology, Inc. He came to SoarTech in 2018 after completing a 35-year federal career in both Army and Navy training science and technology organizations. At the US Army Research Laboratory, he led the adaptive training science and technology program where the focus of his research was automated authoring, instructional management, and analysis tools and methods for intelligent tutoring systems (ITSs) and standards for adaptive instructional systems. He is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT), an open source, AI-based adaptive instructional architecture. GIFT has over 2000 users in 76 countries. Dr. Sottilare has long history as a leader, speaker, and supporter of learning and training sciences forums at the Defense & Homeland Security Simulation, HCII Augmented Cognition, and AI in Education conferences. He is the founding chair of the HCII Adaptive Instructional Systems (AIS) Conference. He is a member of the AI in Education Society, the Florida AI Research Society, the IEEE Computer Society and Standards Association, the National Defense Industry Association (lifetime member), and the National Training Systems Association. He is currently the IEEE Project 2247 working group chair for the development of standards and recommended practices for AISs. He is a faculty scholar and adjunct professor at the University of Central Florida where he teaches a graduate level course in ITS design. Dr. Sottilare has also been a frequent lecturer at the United States Military Academy (USMA) where he taught a senior level colloquium on adaptive training and ITS design. He has a long history of participation in international scientific fora including NATO and the Technical Cooperation Program. He has over 200 technical publications in the learning sciences field with over 1500 citations in the last 5 years. His doctorate is in Modeling & Simulation with a focus in Intelligent Systems from the University of Central Florida. Dr. Sottilare is a recipient of the US Army Meritorious Service Award (2018; 2nd highest civilian award), the US Army Achievement Medal for Civilian Service (2008; 5th highest civilian award), and two lifetime achievement awards in Modeling & Simulation: US Army RDECOM (2012; inaugural recipient) and National Training & Simulation Association (2015).

Nitin Surana is a software engineer, currently working at Google helping build payments. He had been a core polyglot developer at multiple early stage startups. His passion for programming can be traced to his childhood years when he wrote his first software to help his technologically novice father create invoices.

During his college years, he has won, organized & mentored various hackathons. He is a staunch supporter of Intellij against Eclipse and a beta user for Jetbrains products. Outside work, he likes running, hiking and adventure activities.

Raj Thaker is pursuing his Masters in Computer Science at the University of Southern California (USC). Currently, Raj is working with USC Institute for Creative Technologies with the Learning Science Department as a Student Research Programmer. Prior to this, he worked 2.5 years as a Software Engineer at JPMorgan Chase & Co. His strongest interests have been in developing full-stack applications with a focus on the Back-end infrastructure. Lately, Raj has been working around research in Artificial Intelligence and Machine Learning domains.

Richard Tong is the Chief Architect and General Manager of US Operations, Squirrel AI Learning by Yixue Group. He is an experienced technologist, executive, entrepreneur and one of the leading evangelists for standardization effort for global education. He was the Head of Implementation, Greater China Region for Knewton, and Director of Solution Architecture for Amplify Education. He also served as CTO of Phoenix New Media (NYSE:FENG). He has been heavily involved in education technology standardization in the last 8 years and on several LTSC initiatives in the last 3 years. He is a current member of the IEEE AIS (Adaptive Instructional Systems) Standard work group and chair for the 2247.2 Interoperability Subgroup, a member of the IEEE ICICLE (IC Industry Consortium on Learning Engineering) and a founding member of the IEEE Federated Machine Learning work group. He was a member of the Technology Board of Access For Learning Community (A4L), formerly known as the School Interoperability Framework Association (SIFA), and co-chair of the SIFA Assessment Group and IDM Group. He also served as a member of the Assessment Interoperability Framework Working Group for Common Education Data Standard (CEDS), and IMS Global Caliper standard workgroup and Computer Adaptive Testing workgroup, etc.

Dr. Joseph Jay Williams is an Assistant Professor in Computer Science at the University of Toronto focused on redesigning technology for people, with applications like education and health behavior change. He combines human-computer interaction and psychology by conducting randomized A/B comparisons in real-world settings, and applies statistics and machine learning methods like multi-armed bandit algorithms to dynamically adapt randomized A/B comparisons to enhance and personalize. He was previously an Assistant Professor at the National University of Singapore's School of Computing in the department of Information Systems & Analytics, a Research Fellow at Harvard's Office of the Vice Provost for Advances in Learning, and a member of the Intelligent Interactive Systems Group in Computer Science. He completed a postdoc at Stanford University in Summer 2014, working with the Office of the Vice Provost for Online Learning and the Open Learning Initiative. He received his PhD from UC Berkeley in Computational Cognitive Science, where he applied Bayesian statistics and machine learning to model how people learn and reason. He received his B.Sc. from University of Toronto in Cognitive Science, Artificial Intelligence and Mathematics, and is originally from Trinidad and Tobago. More information about his research and papers is at www.josephjaywilliams.com.

Lucy Woodman recently graduated from Seminole State College with a Bachelor of Science in Information Technology. Lucy has supported Synaptic Sparks (SSI) for one year during a successful internship and transitioned to be a major supporter of big data services within SSI in December of 2018. Lucy is a certified Amazon Web Service specialist and is currently studying to obtain further AWS certifications in System Architecting. Lucy also supports the team by providing research and development support in new fields of network and social technology.

INDEX

Design Recommendations for Intelligent Tutoring Systems

Volume 7
Self-Improving Systems

Design Recommendations for Intelligent Tutoring Systems (ITSs) explores the impact of intelligent tutoring system design on education and training. Specifically, this volume examines "Self-Improving Systems". The "Design Recommendations" book series examines tools and methods to reduce the time and skill required to develop Intelligent Tutoring Systems with the goal of improving the Generalized Intelligent Framework for Tutoring (GIFT). GIFT is a modular, service-oriented architecture developed to capture simplified authoring techniques, promote reuse and standardization of ITSs along with automated instructional techniques and effectiveness evaluation capabilities for adaptive tutoring tools and methods.

About the Editors:

- ***Dr. Anne M. Sinatra*** *is a research psychologist at the U.S. Army Combat Capabilities Development Command – Soldier Center – Simulation and Training Technology Center and works on the Generalized Intelligent Framework for Tutoring (GIFT).*
- ***Dr. Arthur C. Graesser*** *is a professor in the Department of Psychology and the Institute of Intelligent Systems at the University of Memphis and is a Senior Research Fellow in the Department of Education at the University of Oxford.*
- ***Dr. Xiangen Hu*** *is a professor in the Department of Psychology at The University of Memphis and visiting professor at Central China Normal University.*
- ***Dr. Keith Brawner*** *is a senior researcher at the U.S. Army Combat Capabilities Development Command – Soldier Center – Simulation and Training Technology Center,* and is a co-creator of GIFT.
- ***Dr. Vasile Rus*** *is a professor in the Department of Computer Science at The University of Memphis with a joint appointment in the Institute for Intelligent Systems.*

A Volume in the Adaptive Tutoring Series

9 780997 725773