Proceedings of the Eighth Annual GIFT Users Symposium

May 2020
Orlando, Florida

GIFT

Edited by:
Benjamin S. Goldberg

Part of the Adaptive Tutoring Series

Proceedings of the 8th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym8)

Edited by:
Benjamin Goldberg

Printed in the United States of America
First Printing, May 2020

U.S. Army Combat Capabilities Development Command
Soldier Center
Orlando, Florida

International Standard Book Number: 978-0-9977258-0-3

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

CONTENTS

FROM THE EDITOR

GIFT is a free, modular, open-source tutoring architecture that is being developed to capture best tutoring practices and support rapid authoring, reuse and interoperability of Intelligent Tutoring Systems (ITSs). The authoring tools have been designed to lower costs and entry skills needed to author ITSs and our research continues to seek and discover ways to enhance the adaptiveness of ITSs to support self-regulated learning (SRL).

This year marks the eighth year of the GIFT Symposia and we accepted 17 papers for publication. None of this could happen without the efforts of a fantastic team. Our program committee this year did an outstanding job organizing and reviewing, and we want to recognize them for their efforts.

- **Elyse Burmester**
- **Keith Brawner**
- **Jeanine DeFalco**
- **Greg Goodwin**
- **Michael Hoffman**
- **Anne Sinatra**
- **Joan Johnston**
- **Rodney Long**

We are proud of what we have been able to accomplish with the help of our user community. This is the eighth year we have been able to capture the research and development efforts related to the Generalized Intelligent Framework for Tutoring (GIFT) community which at the writing of these proceedings has well over 1600 users in over 78 countries.

These proceedings are intended to document the evolutions of GIFT as a tool for the authoring of intelligent tutoring systems (ITSs) and the evaluation of adaptive instructional tools and methods. Papers in this volume were selected with the following goals in mind:

- The candidate papers describe tools and methods that raise the level of knowledge and/or capability in the ITS research and development community
- The candidate papers describe research, features, or practical applications of GIFT
- The candidate papers expand ITSs into previously untapped domains
- The candidate papers build/expand models of automated instruction for individuals and/or teams

The editors wish to thank each of the authors for their efforts in the development of the ideas detailed in their papers. As a community we continue to move forward in solving some significant challenges in the ITS world.

GIFT and the GIFT Symposium will take on a broader perspective as the Army moves forward on modernizing their technology-driven training strategies through the Synthetic Training Environment and (STE) and Army Learning Ecosystem Concept (ALEC) 2035.

Finally, GIFT instructional videos will be available on YouTube this summer.

We would also like to encourage readers to follow GIFT news and publications at www.GIFTtutoring.org. In addition to our annual GIFTSym proceedings, GIFTtutoring.org also includes volumes of the Design Recommendations of Intelligent Tutoring Systems, technical reports, journal articles, and conference papers. GIFTtutoring.org also includes a users' forum to allow our community to provide feedback on GIFT and influence its future development.

Many thanks to all GIFT users…

Ben

Benjamin Goldberg, Ph.D.
GIFTSym8 Chair and Proceedings Editor

THEME I: GIFT UTILITY AND EXTENSIONS

The GIFT Architecture and Features Update: 2020 Edition

Benjamin Goldberg[1], Keith Brawner[1] and Michael Hoffman[2]
[1]U.S. Army Combat Capability Development Command – Soldier Center, [2]Dignitas Technologies, Inc.

INTRODUCTION

The first version of the Generalized Intelligent Framework for Tutoring (GIFT) was released to the public in May of 2012. One year later, the first symposium of the GIFT user community was held at the Artificial Intelligence in Education conference in Memphis, Tennessee. Since then, the GIFT development team has continued to gather feedback from the community regarding recommendations on how the GIFT project can continue to meet the needs of the user community and beyond. This current paper continues the conversation with the GIFT user community in regards to the architectural "behind the scenes" work and how the GIFT project is addressing the user requirements suggested in the previous GIFTSym7 proceedings. The development team takes comments within the symposium seriously, and this paper serves as an update on the requirement and feature requests from prior years.

As a follow up to the previous GIFT Symposium architecture updates (Brawner & Ososky, 2015; Ososky & Brawner, 2016; Brawner, Heylmun, & Hoffman, 2017; Brawner & Hoffman, 2018; Brawner, Hoffman, Nye & Meyer, 2019), this version highlights new tools and implemented feature requests accomplished over the latest development cycle. The feature requests and derived architectural improvements are derived from two primary sources: (1) symposium paper recommendations collected across the GIFT user base, and (2) stakeholder interactions linked to capability and project needs. The features are organized into logical sections within this update and cover modifications across all core modules operating within GIFT.

WELCOME

First, to those new to the GIFT Community – Welcome! There are a number of recommended resources that will help to orient you to this project and ecosystem. GIFT has come a long way since its original goals were defined in its description paper (Sottilare, Brawner, Goldberg, & Holden, 2012). First, we would encourage you to simply get started, as the tools and example courses have been designed to serve as baseline exemplars for exploration and familiarization purposes.

If you struggle with any individual aspect of the system, the team has produced short "how to" videos to try to help around the sticking points. These are available via the GIFT YouTube channel, which is the first result if you search "Generalized Intelligent Framework for Tutoring YouTube" on Google. The YouTube videos have not been updated for the new release, however, the vast majority of the GIFT challenges and authoring has remained unchanged.

Outside of the introductory materials and tutorials available in GIFT, there is also developer support through detailed documentation and active help forums. The GIFT user community is also invited to ask questions and share your experiences and feedback on our forums (https://gifttutoring.org/projects/gift/boards). The forums are actively monitored by a small team of developers, in addition to a series of Government project managers. The forums are a reliable way to interact with the development team and other members of the GIFT community. The forums, at the time of this writing, have over 1400 postings and responses. Documentation has been made freely available online at https://gifttutoring.org/projects/gift/wiki/Documentation, with interface control documentation https://gifttutoring.org/projects/gift/wiki/Interface_Control_Document_2020-1, and a developer guide https://gifttutoring.org/projects/gift/wiki/Developer_Guide_2020-1. These documents are updated each software release. In this release, we would also like to highlight the available instructions for hosting your own AWS instance (https://gifttutoring.org/projects/gift/wiki/Amazon_Web_Service_Install_Instructions).

GIFT Development and Release Strategy

GIFT, throughout its lifecycle, has followed 6 month, 9 month, or 12 month release cycles at various times. Currently, GIFT Cloud follows an every-Friday system update, with a 12 month release for the regression tested desktop version prior to GIFTSym each year. To support experimentation, intermittent extensions of the core GIFT baseline are performed to facilitate data and interaction requirements based on research questions at play. These are performed on a "as needed" basis, and often serve at the feature extensions included in the next public-release. In the upcoming cycle, there will be a heavy focus on applying and extending GIFT tools and methods to support team training and collaborative learning. Adjustments to the release strategy will be considered as more agile software development approaches are being applied at the organizational and enterprise level. As a member of the community, if you see a feature in the cloud release which you would like to use locally, simply ask.

GIFT Cloud General Reporting

GIFT Cloud (see Figure 1) has been running continuously for the last four years over Amazon Web Services. The cloud instance is kept online and updated in advance of the downloadable version, meaning that cloud content must be backwards-ported to be compatible with the perpetually out of date offline version. We do our best to keep the downloadable version to regularly scheduled improvements, but, for ordinary users, we would encourage you to use the Cloud version; it is better supported and more stable than the downloadable version. It supports hundreds of simultaneous users for experiments, with approximately 8 cloned cloud versions operating on different software configurations live at any given time. We are generally confident in the systems' ability to stay up and cope with demand. The current limitation is sensor-based interactions are not supported on the cloud instance, but that requirement will be addressed.

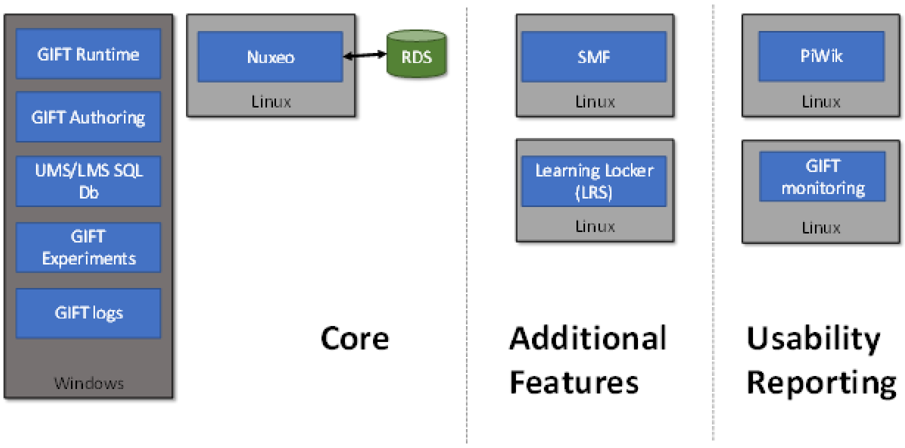


Figure 1. Simple Diagram Overview of GIFT Cloud Items

Behind the scenes, however, the re-tooling to move to a deployment version of dev-desk to dev-cloud to production has been working well. The team has greater ability to manage bug requests, with faster turnaround time. In this paper we reiterate that a clone of cloud.gifttutoring.org is always available upon request, but, however, we have not granted any requests this year. Instead, advanced users have copied clones of the AWS instances for individual usage. For the remainder of the paper, we will cover the latest improvements added over the last development cycle.

NEW GIFT FEATURES AND UPDATES

Since the last feature update from GIFTsym7 (Brawner, Hoffman, Nye & Meyer, 2019), there have been multiple additions to the GIFT capability set. Each tool or method described in this section is now available in the latest public-facing version of GIFT. Each new feature will be presented with information on the functions it supports and the system and data level dependencies to implement.

Team Tutoring

The first set of mature features to support team tutoring practices has been released. In the new version, the GIFT architecture now calculates performance assessment and maintains learner state for an entire (author defined) team organization. These team-level assessments are established within the Domain Knowledge File (DKF; a.k.a. Real-Time Assessment) and operate on interaction data produced during a training exercise. This means that the performance assessment and learner state represented in GIFT messages can now contain references to a team organization at the concept level. The team organization (see Figure 2) can be made up of playable team formations and roles (e.g. platoon sergeant), along with other objects that are important for assessment (e.g. vehicle, shop keeper). The GIFT architecture organizes all learners in the same session under a single GIFT learner called the host. It also maintains a mapping of which GIFT tutor user interface web page corresponds to each learner in order to deliver individualized feedback if necessary.

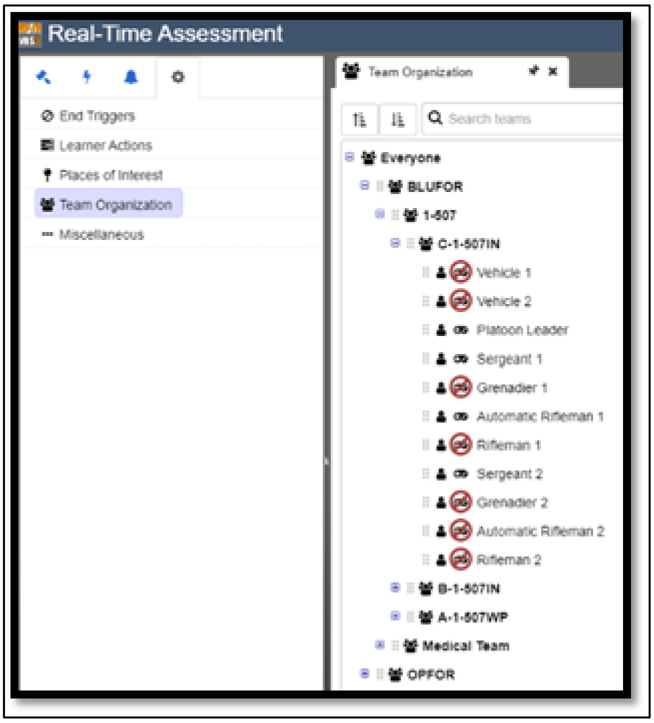


Figure 2. Team organization configuration pane in the GIFT Real-Time Assessment authoring interface.

The team organization panel, located from the "Assessment Properties" tab, is the best place to start when authoring team-level real time assessments. Here, you can define team organizational structures through hierarchical sub-team relationships (see Figure 2). These team structures can include OPFOR team layouts, as well as referenced non-playable but important objects. When adding non-playable entries in the team organization, those specific items are marked as not playable in the hierarchy. This is important when learners are responsible for selecting their role within the specified team structure presented within the Team Lobby prior to a scenario initializing (see Figure 3).

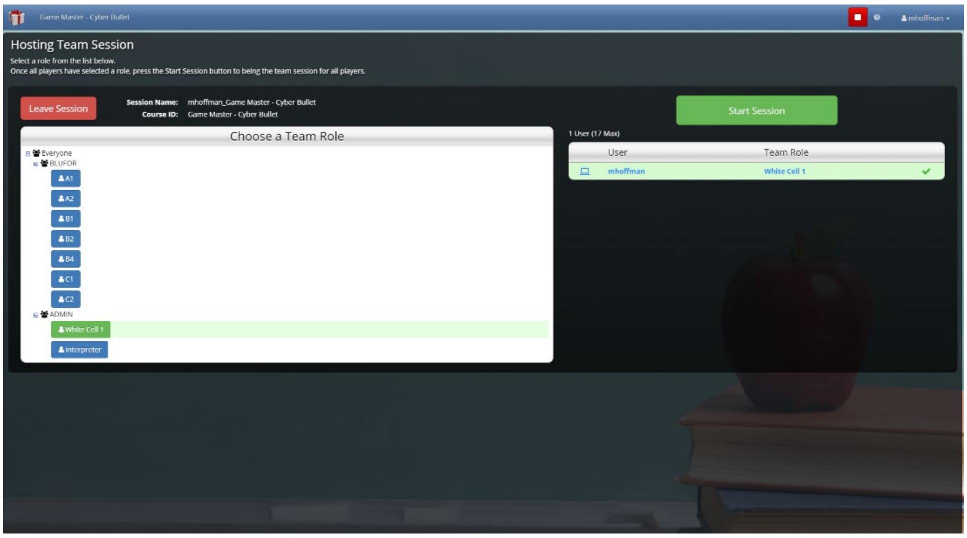


Figure 3. GIFT Team Lobby to assign users/trainees to a role within a define team organization.

Once the team organization is defined, you will be able to access the structure across several panels. You can also select which parts of the team organization should receive specific feedback messages. This allows authors to deliver feedback to one or more individuals instead of to the entire group. Another new feature is the ability to deliver instructions to learners when a task starts. This bypasses the learning effect chain of the GIFT architecture (i.e. no instructional strategies involved) to deliver domain specific information. Its intended purpose is not to deliver performance based feedback messages.

To initiate a team training exercise, a trainee lobby was instantiated. The lobby first allows users to host or join a team session. Users will only be able to host a session if the Gateway module is available and configured correctly. Users will only be able to join a team session if there are available roles (i.e. roles that aren't already taken) and the team session exists on the same GIFT instance/network. Learners have to choose which role they will be playing in the real time assessment. This allows GIFT to pair the GIFT user to the assessments tagged with that role and to the entity the player will be controlling in the training environment.

Human Observer Tool (Game Master)

The Game Master provides the ability to monitor, manipulate and playback events that take place during a real time assessment of a GIFT monitored scenario/exercise. This UI is designed to embed a human observer on GIFT's adaptive logic loop. The objective is to establish a collaboration model where an observer works with an intelligent tutor (e.g., add observation based assessments, push recommended injects into the environment, etc.) to optimize a training experiences. The current Game Master operates in two modes: (1) intelligent real-time Exercise Control (EXCON), and (2) AAR Playback Mode.

Intelligent Real-time EXCON

In this interaction mode, the Game master (see Figure 4) will display information about a current active session in real time. The observer has the option of simply monitoring the learner(s) activities, with tools established for adding observed assessments, adding bookmarks/notes, sending feedback to learners, manually triggering scenario adaptations, and managing strategy requests driven by GIFT.

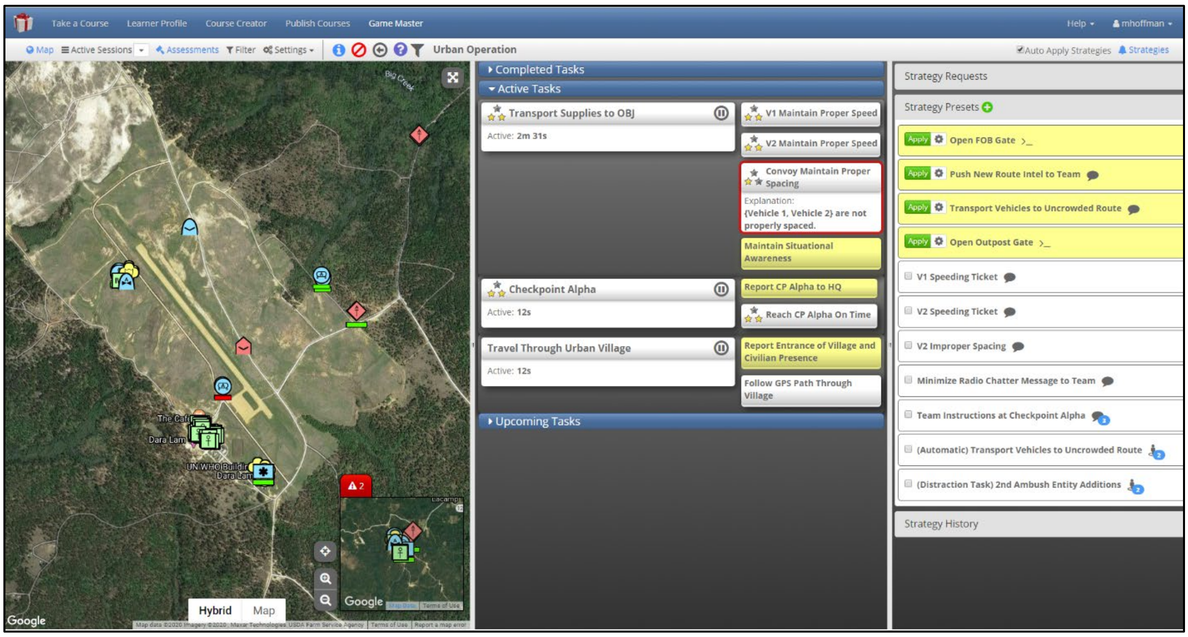


Figure 4. GIFT Game Master User Interface during scenario execution.

The assessment panel of the game master (see Figure 4) visualizes all tasks associated with a scenario and concepts assessed within each task. Each task has a life-cycle of not-activated yet, currently activated and no longer activated. This panel gives the observer a quick overview of the state of each task and descendant concepts including current assessment values. By adding a human to the loop, we can now extend the assessment space to include Subject Matter Expert (SME) inputs on concepts that lack mature data-driven methods to automate. To assist in this activity, yellow colored concepts displayed on the Game Master are those that will NOT be automatically assessed; they require a human observer to provide an assessment of the learner(s) performance when appropriate. Specific to bookmarking, a new available feature provides the option to annotate a bookmark with text or audio inputs across each concept represented in the DKF in real time. The audio component allows observers who don't have the time and/or means to type in text to quickly provide comments by simply verbalizing their observation. To use this feature you will need a microphone.

AAR Playback Mode

In AAR Playback Mode, Game Master users can playback a GIFT real-time assessment session for assessment verification and AAR purposes. Figure 5 shows an example of a past session playback experience with a new explorable timeline panel at the bottom. This new feature provides a temporal view of the scenario tasks and related performance assessments that were captured during a session. The user can manipulate the timeline to play, pause, and loop data visualization at any given moment. In addition, recorded bookmarks and observer assessments are marked on the timeline for easy navigation. Furthermore, a user can also decide to send game state messages out to external training applications for enhanced visualization functions (i.e., sending data to Augmented Reality Sandtable (ARES; Garneau, Boyce, Shorter, Vey & Amburn, 2018) for 3D Playback).

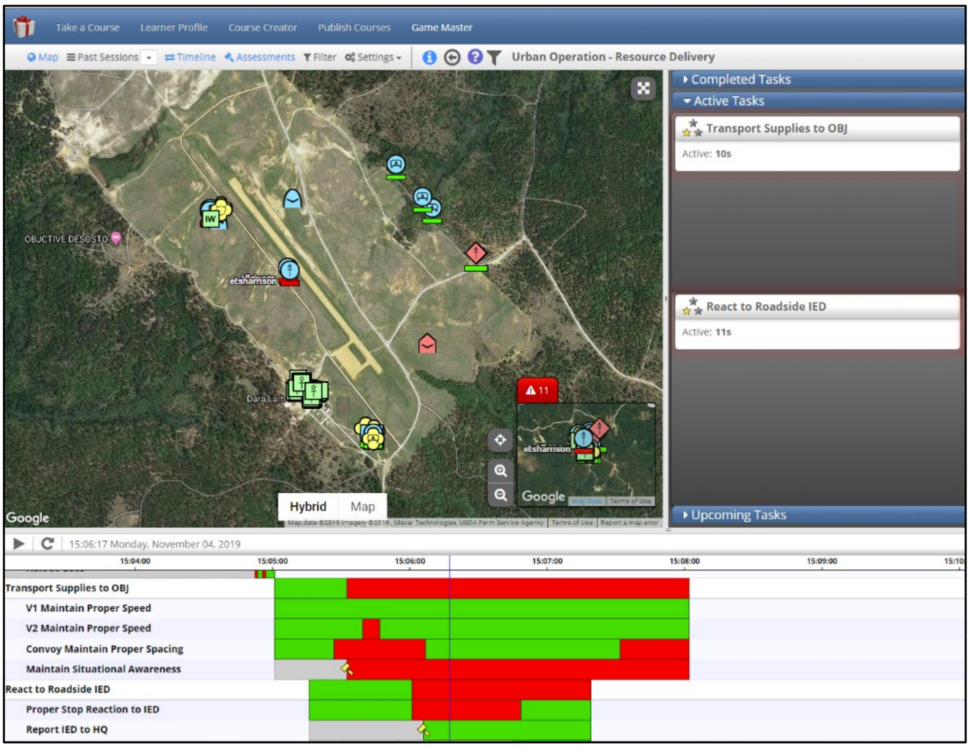


Figure 5. Game Master AAR Playback Mode with explorable timeline panel.

Training Applications and Extensions

Since the previous GIFTsym7 update, there have been multiple features developed that associate with specific training applications and capability extensions. These include: (1) integrating with the VR Engage training environment, (2) the new Interactive-Constructive-Active-Passive (ICAP) remediation models in GIFT's Engine for Management of Adaptive Pedagogy (EMAP) module (Rowe et al., 2016), (3) release of an exemplar COunter Insurgency (COIN; Spain, Rowe, Goldberg, Pokorny & Lester, 2019) lesson with the ICAP remediation model in use, and (4) a new game-state condition class that tracks muzzle flagging within a specified team organization.

VR Engage Training Environment

GIFT is now integrated with VR Engage (version 1.5 IPB3) created by VT MAK (www.mak.com; see Figure 6). From the course creator screen, there is now a VR Engage GIFT course object that is used to author real time assessments. Most of the same assessment techniques and supported scenario adaptations available for VBS are also available for VR Engage (e.g. create actor, change weather, teleport player, etc.). To allow bi-directional communication between the game engine and GIFT' Gateway module, a plug-in was developed for the VR Forces. Install and configuration instructions, along with further detailed information on the VR Forces plugin, is available in the GIFT Documentation and Release Notes for the most recent version.

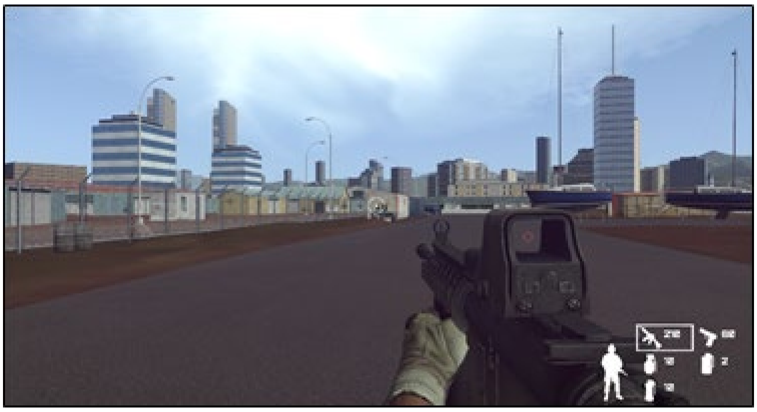


Figure 6. VR Engage Training Environment

ICAP Remediation and the COIN Chapter One Lesson

The ICAP pedagogical model was developed to support personalized remediation practices for individual learners. It was designed to apply Markov Decision Processes (MDP) and reinforcement learning techniques to establish and refine remediation policies that determine what learning concepts to target and what type of remediation content to deliver (i.e., based on interactivity levels; Rowe et al., 2016). When enabled, the course creator will allow authors to manage remediation material in a similar manner as can already be done with Adaptive courseflow course objects. Authors can also define the number of allowed attempts on the training application scenario and whether the scenario should be repeated or not after remediation is given. Remediation is only given when appropriate.

To support development and evaluation of the ICAP pedagogical approach, an exemplar course was created in the domain of COunter INsurgency (COIN) by leveraging instructional primer content used within the Institute for Creative Technology's (ICT) Bilateral Negotiation Trainer (a.k.a., BiLAT; Kim et al., 2009). This exemplar course was used to drive the initial build of ICAP policies through an experiment delivered within the Amazon Mechanical Turk platform (Spain et al., 2019). To provide the GIFT community with a use case of the ICAP in action, Chapter One of the aforementioned exemplar course is now available within the latest version of GIFT. This sub-set of the course will provide an excellent example to familiarize yourself with the remediation functions the ICAP provides, and the configuration workflows to setup the content and associated logic. It is worth noting that the current policies in place are 'Random', in that they randomly select remediation content when a concept or set of concepts is identified. Current research is establishing the first set of generalizable remediation policies that act on domain-agnostic data features that associate with student model variables and learning environment characteristics.

Muzzle Flagging Condition Class

A primary function of GIFT is to integrate with dynamic training environments and to establish assessment techniques that convert raw data into meaningful metrics for determining performance. With a primary focus on training and developing collective teams in these environments requires new condition classes that account for team dynamics and dependencies. One of the first new condition classes created examining team behavior at the tactical level is called Muzzle Flagging. When applied, this condition can be used to determine when a designated team member is observed aiming their weapon at another member within the team organization (see Figure 7 for graphical overview of condition logic). Authors can use this to assess when entities on the same side or opposing sides are facing each other.

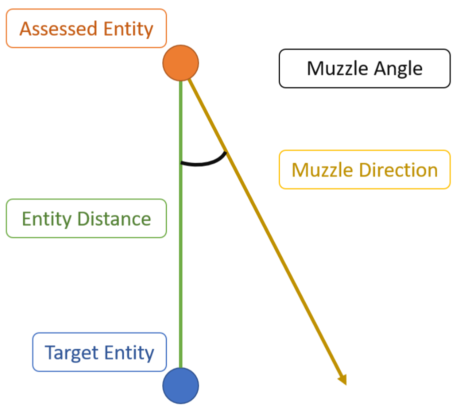


Figure 7. Muzzle Flagging condition class enables user to specify Muzzle Angle thresholds from a defined target entity (i.e., teammate).

Potpourri

In this final sub-section, we highlight the remaining miscellaneous feature updates included in the latest GIFT release. These include:

- Motivational Assessment Tool (MAT) Surveys
  - GIFT now has two more validated surveys for authors to include in their courses. These come from the MAT work that was conducted by IST. The goal for the MAT is to capture an individual's general motivation type and motivator preferences. GIFT converts the survey scores into learner state values which can be used to make decisions in the course (Reinermain-Jones, Lameier, Biddle & Boyce, 2017).
- GIFT Published Course Editing
  - In the past, when you created a published course the logic would make a copy of the source course. That copied version could not be edited post publishing. Now, when you create a new published course, a copy is not made and you have the option to edit the course after unlocking it.
- Structured Review in Adaptive Courseflow Object
  - In the prior version, the Structured Review page generated in TUI webpage was only shown when a learner failed a recall or practice phase. Now it is shown no matter pass or fail to give an opportunity for the learner to view the results before transitioning to the next activity.
- Other Feature Improvements
  - Better filtering of the types of overall scoring configurations a condition supports
  - Easy to author mid lesson surveys (i.e., less clicks and hunting)
  - Show images instead of file names when possible
  - Show embedded YouTube video instead of YouTube URL when possible

REQUESTED FEATURES FROM GIFTSYM7

GIFT is community-driven and we take pride in our user base. Especially as it relates to functions and processes requested to support their research and content delivery needs. From last year's symposium, several papers requested features, covering topics of authoring, session management, data collection, data processing, and data storage functions (Sinatra, 2019; Sottilare, Hoehn, Chen & Mostafavi, 2019).

Generally, these features fall into a few categories. The first of these considers user roles in GIFT and associated goals across roles when interacting with a system (i.e., student vs. instructor; Sinatra, 2019). Currently, GIFT supports experimenter and student roles. Experimenter permissions allow the users to assign items, make items available, collect data, and export it into a format which can be analyzed. Notably, GIFT lacks features that would be demanded in relatively common classroom settings, such as a gradebook, import of test questions from a publisher, and certified logins tracked to individuals. These were requested features which went unanswered this year. Instructor-driven efforts to streamline content selection and curation, requested elsewhere (Bell, Brawner, Robson, Brown & Kelsey, 2019) are available outside of the GIFT system, upon request.

Answered feature requests in regards to the teacher/experimenter division revolve around the ease of authoring and creation of an authoring dashboard (Sottilare, Hoehn, Chen & Mostafavi, 2019), as well as the creation of a condition class dashboard, which was recommended from multiple papers (Sottilare, Hoehn, Chen & Mostafavi, 2019; Davis, Riley & Goldberg, 2019). Condition classes and their authoring requirements when applied to dynamic virtual environments are designed to link to locations in either a real world location, virtual location, or mixed reality location as recommended from multiple papers (Davis, Riley & Goldberg, 2019; Mishra, Biswas, Mohammed & Goldberg, 2019), which have been developed into the current release. A simple dashboard of analysis and reporting of these items was also requested and answered - there are new dashboard for experiments as well as new visualizations for during-experiment items available in the current release. The population-level dashboard to run sample experimental populations, however, has not been constructed (Davis, Riley & Goldberg, 2019), and is left as a feature to future developments.

Other feature requests were for learner-specific activities, such as tracking and selecting learning activities as part of a course. For tracking users, the xAPI statements have been significantly expanded to the objects within a course, stored within an LRS, and merged with xAPI profiles (Hu, Cai, Graesser & Cockroft, 2019). Other authors requested bilateral xAPI data transfer (Aleven, Sewall, Andres, Popsecu, Sottilare, Long & Baker, 2019), which we perform at the level of both content and experimentation, although not affective state. A user seeking to log out affective state should use the included LearnSphere tool to synchronize affective state with simulation interactions and upload to LearnSphere. Learners are now able to select the types of remediation that they would like within the practice section of the Adaptive Courseflow object by default as requested (Matsada & Shummei, 2019), if further customization is needed, we suggest that you look at the implementation and modify it to suit individual needs.

Instructional-specific activities were requested, such as the ability to incorporate fixed models of feedback intervention (Hu, Cai, Graesser & Cockroft, 2019). These have been made available via AMI API calls in accordance with the ICD. Custom models, such as the model suggested in the requesting paper about Expectation Misconception Dialogue (EMD) can be configured through extensions of the existing course objects. I would urge the authors, or authors looking for similar functionality, to look at the customization of the existing Adaptive Courseflow object to suit their individual needs.

Experimental-specific activities were requested, so as the ability to author manual branches of content for experimental purposes (see Figure 8; Sinatra, 2019). Manual branches are part of the GIFT-2020-1 release, as shown in the below picture. Other requestors asked for the ability to use data imputation during data collection (Henderson, Rowe & Lester, 2019). It was determined that a priori modification of data inflow was outside of the scope of the program – but that the existing capabilities within RapidMiner, python XML-RPC, the Sensor and Learner Modules were sufficient to accomplish this goal. As an example, someone who wished to create a “assume that the last transmitted data point is the current state if a new data point has not arrived” logic can implement this directly into the code of the Sensor Module or into the filtering plugins. How an experimenter wants their data handled should be up to the individual experimenter, rather than a default such as “impute all data”. We did not enhance the experimental reporting functionality, as requested (Spain, Rowe, Goldberg, Pokorny Mott & Lester, 2019), although this is continuously on our list of near-term objectives.

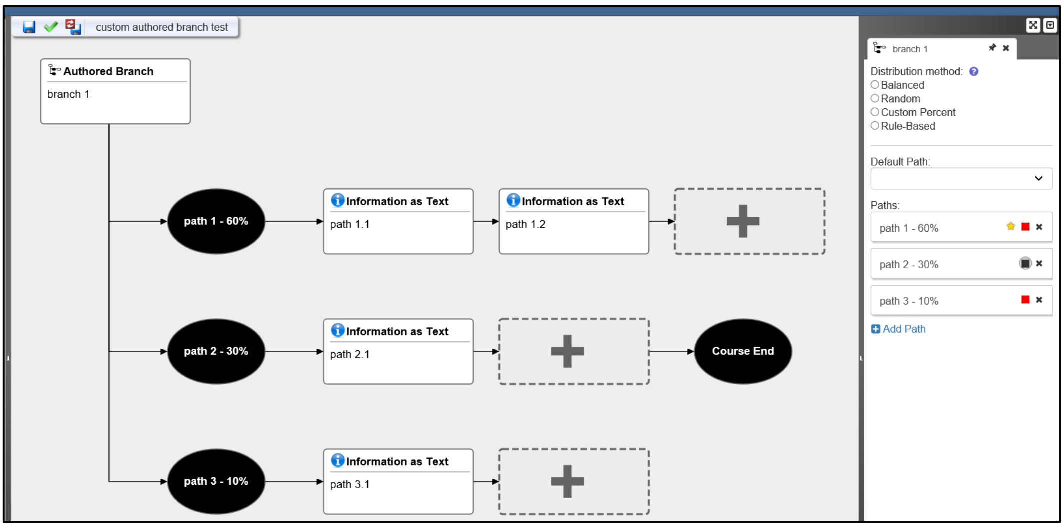


Figure 8. Conditional Branching for Controlled Experiments and Course Delivery

One author suggested the ability to integrate with a new military training simulator, which it appears that the military was going to use for a significant amount of its augmented reality training (Mishra, Biswas, Mohammed & Goldberg, 2019). We have unofficially integrated with this simulator, with the code available upon request, but it is not in the official release. In addition to this integration outside of the official release, we have integrated with the VT-MAK simulator, which, at the time of writing, is the choice of augmented reality training applications moving forward.

As part of the pivot towards team tutoring, a team tutoring reference architecture was requested. We are pleased to announce, discussed later in this work, that this has been a major undertaking and is in the current 2020-1 release (McCormack, Kilcullen, Sinatra, Case & Howard, 2019). If the reader is seeking an intelligent tutoring systems architecture for tutoring multiple team members on individual taskwork and teamwork simultaneously, the world's first system to do this is now released at gifttutoring.org. Features over top of a team tutoring architecture were requested (Swiecki, Ruis & Shaffer, 2019), such as Epistemic Network Analysis for the analysis of team tutoring interactions and visualization. These features have gone unanswered, but are on the developmental timeline now that the basic team interactions work.

GIFT AND IEEE STANDARDS ON ADAPTIVE INSTRUCTIONAL SYSTEMS

The IEEE Learning Technologies Standards Committee (LTSC), with support from the GIFT community and the Government, seeks involvement in standardization activities. The GIFT community invites the reader to join the conversation on what data exchange standards for adaptive instructional system technologies might look like in the future. There is an active IEEE community, to which the GIFT project is contributing meaningfully. Interested readers are encouraged to go to the IEEE LTSC meetings to become involved.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

The GIFT program has seen significant advancement since its conception in 2011. Each year, the community continues to build out new features and use cases that extend the boundaries of adaptive instructional systems. With a near-term focus on utilizing GIFT to address team tutoring challenges, we are excited to continue evolving the tools and methods to address critical capability gaps to drive future training requirements and system development. While the focus is on teams, it is well understood that the individual cannot be ignored. Stay tuned for continued improvements that address all facets of intelligent tutoring in today's education and training climate. Check back next year to see what kind of progress we're able to make!

REFERENCES

Aleven, V., Sewall, J., Andres, J, Popescu, O., Sottilare, R., Long, R. & Baker, R. (2019). Towards Deeper Integration of Intelligent Tutoring Systems: One-way Student Model Sharing between GIFT and CTAT. In Proceedings of the *7th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym7).* U.S. Combat Capability Development Command – Soldier Center, Orlando, FL.

Bell, B., Brawner, K., Robson, E., Brown, D. & Kelsey, E. (2019). Authoring Team Tutors in GIFT: An Automated Tool for Alignment of Content to Learning Objectives. In Proceedings of the *7th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym7).* U.S. Combat Capability Development Command – Soldier Center, Orlando, FL.

Brawner, K., & Hoffman, M. (2018). *Architecture and Ontology in the Generalized Intelligent Framework for Tutoring: 2018 Update.* Paper presented at the Proceedings of the 6th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym6).

Brawner, K., Heylmun, Z., & Hoffman, M. (2017). The GIFT 2017 Architecture Report. Paper presented at the Proceedings of the 5th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym5).

Brawner, K., & Ososky, S. (2015). The GIFT 2015 Report Card and the State of the Project. Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym3), Orlando, FL.

Davis, F., Riley, J. & Goldberg, B. (2019). Extending GIFT Wrap to Live Training. In Proceedings of the *7th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym7).* U.S. Combat Capability Development Command – Soldier Center, Orlando, FL.

Garneau, C., Boyce, M., Shorter, P., Vey, N. & Amburn. C. (2018). The Augmented Reality Sandtable (ARES) Research Strategy. Army Research Laboratory Technical Report (ARL-TN-0875). Aberdeen Proving Grounds, MD.

Henderson, N., Rowe, J. & Lester J. (2019). Multimodal Machine Learning in Adaptive Instructional Systems: A Survey. In Proceedings of the *7th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym7).* U.S. Combat Capability Development Command – Soldier Center, Orlando, FL.

Hu, X., Cai, Z., Graesser, A. & Cockroft, J. (2019). GIFT as a Framework for Self-Improvable Digital Resources in SIAIS. In Proceedings of the *7th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym7).* U.S. Combat Capability Development Command – Soldier Center, Orlando, FL.

Kim, J., Hill, R., Durlach, P., Lane, H. C., Forbell, E., Core, M., Stacy, M., Pynadath, D. & Hart, J. (2009). BiLAT: A Game-based Environment for Practicing Negotiation in a Cultural Context. *International Journal of Artificial Intelligence in Education (AIED)*, 19(*3)*, 289-308.

Matsada, N. & Shimmei, M. (2019). Application of Reinforcement Learning for Automated Contents Validation Towards Self-Improving Online Courseware. In Proceedings of the *7th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym7).* U.S. Combat Capability Development Command – Soldier Center, Orlando, FL.

McCormack, R., Kilcullen, T., Sinatra, A., Case, A. & Howard, D. (2019). Teamwork Training Architecture, Scenarios, and Measures in GIFT. In Proceedings of the *7th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym7).* U.S. Combat Capability Development Command – Soldier Center, Orlando, FL.

Mishra, S., Biswas, G., Mohammed, N. & Goldberg, B. (2019). Learner Modeling of Cognitive and Psychomotor Processes for Dismounted Battle Drills. In Proceedings of the *7th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym7).* U.S. Combat Capability Development Command – Soldier Center, Orlando, FL.

Ososky, S., & Brawner, K. (2016). The GIFT 2016 community report. Paper presented at the Proceedings of the 4th Annual GIFT Users Symposium.

Rowe, J., Frankosky, M., Mott, B., Lester, J., Pokorny, B., Peng, W. & Goldberg, B. (2016). Extending GIFT with a Reinforcement Learning-Based Framework for Generalized Tutorial Planning. In *Proceedings of the 4th Annual GIFT Users Symposium*, Orlando, FL.

Sinatra, A. (2019). The 2019 Instructor's Guide to GIFT. In Proceedings of the *7th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym7).* U.S. Combat Capability Development Command – Soldier Center, Orlando, FL.

Sottilare, R., Hoehn, R., Chen, D-W. & Mostafavi, B. (2019). Enhancing GIFT Authoring User Experience through Interaction Design. In Proceedings of the *7th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym7).* U.S. Combat Capability Development Command – Soldier Center, Orlando, FL.

Sottilare, R., Brawner, K. W., Goldberg, B. S., & Holden, H. A. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). US Army Research Laboratory. Aberdeen Proving Grounds, MD.

Spain, R., Rowe, J., Goldberg, B., Pokorny, B. & Lester, J. (2019). Enhancing Learning Outcomes Through Adaptive Remediation with GIFT. *In Proceedings of the 2019 Interservice/Industry Training Simulation and Education Conference (I/ITSEC)*. Orlando, FL.

Spain, R., Rowe, J., Goldberg, B., Pokorny, R., Mott, B. & Lester, J. (2019). Towards Data-driven Tutorial Planning for Counterinsurgency Training in GIFT: Preliminary Findings and Lessons Learned. In Proceedings of the *7th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym7).* U.S. Combat Capability Development Command – Soldier Center, Orlando, FL.

Swiecki, Z. Ruis, A. & Shaffer D. (2019). Modeling and Visualizing Team Performance using Epistemic Network Analysis. In Proceedings of the *7th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym7).* U.S. Combat Capability Development Command – Soldier Center, Orlando, FL.

VR Engage. (2020). Retrieved from the internet on 10 May 2020 from https://www.mak.com/products/simulate/vr-engage.

ABOUT THE AUTHORS

Dr. Benjamin Goldberg *is a senior research scientist at the U.S. Army Combat Capability Development Command – Soldier Center, and is co-creator of the Generalized Intelligent Framework for Tutoring (GIFT). Dr. Goldberg is the team lead for a research program focused on the development and evaluation of Training Management Tools for future Army training systems. His research is focused on the application of intelligent tutoring and artificial intelligence techniques to build adaptive training programs that improve performance and accelerate mastery and readiness. Dr. Goldberg has researched adaptive instructional systems for the last 12 years and has been published across several high-impact proceedings. He holds a Ph.D. in Modeling & Simulation from the University of Central Florida.*

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 14 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 the Combat Capabilities Development Command Soldier Center. He manages research in artificial intelligence for training with the Army's AI Task Force, the University of Southern California's Institute for Creative Technologies, and others.*

Michael Hoffman *is a senior software engineer at Dignitas Technologies and the technical lead for the GIFT project. He has been responsible for ensuring that the development of GIFT, meeting community requirements, and supporting production ITS systems, ITS research, and the growing user community. Michael manages and contributes support for the GIFT community through various mediums including the GIFT portal (www.GIFTTutoring.org), annual GIFT Symposium conferences and technical exchanges with CCDC and their contractors. In addition he utilizes his expertise in integrating third party capabilities such as software and hardware systems to enable other organizations to integrate GIFT into their training solutions.*

The 2020 Research Psychologist's Guide to GIFT

Anne M. Sinatra
U.S. Army Combat Capabilities Development Command (CCDC) – Soldier Center –
Simulation and Training Technology Center (STTC)

INTRODUCTION (DON'T PANIC)

Welcome to the 2020 Research Psychologist's Guide to conducting research with the Generalized Intelligent Framework for Tutoring (GIFT). The name of this series of guides is a reference to *The Hitchhiker's Guide to the Galaxy*, which has the words "Don't Panic" written in "large friendly letters" on the cover (Adams, 1979). In a reference to this, the current paper also says "Don't Panic" at the top, as at first when encountering GIFT, it can seem to be overwhelming. However, once you are familiar with GIFT, and the authoring tools, it is fairly straight forward to use. The guide has been through a number of different versions, with the first one in 2014 (Sinatra, 2014), which served as an introduction on conducting research in GIFT. At the time of the first publication, GIFT 4.0 and GIFT 2014-1X were current, and the authoring tools were primarily modified XML editors. The current version of the guide covers GIFT 2019-1, the upcoming GIFT 2020-1, and the associated Cloud version of GIFT. The subsequent versions of the Research Psychologist's Guide to GIFT were associated with GIFT 2015-2X (Sinatra, 2016), and GIFT 2017-1 (Sinatra, 2018). There has also been an additional series of guides which are written from the perspective of an instructor who would like to use GIFT (Sinatra, 2015; Sinatra, 2019). The goals of these guides were to provide an explanation of GIFT, how it could be used for a specific applied purpose, and to recommend additional features that would improve the experience of using GIFT for those specific purposes. While the overall context and general GIFT information in the previous guides are still valid, some of the specific processes and features of GIFT have changed over time. Therefore, the current paper serves as an introduction on how to conduct research in GIFT, and why a researcher might use it. It also highlights some of the changes that have occurred in more recent versions of GIFT, discusses some potential areas for improvement, and provides feature suggestions.

WHAT IS GIFT AND HOW DO I USE IT?

GIFT provides tools, and a platform that allows individuals to create and deliver intelligent tutoring (Sottilare, Brawner, Sinatra, & Johnston, 2017). The individual who is authoring the GIFT course does not need to have computer programming knowledge, but can interact with the already developed tools that are used to design the course. In GIFT, a course is a series of materials that are presented to the learner in a specific order called a courseflow. The course can include elements such as text, images, surveys, external training applications (e.g., PowerPoint, Virtual Battlespace 3), etc. GIFT was designed to not only be domain independent, but also to be able to be used for both creating intelligent tutors, and conducting research/experiments. While the tools in GIFT support research that specifically addresses the unique aspects of intelligent tutoring systems, they do not require that adaptation or a tutoring element be included in the GIFT course. Therefore, GIFT can be leveraged to create a traditional research experiment that does not include any adaptation or remediation. For instance, GIFT has been used for a number of different experiments, with a variety of research questions (Boyce, DeFalco, Davis, Kober & Goldberg, 2016; Goldberg, Ragusa & Chen, 2018; Sinatra, Sottilare & Sims, 2016) varying from the effectiveness of cognitive psychology principles in learning how to solve logic puzzles to marksmanship training. As the authoring tools and GIFT are highly generalizable, the experiments that have been conducted with GIFT include a number of different topics, mediums (e.g., computer based, psychomotor, etc.) and configurations.

Creating a Course/Using the Course Authoring Tool

When using GIFT for research, the first step is to create a GIFT course. A GIFT course is created using the Course Authoring Tool. The GIFT course will include all of the course objects that the researcher wants the participant to engage with during the interaction. The objects that can be used include images, slideshows, PowerPoints, authored branching, surveys, question banks, PDFs, etc. See Figure 1 for a screenshot of the course authoring tool. The left side of the screen has the different course objects that can be used, the center of the screen has the courseflow (the order that the objects will be experienced in), and the right side of the screen has details related to the item that is highlighted in the courseflow.

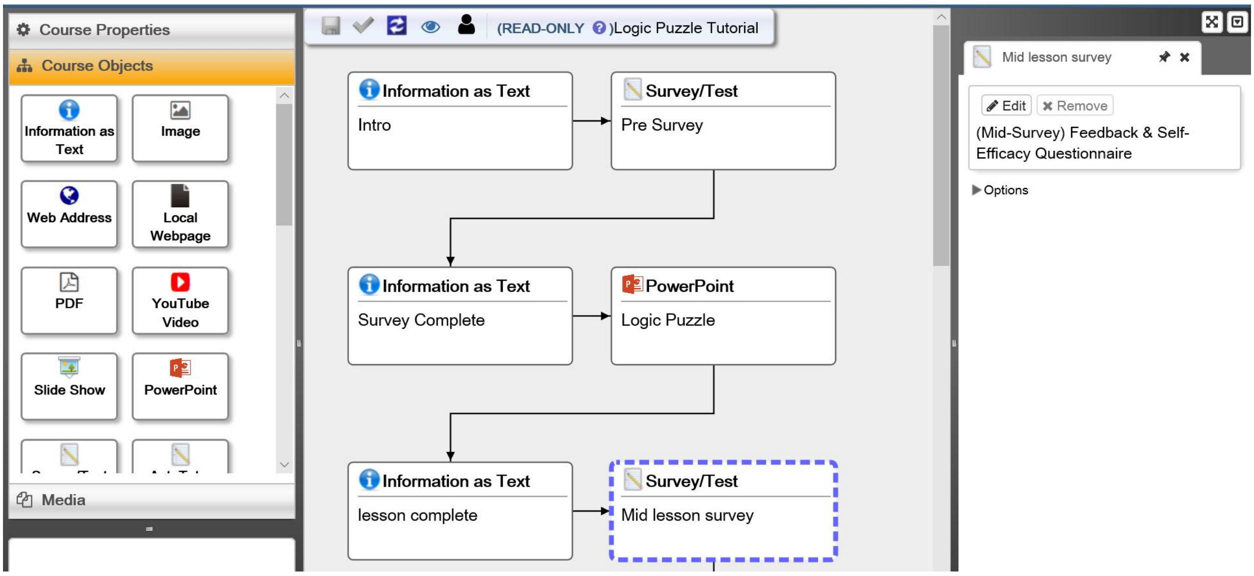


Figure 1. Example Screenshot of the GIFT Authoring Tool interface. The left side of the screen has the course objects, the middle of the screen has the courseflow, and the right side of the screen has the highlighted object's properties. In this example a Survey/Test object is selected.

In order to build a course, the items from the right side of the screen are dragged and dropped to the courseflow in the center of the screen. Once they are on the courseflow, they can also be reordered. As there are many different types of course objects, there is a great deal of flexibility in how one can construct a course for an experiment. For instance, information can be provided to the participant through the "Information as Text" course object, which will provide the author with a textbox to enter the information into. If they prefer, an author could provide similar information through an html page that they previous generated, a PDF, an image, or a slide show. Similarly, experimental materials can be provided to the participant using any of these options.

Slide Show Object vs. PowerPoint Object

One important note, which has been highlighted in previous guides is the distinction between the PowerPoint course object and the Slide Show course object. Unless there is a specific need to use the program PowerPoint (e.g., macros, interaction, or monitoring time in the presentation), it is recommended that the Slide Show object be used. If the PowerPoint course object is used, it requires that the participant have a compatible version of PowerPoint installed on his or her computer, and a connection to be made between GIFT and the instance of PowerPoint. In the case of the Slide Show object, it is created by the GIFT course author in PowerPoint, but it is converted to a series of images with an advance arrow when shown in GIFT. This method removes some of the barriers to using information constructed in PowerPoint in the Cloud version of GIFT, and reduces the possibility of user error when engaging with the course (with the Slide Show object nothing will need to be downloaded to the participant's computer). When creating materials intended to be used as Slide Show object in the PowerPoint program, it is important to save it as a .pps which is the type of file that will need to be uploaded to GIFT. This is done by selecting "PowerPoint 97 – 2003 Show" as the file type in the PowerPoint save menu.

Survey vs. Question Bank

An important component of many experiments is collecting data about the participant in the form of both demographics and answers to questionnaires. GIFT allows for course authors to create their own surveys and questions. Further, there is a function to import surveys that were previously created in Qualtrics. After these surveys have been imported, it is advised that they carefully be reviewed to ensure that everything ported over properly, and that there is nothing additional that needs to be added.

There is a distinction between the Survey course object and the Question Bank course object in GIFT. The Question Bank is used when you have many different questions that represent different concepts that you want to be presented to the participant in a randomized order. You will select the number of questions to present. For example, if you select that 4 questions should be shown, and you have 10 in the question bank, the participant will receive a subset of those 4 questions. The same 4 questions might not be presented to each participant. If you want to make sure that the same questions are received every single time, and in a specific order, you will use the Survey object instead. Once you select the Survey object, there are three different options. See Figure 2 for a screenshot of the Survey options.

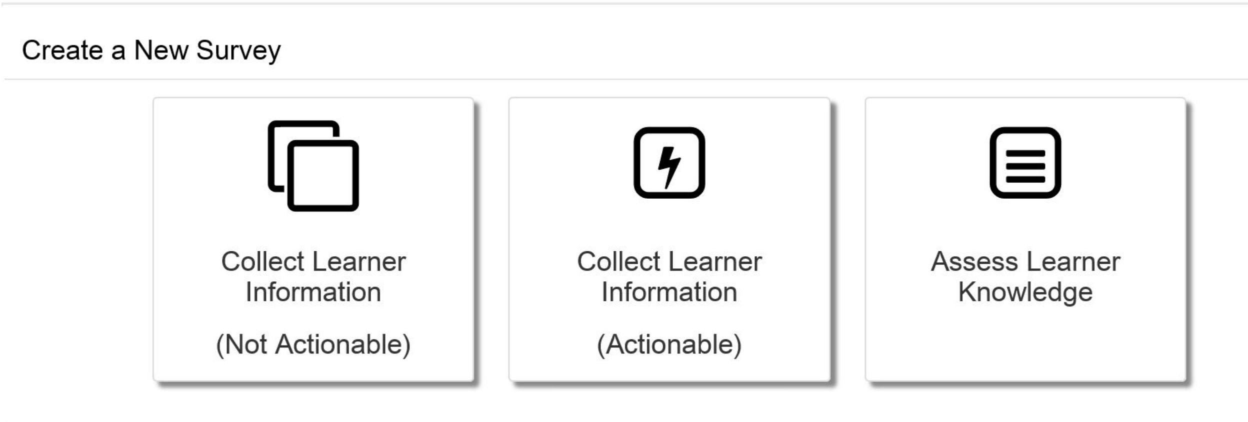


Figure 2. Screenshot of the Survey course object options.

For traditional experiments, the first option (Collect Learner Information [Not Actionable]), is what is most likely needed to be used (especially if adaptation is not intended to occur during the experiment). By selecting this option it will allow you to collect data about the participant, and always include the same questions in the same order. One of the restrictions of this type of survey is that you cannot provide correct answers using “Scoring Mode”, and it will require any grading to happen after the fact using the exported data. If you would prefer to have the system grade the questions for you, you can leverage one of the other two survey options. With these actionable survey options, you will have the ability to provide a correct and incorrect answer for the questions in the form of entering a number (for instance, you can use 1 for correct, and 0 for incorrect; if you want to leverage the system to provide coding for a questionnaire, you can put the appropriate number next to the response). The graded information should be provided to you in the extracted data output, even if you do not actively use it for adaptation during the GIFT course.

The survey authoring tool has gone through a design process to ensure that it is straight forward and easy to use. A screenshot of the survey authoring tool can be seen in Figure 3. There is a “Writing Mode” and a “Scoring Mode”. After you complete entering the question (for the actionable survey types) you can toggle to scoring mode by clicking on it on the top of the screen. In scoring mode you can indicate the correct answer, and assign points to the answer options. On the right side of the screen in the writing mode, there are fields that can be filled in. One of particular importance is called “Tag”. The information that is added into the “Tag” field will be linked to the output data so that it is clear to the researcher which question it was associated with. This was of great importance in previous versions of GIFT since only the question number without any text used to be provided at the top of the exported data column. As of more recent versions of GIFT, the full question text is now included in the title of the exported data column. This is a very important improvement as it ensures that the researcher is able to match up the answers that were provided with the questions that were asked.

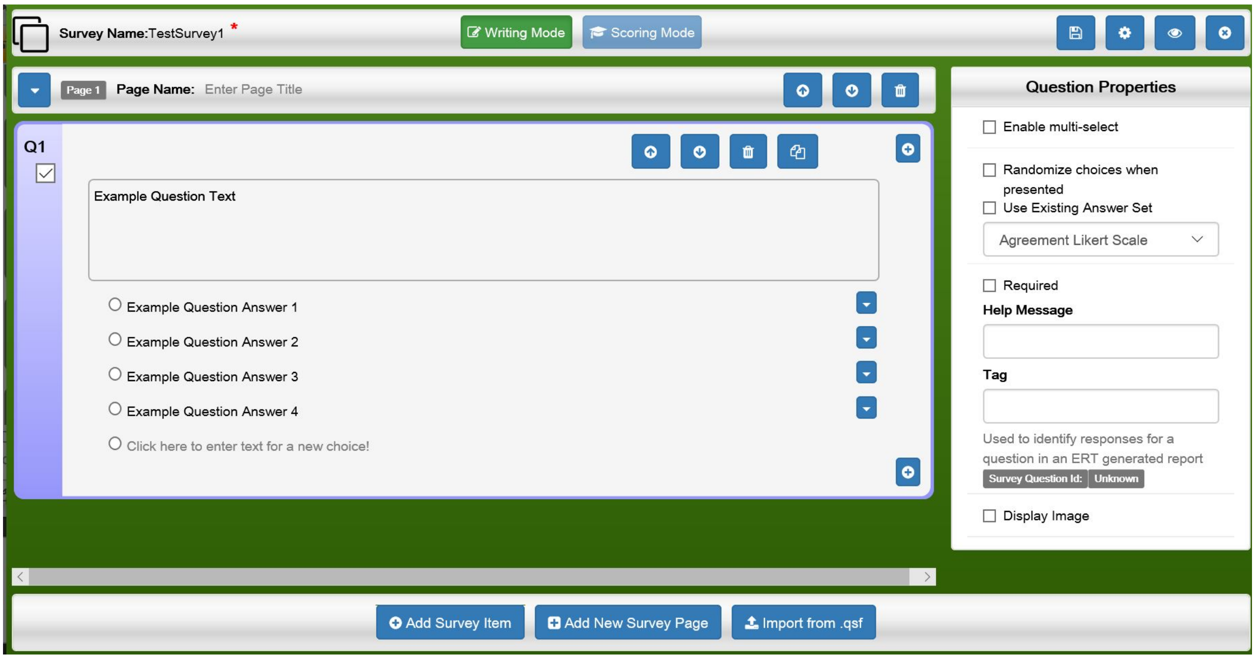


Figure 3. Screenshot of Question Authoring interface for the Non-Actionable Survey Type (note that the Scoring Mode button is grayed out).

IMPORTANT DECISIONS

When creating an experiment in GIFT there are a few important decisions that need to be made. Among them are whether the Cloud or Desktop version of GIFT will be used. Further, the design of the experiment itself can either be created with multiple courses or one single course utilizing the new Authored Branch course object. Additionally, it is important to carefully consider the login method that will be used when participants interact with the GIFT system.

Cloud vs. Desktop

There are two main ways of accessing GIFT, either on the Cloud (https://cloud.gifttutoring.org) or on the Desktop (downloadable at https://gifttutoring.org). Depending on the goals of the experiment, it may be advantageous to use one version over another.

Data extraction and logs are an important consideration when using GIFT for an experiment. There are two different methods for taking an authored course in GIFT: through using a published experiment link, or through being logged into GIFT and using your account. If you wish to have participants login to the system and use a GIFT account, then it is advantageous to use the desktop version, as you will have easy access to the participant logs. If you use the cloud version and the login method, you will not have manual access to the logs. If you choose to use the publish experiment link version, it will not be linked to an account or name, but you will have the ability to pause the data collection and export the data.

Another consideration is if you will have external training applications such as PowerPoint or Virtual Battlespace 3 that the participant is interacting with. Currently, VBS3 is not traditionally used with the Cloud version of GIFT, as too much data is being passed between GIFT and the training application. Sensors are not used with the Cloud version of GIFT for the same reason. Further, if you are using the Cloud version of GIFT with a training application, there is an extra step that needs to occur before a course can be run: the gateway module needs to be downloaded and run on the participant's computer. The gateway module serves as the link between GIFT and the external training application. If you want to use PowerPoint and the Cloud version of GIFT, the computer that is accessing GIFT will need to have a GIFT compatible version of PowerPoint installed on it. The individual on the computer will need to download and run the gateway module when prompted, so that GIFT can communicate with it. This creates an extra step, and potential point of failure on the part of the participant. Therefore, if you are not using any of the unique features of PowerPoint (e.g., videos, animations, macros), then it is best to use the Slide Show object when authoring a course in GIFT, as it will not require a connection be made to the individual's computer in order to run the course.

If you want to run a fully online study, it can be done using the Cloud version of GIFT. You can publish the course link and then provide it through the recruiting medium (e.g., SONA systems; Amazon Mechanical Turk). If this is done, you will need to make sure to provide information about what their participant number is for credit or payment purposes so that it can be linked back to the other system. This has been successfully done in a few different cases, and examples of the process have been documented on the GIFT Forums. See Table 1 for a diagram that provides explanations of why you might use one version over another.

Table 1. Examples of when it is preferred to use the Cloud and Desktop Versions of GIFT

Requirement	Cloud	Desktop
Easy Access to Saved Logs regardless of login method		X
Easily Interacts with External Training Applications		X
Only uses Survey Based responses and images/text for content	X	
Uses sensors		X
Participants can engage with the experiment at any day or time on their own computers	X	

Published Experiment vs. GIFT Course

An additional decision that needs to be made about how the experiment will be run. The participant can login using the GIFT course, or the finished course can be "Published". Both the desktop and cloud versions of GIFT provide both of these interaction methods. The way that the participant logs in to GIFT will be different based on the approach that is used. An example of the two different types of login experiences are shown in Figures 4 and 5.

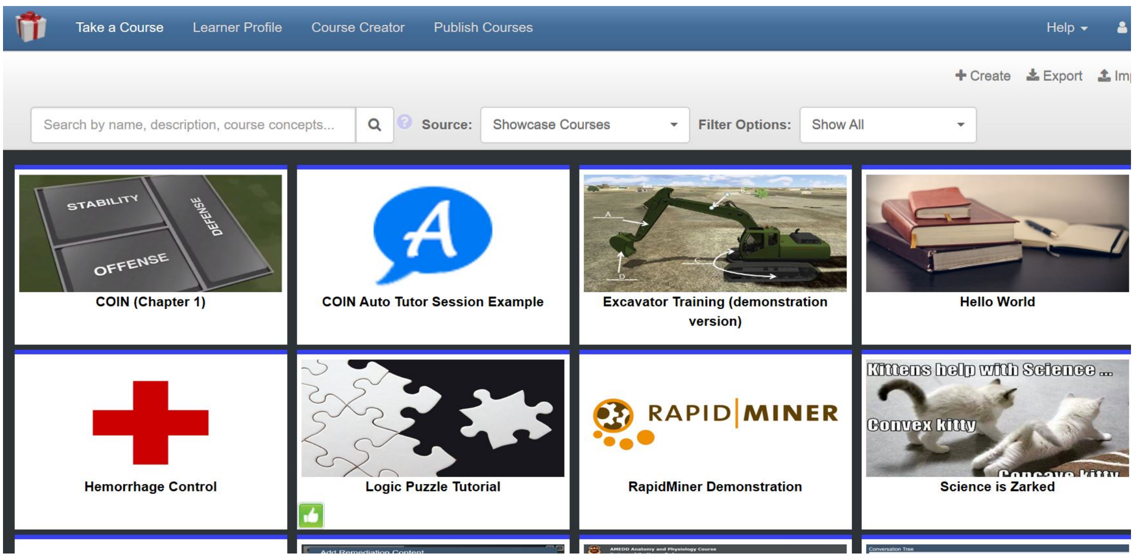


Figure 4. Login Screen with GIFT account

Figure 5. Login Screen for Published Course

Experiment Courseflow and Authored Branching

One update that has occurred since the prior versions of GIFT is that there are now two possible methods for creating an experimental path in GIFT.

The original strategy was to create the experiment courseflow using GIFT, and saving the course. The experimentally manipulated item would then be changed within a copy of the course. This would be repeated for each separate experimental condition. For instance, if there were three different experiment conditions, there would be three different GIFT courses. Looking at Figure 1, imagine that the manipulation is occurring during the PowerPoint course object. In this case there would then be three courses which are identical except for the PowerPoint that is being opened at that specific part of the course. When using this method, the experimenter would need to track the specific URL or course that should be provided to the participant, and would then need to recombine the data from each experimental condition for analysis. One way to make sure that the condition was clear to the experimenter would be to also change something in the survey names or tags in order to reduce the possibility of confusion about which condition the participant was in.

There is now a second option available, as a new Authored Branching course object is now in GIFT. If the Authored Branching course object is used then a very similar course authoring strategy is employed. However, instead of creating multiple versions of the course, an Authored Branching object would be added at the branch off point. For instance, looking at Figure 1, if the manipulation in the course occurred as a different PowerPoint, instead of having the PowerPoint object, the author would add the Authored Branching object where it would have been, which would direct the participant to one of three different PowerPoints. The course author has the option of ending the course after the branch has been completed, or continuing back to the main courseflow. In an experiment, the most frequently used approach is to let the interaction continue back to the main courseflow so that the rest of the experiment can be completed.

The strategy of using the Authored Branching object instead of multiple courses for the experiment has both pros and cons. A benefit of this is that three identical courses do not need to be created. Additionally, there is no need to determine which link to provide to individuals for different conditions, since all the conditions exist within one GIFT course. The cons of it are that the control is less visible for the experimenter; there is more reliance on the system to correctly distribute participants between the versions of the courses. Further, the researcher needs to be very careful when looking at the data outputs to determine which path the participant was sent down and that there is an accurate way to represent the experimental condition within the data output. See Figure 6 for a screenshot of the distinction between different distribution methods using the Authored Branching object, and Figure 7 for an example of the Authored Branching course object.

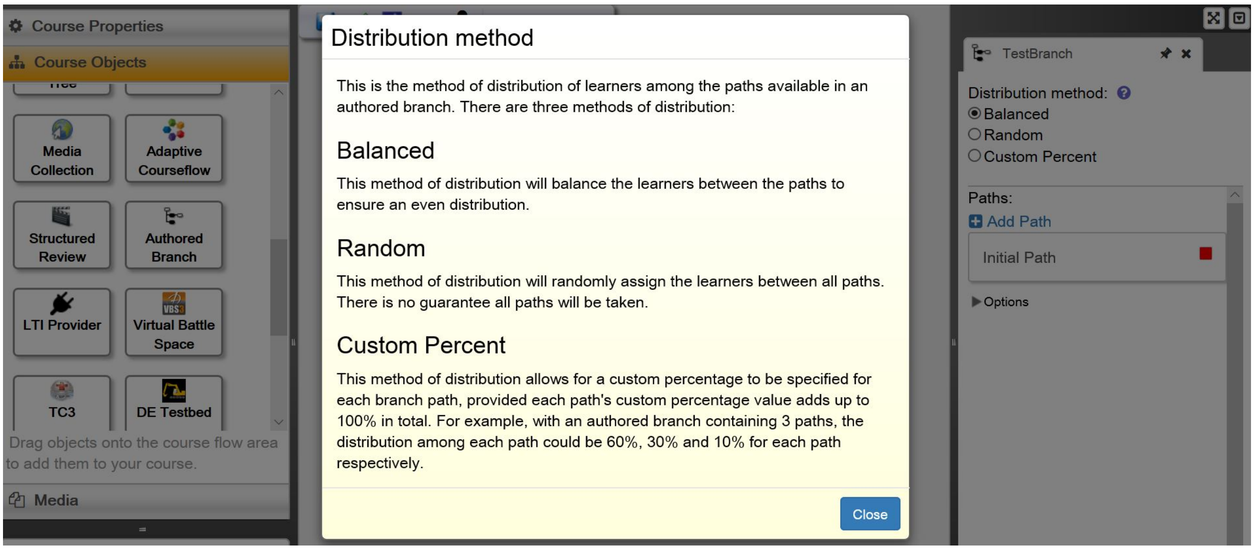


Figure 6. Screenshot of the Authored Branching Object, and the different participant distributions that are possible.

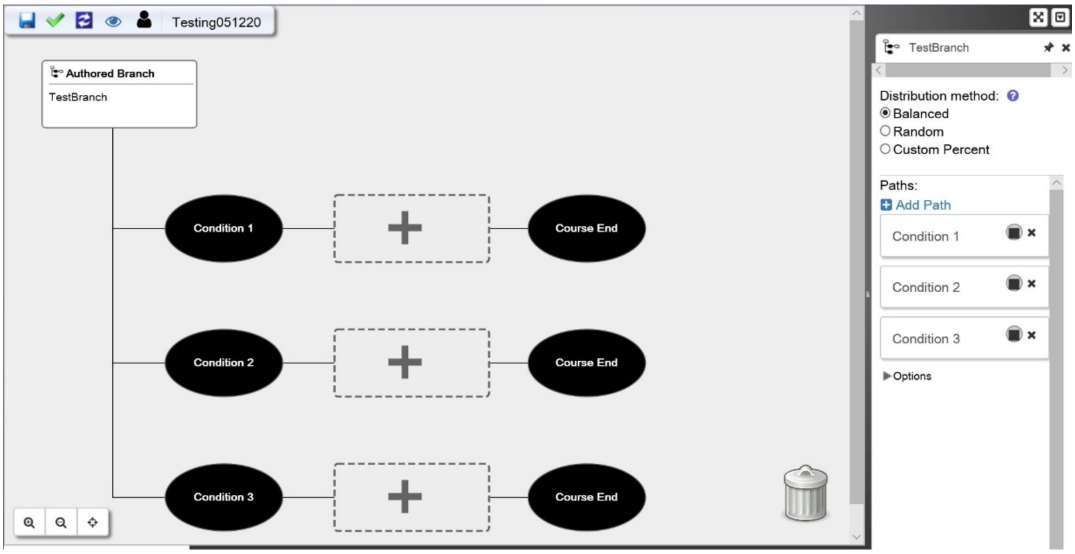


Figure 7. Screenshot of the Authored Branching Object, and the paths that are created when additional branches are added.

RUNNING YOUR EXPERIMENT IN GIFT

To simplify the discussion, in the current section we will discuss how you would run an experiment using the "Publish Course" approach. Once you are happy with the course that you are created and want to share it with participants, you will click "Publish Course" on the GIFT interface. It will then display an interface that can be seen in Figure 8. You will select the course that you want, and it will be published. Figure 9 shows the interface after the course has been published. The URL that can be provided to the participant is available to the researcher. Additionally, the interface tells you how many responses the course has, and gives you the option of pausing it, and downloading the data that has been collected. When using this publish course method, when participants go to the URL they will be greeted by the interface in Figure 5. They will click start and be entered into the GIFT course. It is very important that you provide the participant with their participant number, and that there is a question within the course that asks them for that number. This will ensure that you know the participant number, which will be especially important if you are matching up the data with anything else that they have previously filled out.

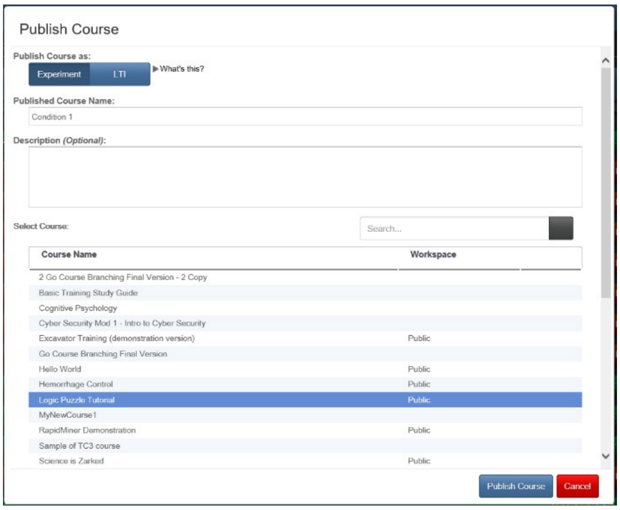


Figure 8. The Publish Course interface.

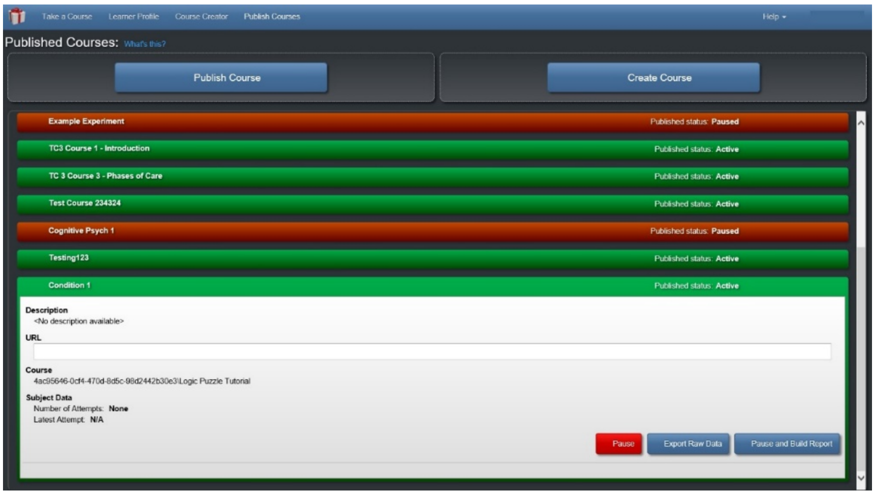


Figure 9. Published course interface.

As a note, there has been a change in recent versions of GIFT such that if you edit the original GIFT course that was created, those changes will now occur within the published version of the course. In previous versions of GIFT the published course was a copy at the original moment in time when it was published. Therefore, it is very important to be mindful of any changes that you make to the original course, as they will now populate to your published experiment course.

After all the participants have engaged in the study, you can pause the experiment to make the link inaccessible. You can reactivate it if you wish to in the future. Additionally, you can “Pause and Build Report” in order to download the data that has been collected to date. You will need to reactivate the experiment afterward if you want additional participants to respond. Figure 10 shows the “Build Report” interface. For a traditional experiment, the information that will be most important to you is “Survey Responses”. Additionally, if you want to easily be able to put your data into SPSS, you should check the box that says “Merge participant’s events into a single row”. This will result in data from each individual participant being on a single row, and each column will represent a question that was asked. The extracted data will be a .csv file which can be opened and edited in Excel, and then later imported into SPSS for analysis.

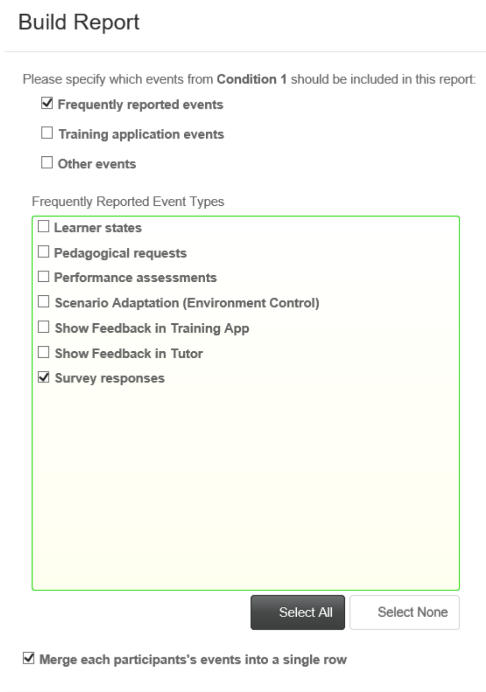


Figure 10. Build a report interface for extracting student data.

UPDATES TO GIFT THAT ARE RELEVANT FOR EXPERIMENTERS

In summary, there are a number of changes and improvements that have been implemented in GIFT since previous versions of this Guide. They include:

- The full text of the question is now exported in the question name during survey data extraction.
- There is the ability to create an authored branch and keep your entire experiment within one course.
- There is now a link between the original course file and the published experiment (it is no longer a copy). If you update something in the original course file, it will now instantaneously change in the published version that you are using for your experiment. It is important to be mindful to this, and to understand that when you merge question data it may look a little bit different.

POTENTIAL IMPROVEMENTS THAT CAN BE MADE TO GIFT

It is easy to extract experiment data when using the "Published Course" functionality. However, if you have participants login to their GIFT account to take the course instead, data extraction is difficult. Further, if you use the method of having participants login with the online version of the GIFT, there is no way for the experimenter to extract the data themselves (it would require contacting the GIFT research team). Similarly, if this method of interaction is used on the desktop there are no recent/updated data tools available to extract the data. The previous process (which has been documented in earlier guides) is still used, where the Event Report Tool (ERT) needs to be opened, and the logs need to be present in the correct folder in order to be read in. In order to this it will require the researcher to click through the GIFT folders that are installed on their computer and find the appropriate items to run. It would be advantageous for an additional up to date tool to be available such that information that was entered when the participants were logged in could be easily extracted and examined by experimenters.

It may also be beneficial to provide an example experiment Public Course in GIFT. This could include an introduction (information as text object), survey objects, and an authored branching object that represents multiple different conditions. This course could serve as an example for researchers, who could then run through it themselves, and practice going through the process of extracting data. It could also include an example survey that asks for the participant number, to highlight to researchers that this will be an important component of their course. This example course could also serve as an initial template that could be copied, and then edited to include the content that the experimenters wish to use if they would like to run a simple experiment.

CONCLUSION

As GIFT continues to be developed and moves toward including team tutoring functionality, it opens up additional opportunities for conducting experiments. The existing tools and functionality are being extended to be used by more than one individual at a time, and this could provide the ability to investigate interesting industrial/organizational psychology questions. The current guide provides an overview of the features and functionality of GIFT at present time, and how it can be used to support research. In the future, GIFT will continue to be improved, and continue to be a great tool for researchers that are conducting experiments.

REFERENCES

Adams, D. (1979). *The hitchhiker's guide to the galaxy*. New York: Pocket Books.

Boyce, M. W., DeFalco, J. A., Davis, R. C., Kober, E. K., & Goldberg, B. (2018, July). Mitigating Skill Decay in Military Instruction and Enemy Analysis via GIFT. In International Conference on Augmented Cognition (pp. 171-182). Springer, Cham.

Goldberg, B., Amburn, C., Ragusa, C., & Chen, D. W. (2018). Modeling expert behavior in support of an adaptive psychomotor training environment: A marksmanship use case. International Journal of Artificial Intelligence in Education, 28(2), 194-224.

Sinatra, A. M. (2014). The research psychologist's guide to GIFT. In Proceedings of the 2nd Annual GIFT Users Symposium (pp. 85-92).

Sinatra, A. M. (2015, August). The Instructor's Guide to GIFT: Recommendations for using GIFT In and Out of the Classroom. In Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym3) (p. 149).

Sinatra, A. M. (2016). The Updated Research Psychologist's Guide to GIFT. In Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym4) (p. 135).

Sinatra, A. M. (2018, May). The 2018 Research Psychologist's Guide to GIFT. In Proceedings of the Sixth Annual GIFT Users Symposium (Vol. 6, p. 259). US Army Research Laboratory.

Sinatra, A. M., Sottilare, R. A., & Sims, V. K. (2016, September). The effects of self-reference and context personalization on task performance during adaptive instruction. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting (Vol. 60, No. 1, pp. 398-402). Sage CA: Los Angeles, CA: SAGE Publications.

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.

ACKNOWLEDGEMENTS

The research described herein has been sponsored by the U.S Army Combat Capabilities Development Command – Soldier Center – Simulation and Training Technology Center. The statements and opinions expressed in this article do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

ABOUT THE AUTHOR

Dr. Anne M. Sinatra *is part of the adaptive training research team within CCDC Soldier Center STTC, and works on the Generalized Intelligent Framework for Tutoring (GIFT) project. Her background is in Human Factors and Cognitive Psychology.*

Theme II: Learner Modeling

An Ontology for Motor Skill Acquisition Designed for GIFT

Laurențiu-Marian Neagu[1], Vincent Guarnieri[1], Eric Rigaud[1], Sébastien Travadel[1]
Mihai Dascălu[2] and Răzvan-Victor Rughiniș[2]
MINES ParisTech, PSL - Research University, CRC, France
{laurentiu.neagu, vincent.guarnieri, eric.rigaud, sebastien.travadel}@mines-paristech.fr[1],
Computer Science Department, University Politehnica of Bucharest,
{mihai.dascalu, razvan.rughinis}@upb.ro[2]

INTRODUCTION

The North Atlantic Treaty Organization (NATO) Training Group's (NTG) demonstrated substantial efficiency in using computer technologies with both reduced costs and enhanced effectiveness while investigating on the nature, extent, availability, and feasibility of Intelligent Tutoring Systems (ITSs) in education and training, (Paviotti, Rossi, & Zarka, 2012; Sottilare, Brawner, & Sinatra, 2017). The usage of ITS in well-defined domains was argued to be as effective as expert human tutors in well-defined domains, such as mathematics or physics, while even surpassing traditional classroom training environments (Sottilare et al., 2016). Even though research in this field dates for more than half a century, these systems are not used at scale, either in education or in other fields (Sottilare et al., 2016). Primary reasons include their high development costs, limited reuse, a lack of standards, and poor adaptability to the learner's needs. However, new frameworks such as the Generalized Intelligent Framework for Tutoring (GIFT) partly overcome these shortcomings.

Our aim is to develop an ITS based-on GIFT, centered on motor skills acquisition and injury prevention when practicing daily physical activities either at professional, sportive, or recreational levels. A systematic literature review conducted on the usages of ITS in the psychomotor domain by Neagu, Rigaud, Travadel, Dascalu, and Rughinis (in press) has shown that such systems were used in several domains, from medicine (e.g., surgery, radiology) to military (e.g., training marksmanship), but there is no previous work focused on motor skills acquisition in sports for general health.

Ontologies are a formal, explicit description of concepts and relations from a domain, having properties assigned to concepts and describing various features and attributes, together with potential restrictions (Noy & McGuinness, 2001). Ontologies are one of the most frequently employed and powerful approaches for domain, student, and tutoring modeling in Intelligent Tutoring Systems (Nkambou, Bourdeau, & Mizoguchi, 2010). This paper presents the initial results of an ongoing process of designing a motor skill acquisition ontology.

We begin with the description of the context, the domain, the scope, and the structure of the proposed ontology. Afterwards, existing ontologies integrated in our work are introduced, followed by the presentation of essential terms from the ontology, together with our hierarchy of classes. The lessons from the first validation conclude the presentation of our work.

RELATED WORK

Ontologies tend to be used everywhere nowadays, being perceived as a general solution for many applications, such as peer-to-peer systems, database integrations, e-commerce, or semantic web services (Euzenat & Shvaiko, 2007). An ontology typically provides a vocabulary describing a domain and a specification of the meaning of terms in that vocabulary. The incorporation of various specifications for modeling web resources and their metadata through syntax notations and data serialization formats is accomplished using Resource Description Framework[1] (RDF). Through RDF, data can be handled and queried using SPARQL (W3C, 2013) in knowledge representation systems. Data is represented as triples with < subject, predicate, object >, enabling the creation of knowledge graphs, i.e. RDF graphs. A predicate can be seen as the relation between a subject (a node in the graph) and an object (another node in the graph, or a value). The standard for defining ontologies is Ontology Web Language (OWL) (McGuinness & Van Harmelen, 2004), which allows the definition of specific classes, subclasses, domains and ranges of relations, as well as constraints and axioms.

One of the most common tools for constructing large electronic knowledge bases is Protégé (Noy et al., 2003), a software that allows developers to create and edit domain ontologies through direct manipulation. Our ontology was developed through both Protégé desktop and WebProtégé[2] (a more straightforward web-based solution). As Protégé is an open-source, component-based solution, several plugins were developed to enhance the system's capabilities. ProtégéVOWL–VOWL Plugin for Protégé for ontology visualizations– and WebVOWL[3] (used in the current work, the web-based version of this plugin) can be used for visualizing the ontology. The latter works independently of Protégé and is better maintained than the Protégé plugin.

Our approach uses Ontology Development 101 (OD101) (Noy & McGuinness, 2001) as methodology. In line with OD101, an ontology includes the following key constituents: a) *Classes*, also known as concepts; b) *Properties* of classes, describing features and attributes of classes, also known as slots or roles; and c) *Restrictions* on slots, also known as facets or role restrictions. Other existing methodologies for ontology development were defined, for example: Methontology (Fernández-López, Gómez-Pérez, & Juristo, 1997), OnToKnowledge, NeON, OntoSpec, DiDOn, and Melting Point methodologies (Khan & Keet, 2012). OD101 is considered a 'micro-level' methodology, which focuses on guidelines to formalize the subject domain.

Motor Skill Acquisition and GIFT

Learning a movement pattern is a complex process (Button, 2021; Newell, 1985). First, learners have to elaborate on a suitable coordination pattern by assembling the appropriate relative motions among relevant body parts (e.g., legs, hips, trunk, and arms). Second, they have to gain a tighter fit between the assembled coordinative structure and the performance environment. Afterward, they have to be able to exploit environmental information sources to optimize the coordinative structure while enhancing efficiency and control. Various elements influence motor skill acquisition, namely: factors within the individual (neuromotor maturation, rate of growth, readiness, etc.), factors in the environment (parent to infant attachment, stimulation, and deprivation, etc.), and mechanical and physical factors (strength, endurance, speed, coordination, flexibility, etc.).

GIFT is particularly suited to motor skill acquisition, a process in which a performer learns to control and integrate posture, locomotion, and muscle activations, all enabling individuals to engage in a variety of motor behaviors that are constrained by a range of task requirements (e.g., athletic context) (Newell, 1991). The necessity to consider learners' characteristics such as prior experience, genetic attributes, anthropometry, or focus of attention when designing, applying, and monitoring psychomotor development programs motivates the development of a GIFT (Newell, 1985).

[1] https://www.w3.org/RDF
[2] https://protege.stanford.edu/
[3] http://www.visualdataweb.de/

GIFT Ontology Integration

There are several recent works that enable GIFT architectures to be more ontology-driven (Brawner, Hoffman, & Nye, 2019); in tight relation, the last GIFT Symposiums showed that developers are shifting towards this approach. In particular, GIFT offers a set of XML-based configuration tools for enhancing authoring capabilities and usability. Sottilare (2012) proposes five core authoring processes for building an ontology for GIFT: 1) Authoring user/learner model; 2) Authoring domain-specific knowledge; 3) Authoring instructional strategies; 4) Authoring user-tutor interfaces; and 5) Integrating tutor components.

ONTOLOGY DESIGN AND DEVELOPMENT

Our ontology development process used a simple knowledge-engineering methodology, as described in Ontology 101, which included the following phases: (1) determine the domain and scope of the ontology, (2) consider reusing existing ontologies, (3) enumerate essential terms in the ontology, (4) define the classes and the class hierarchy, (5) define the properties of classes – slots, (6) define the facets of the slots, and (7) create instances.

Step 1. Determine the domain and scope of the ontology

The design of a GIFT dedicated to the training of human movement skills motivates the definition of a psychomotor domain ontology. The psychomotor domain is related to the processes of change, stabilization, and regression in physical structure and neuromuscular function (Goodway, Ozmun, & Gallahue, 2019). Its vocabulary focuses on the human motor development and sportive performance development domains. The ontology supports the learning process by providing answers to requests associated with the following essential tasks:

- *Definition of trainee learning objectives*. Before starting the learning process, GIFT defines the learning objectives by interacting with the trainee. Trainees indicate their physical activities (e.g., daily life, leisure, sports, professional), which they aim to efficiently and safely perform. GIFT generates the list of movement skills that constitute the objectives of the learning process. The ontology supports this process by answering the following requests: What are the physical activities supported by GIFT? For a given activity, what are the fundamental and the specialized movement skills the trainee needs to learn?

- *Trainee initial evaluation*. GIFT evaluates the trainee readiness to perform movement skills level when starting the learning process. GIFT requires trainees to accomplish a set of test exercises, analyzes their performance, and deduces their level. The ontology supports this process by answering the following queries: For a given movement skill, what are the corresponding readiness requirements? For a given readiness requirement, what are the corresponding tests supporting its assessment? For a given test, what are its associated assessment level and performance criteria?

- *Training program definition*. GIFT defines training objectives by integrating the readiness to perform requirements based on the trainee goals and necessary movement skills. Then, GIFT designs an initial training strategy by combining relevant training modalities. The ontology supports this process by answering the following requests: For a given movement skill, what are the required qualities and the minimum performance level? What are the associated training modalities for a given quality?

- *Applying training program*. Finally, GIFT generates training workouts, analyses their effects on trainee performances, and adapts accordingly to the training strategy until the end of the program (objective achieved or duration completed). The ontology supports this process by answering the following requests: what are the workouts required to achieve a training method?

Ontology supports the description of movement skills and associated assessment and development modalities. The next sections describe the essential vocabulary formalized by the ontology.

Step 2. Consider reusing existing ontologies

The design of an ontology of the psychomotor domain relies on existing taxonomies and ontologies. Bloom and his colleagues (1956) distinguished the cognitive, affective, and psychomotor domains when designing the first taxonomy of educational goals. They did not detail the psychomotor domain, except that it includes physical movement, coordination, and use of the motor-skill areas. Later, Dave (1970), Harrow (1972), and Simpson (1972) proposed taxonomies for the psychomotor domain. They distinguished several types of movements (imitation, manipulation, precision, reflex, fundamental, etc.).

Ontologies developed for supporting motion recognition in the artificial vision domain provide classes which can be integrated in our ontology. Ma and Kevitt (2004) proposed an ontology of verbs describing human motion. Video Movement Ontology (VMO) (Saad, Mahmoudi, & Manneback, 2012) provides classes describing movements. The VMO ontology used the Benesh Movement Notation (BMN). This notation (Benesh & Benesh, 1983) uses a five-line horizontal stave (similar to music notation), to form a suitable basis or matrix for the human figure. The Kinect ontology (Diaz-Rodriguez, Wikstrom, Lilius, Cuellar, & Flores, 2013) provides classes describing bones and joint hierarchy and classes describing human activities and behavior.

Step 3. Enumerate important terms in the ontology

Four main entities compose the ontology of the psychomotor domain (see Figure 1), briefly presented below:

- *Movement Skill* - Variables supporting assessment and development of a movement pattern: Readiness to Perform the Requirements, Leveling System, Skill Requirements to perform the Movement;

- *Movement Pattern* - Variables describing form, accuracy, and control in the performance of change in the position of any part of the body: Goals, Rules, Complexity, Equipment, etc.;

- *Psychomotor Profile* - Variables describing psychomotor properties: Weight, Body Mass Index (BMI), Low Body Size, Sagittal Balance, Maximal Speed, etc.;

- *Training Program* - Variables describing the different activities proposed to enhance the individual psychomotor profile to acquire movement skills: Objectives, Training Period, Performance Factor, etc.

Classes are the core component of most ontologies, as they describe the concepts in the represented domain. From the hierarchical class grouping, a Movement Skill is a Thing (base class, superclass of all classes in Protégé), and has the following derived classes: Fundamental Movement Skill and Specialized Movement Skill. Each derived class has more specialized subclasses; the full class hierarchy is presented in **Step 4**.

The ontology does not and should not contain all information existing on the psychomotor domain; thus, we considered specializations and generalizations for the most representative classes. Also, we did not represent all possible properties for the classes, but just the key attributes of each particular concept.

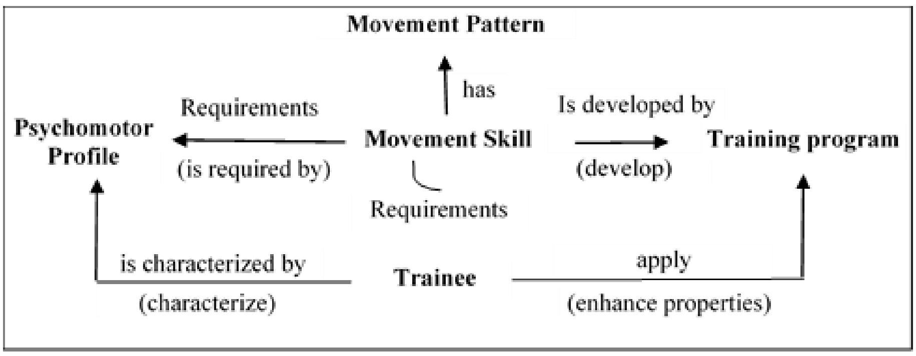


Figure 1. Main Entities of the Motor Skill Acquisition Domain Ontology.

We have chosen a naming convention for classes and slots, and adhered to it, as described in the ontology development guide. Protégé is case-sensitive and maintains a single namespace for all frames (not possible to have a class and a slot with the same name). Therefore, we relied on the classic naming rules described in the guide: a) capitalize class names; b) introduce spaces between words, together with the capitalization of each new word, when a concept has more than one word (e.g., "Movement Pattern," or "Movement Skill"); c) start with lower cased words, followed by capitalized subsequent words without any spaces, for slot names ; d) use the "has-" prefix convention, but others such as "requires-" (e.g., the "requiresMinimumLevel" property) or "isDevelopedBy" (e.g. the "isDevelopedByTrainingProgram" property).

Step 4. Define the classes and the class hierarchy

We used a top-down approach for defining the classes and developing the class hierarchy, a process that starts with defining the most general concepts in the domain, followed by the subsequent specialization of concepts. The "Movement Skill" class and derived classes included in the psychomotor domain ontology, together with a brief explanation of classes, are presented in Table 1.

Table 1. Motor Skill Acquisition Domain Ontology – Movement Skill Class Hierarchy.

Main Class	Subclasses in the hierarchy		Description
Movement Skill			Contains subclasses that categorize movements
	Fundamental Movement Skill	Stability Movement Skill	Contains subclasses that categorize movements, in which one's body orientation aims to gain and/or maintain a stable body orientation
		Locomotor Movement Skill	Contains subclasses that categorize movements, having the purpose of transporting the body from one point to another
		Manipulative Movement Skill	Contains subclasses that categorize movements, giving strength to an object or receiving strength from an object
	Specialized Movement	Daily Life Movement	Contains subclasses that categorize daily life movement patterns
		Leisure Movement	Contains subclasses that categorize movement patterns associated with leisure practices
		Professional Life Movement	Contains subclasses that categorize movement patterns associated with professional activities
		Sport Movement	Contains subclasses that categorize movement patterns associated with sports practices

There are also several other classes included in the ontology, with their corresponding hierarchies. For example, the Generic Profile class has the following derived sub-classes: "Anthropometric Profile" (class describing athlete body dimensions), "Morphologic Profile," "Physiologic Profile," "Flexibility Profile," "Functional Profile," "Perceptive Profile," and "Psychologic Profile". Other classes, such as "Movement Pattern," "Training Program," "Training Objective," "Performance Factor," "Training Period," "Macrocycle Period," "Microcycle Period," "Workout Period," "Warmup Period," etc. are part of the current representation.

Step 5. Define the properties of classes—slots

Our next step was to describe the internal structure of concepts. As we already listed the most important ontology terms in **Step 3**, the ones, which are not covered in the class hierarchy described in **Step 4,** become properties for the classes. Our ontology lists both intrinsic and extrinsic properties. Through inheritance, all subclasses of a class inherit the slot of that class. When designing the ontology, we ensured that the slots are attached to the most general class which supports that property. For example, the properties for class "Movement Pattern" are displayed in Table 2. The properties "hasGoal" and "hasRule" are used to encode a list of possible goals and rules respectively. The property "hasComplexity" is a String describing the complexity of the pattern, while the "hasEquipment" property encapsulates the representation of multiple "Equipment" instances. The remaining properties are of type Symbol and have a limited range of possible values (represented through Facets), which are explained further in **Step 6**.

Table 2. Movement Pattern Properties.

Name	Type	Cardinality	Other Facets
hasGoals	String	Single	-
hasRules	String	Single	-
hasComplexity	String	Single	-
hasEquipment	Instance	Multiple	Class={Equipment}s
hasMuscularAspect	Symbol	Single	Allowed values={Grow Motor, Fine Motor}
hasTemporalAspect	Symbol	Single	Allowed values={Discrete, Serial, Continuous}
hasEnvironmentalAspect	Symbol	Single	Allowed values={Open, Closed}

Another example is the "Psychomotor Profile" class. Its object properties are all the derived classes of "Generic Profile"; thus, we should instantiate each of them when creating an entity of this class. We have created specific slots for each sub-class of "Generic Profile", namely: "Anthropometric Profile" (7 properties), "Morphologic Profile" (8 properties), "Physiologic Profile" (9 properties), "Flexibility Profile" (49 properties), "Functional Profile" (4 properties), "Perceptive Profile" (2 properties), "Psychologic Profile" (6 properties). The visual representation of some of the classes' properties can be seen in Figure 2. Moreover, we also described the most important classes, with their relevant attributes, for the training planing part: "Training Program," "Training Objective," "Training Period," etc.

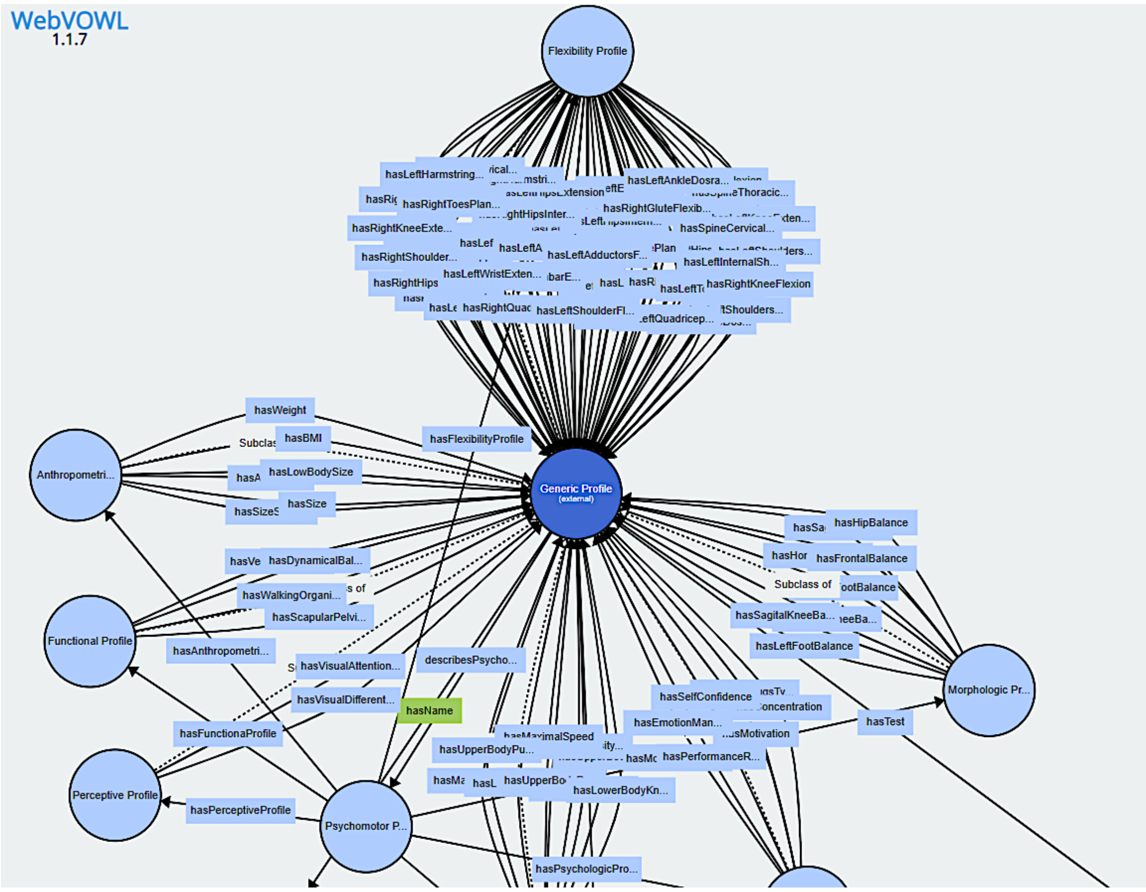


Figure 2. Partial WebVOWL Visualisation of Generic Profile sub-classes properties.

Step 6. Define the facets of the slots

The slot facet describes the value type, the number of values (also known as cardinality), together with other features the slot can take. As it can be seen in Table 2, the property “hasEquipment” is a value-type facet of type Instance, creating a relationship between an instance of “Movement Pattern” and multiple instances of class “Equipment” (cardinality is set to Multiple). Also, “hasMuscularAspect”, “hasTemporalAspect,” and “hasEnvironmentalAspect” properties are of type Symbol, allowing only values in a specified range (also known as enumerated slots). For example, the “hasTemporalAspect” property allows only the following values encoded as String: “Discrete,” “Serial,” and “Continuous”. The preliminary ontology representation following the previous six steps is presented in Figure 3. Step 7 as per Ontology 101 Framework, which implies the creation of instances for classes, will be detailed in a future ontology representation.

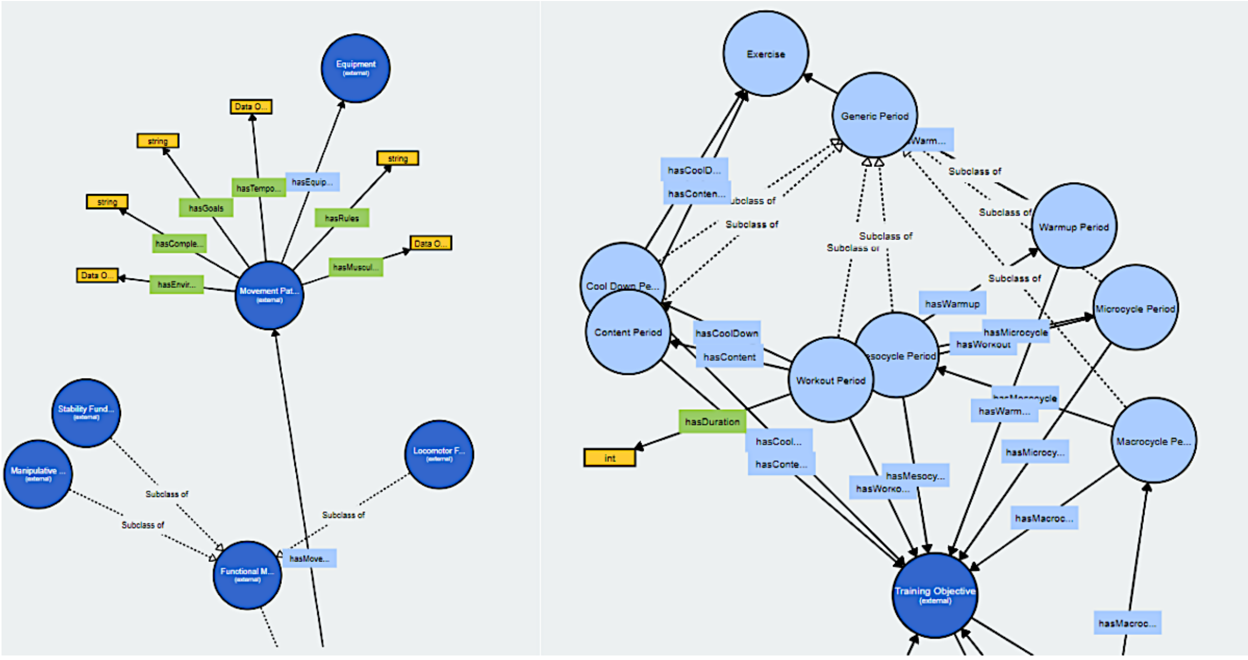


Figure 3. Partial WebVOWL Visualisations corresponding to the Motor Skill Acquisition Ontology.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

The current paper presents our ongoing work for designing a comprehensive ontology to represent the psychomotor domain. The goal of this process is to understand the mechanics of the motor skills acquisition process and integrate the ontology in a learning environment capable of assessing and enhancing a user's progress in the sport for general health activities. As Intelligent Tutoring Systems (ITSs) are the most suitable environments for such activities, we decided to use the Generalized Intelligent Framework for Tutoring (GIFT), an extension for ITS that overcomes many of the legacy ITS implementations and also has real-world applications in the Psychomotor Domain.

Overall, the proposed ontology follows the features specific to knowledge systems: it is consistent in structuring and organizing information while ensuring extensibility and reusability for follow-up experiments.

The proposed ontology presents the concepts and relations of the psychomotor domain and may be reused by future developers in their work – once finalized, it will be published online under an open-source license, and it will be integrated into the Linked Open Vocabularies (LOV) initiative. Future research includes the representation of part of the entities, specific to this domain (e.g., defining few exercises, movement skills, and training profiles). Based on these representations, future work may include applying SPARQL queries to determine specific information (e.g., retrieve all training programs which have more than four microcycles or get all trainees who have passed the Maximal Aerobic Speed test). Our future research interests focus on implementing the complete psychomotor domain ontology, integrating it in GIFT, and developing an ITS based on GIFT for training sport-for-health skills.

REFERENCES

Benesh, R., & Benesh, J. (1983). *Reading dance: The birth of Choreography* (First Ed.). New York, NY, USA: McGraw Hill.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain*. New York, NY, USA: David McKay Co Inc.

Brawner, K., Hoffman, M., & Nye, B. (2019). Architecture and Ontology in the Generalized Intelligent Framework for Tutoring: 2018 Update. In *7th Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium* (pp. 11): US Army Combat Capabilities Development Command–Soldier Center.

Button, C. (2021). *Dynamics of skill acquisition: an ecological dynamics approach* (2nd ed.). Champaign, IL, USA: Human Kinetics, Inc.

Dave, R. H. (Ed.). (1970). *Psychomotor levels in Developing and Writing Behavioral Objectives*. Tucson, Arizona, USA: Educational Innovators Press.

Diaz-Rodriguez, N., Wikstrom, R., Lilius, J., Cuellar, M. P., & Flores, M. D. C. (2013). Understanding Movement and Interaction: An Ontology for Kinect-Based 3D Depth Sensors. In U. G., O. S. F., B. J., C. L. L. & O. J. (Eds.), *Ubiquitous Computing and Ambient Intelligence. Context Awareness and Context-Driven Interaction (UCAmI 2013)* (pp. 254–261). Carrillo, Costa Rica: Springer.

Euzenat, J., & Shvaiko, P. (2007). *Ontology Matching*. New York, NY, USA: Springer Heidelberg.

Fernández-López, M., Gómez-Pérez, A., & Juristo, N. (1997). Methontology: from ontological art towards ontological engineering.

Goodway, J. D., Ozmun, J. C., & Gallahue, D. L. (2019). *Understanding Motor Development: Infants, Children, Adolescents, Adults*. Burlington, MA, USA: Jones & Bartlett Learning.

Harrow, A. J. (1972). *A taxonomy of the psychomotor domain*. New York, NY, USA: David McKay Co.

Khan, Z., & Keet, C. M. (2012, October 8–12). ONSET: automated foundational ontology selection and explanation. In *18th Int. Conf. on Knowledge Engineering and Knowledge Management* (pp. 237-251). Galway City, Ireland: Springer, Berlin, Heidelberg.

Ma, M., & Kevitt, P. M. (2004). Visual Semantics and Ontology of Eventive Verbs. In *International Conference on Natural Language Processing (IJCNLP 2004)* (pp. 187–196). Hainan Island, China: Springer.

McGuinness, D. L., & Van Harmelen, F. (2004). OWL Web Ontology Language overview. *W3C recommendation, 10*(10), 2004.

Neagu, L.-M., Rigaud, E., Travadel, S., Dascalu, M., & Rughinis, R.-V. (in press). Intelligent Tutoring Systems for Psychomotor Training – A Systematic Literature Review. In *16h Int. Conf. on Intelligent Tutoring Systems (ITS 2020)*. Athens, Greece: Springer.

Newell, K. M. (1985). Coordination, control and skill. In D. Goodman, R. B. Wilberg & I. M. Franks (Eds.), *Differing perspectives in motor learning, memory, and control* (pp. 295–317). Amsterdam, Holland: Elsevier Science Publishing Company.

Newell, K. M. (1991). Motor skill acquisition. *Annual review of psychology, 42*, 213–237.

Nkambou, R., Bourdeau, J., & Mizoguchi, R. (2010). Introduction: What Are Intelligent Tutoring Systems, and Why This Book? In R. Nkambou, J. Bourdeau & R. Mizoguchi (Eds.), *Advances in Intelligent Tutoring Systems* (pp. 1–12). Berlin, Germany: Springer.

Noy, N. F., Crubezy, M., Fergerson, R. W., Knublauch, H., Tu, S. W., Vendetti, J., & Musen, M. A. (2003). Protege-2000: an open-source ontology-development and knowledge-acquisition environment. In *AMIA Annual Symposium* (pp. 953). Washington, DC, USA: American Medical Informatics Association.

Noy, N. F., & McGuinness, D. L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Stanford, CA, USA: Stanford Knowledge Systems Laboratory.

Paviotti, G., Rossi, P. G., & Zarka, D. (2012). *Intelligent Tutoring Systems: an overview*. Lecce, IT: Pensa Multimedia.

Saad, S., Mahmoudi, S., & Manneback, P. (2012). Semantic analysis of human movements in videos. In *Proceedings of the 8th International Conference on Semantic Systems - I-SEMANTICS '12*. Graz, Austria: ACM.

Simpson, E. J. (1972). The Classification of Educational Objectives in the Psychomotor Domain. Washington, DC: Gryphon House.

Sottilare, R. A. (2012). Considerations in the development of an ontology for a generalized intelligent framework for tutoring. In *International Defense & Homeland Security Simulation Workshop in Proceedings of the I3M Conference* (pp. 19–25). Vienna, Austria.

Sottilare, R. A., Brawner, K. W., & Sinatra, A. M. (2017). An updated concept for a Generalized Intelligent Framework for Tutoring (GIFT). *GIFTtutoring.org*.

Sottilare, R. A., Graesser, A., Hu, X., Olney, A., Nye, B., & Sinatra, A. M. (2016). Introduction to Domain Modeling and GIFT *Design Recommendations for Intelligent Tutoring Systems* (Vol. 4 - Domain Modeling, pp. 1–15). Orlando, Florida: US Army Research Laboratory.

W3C. (2013). SPARQL Query Language for RDF. Retrieved August 15, 2019, from https://www.w3.org/TR/rdf-sparql-query/

ABOUT THE AUTHORS

Laurențiu-Marian Neagu *is a PhD Candidate at MINES ParisTech / PSL Research University. His main areas of interest are Intelligent Tutoring Systems and Smart Learning Environments applied to Sports Training*

Vincent Guarnieri *is a Master's Degree Candidate in Sports Science at Université Côte d'Azur, France. His main area of interest is the optimization of human performance.*

Eric Rigaud *is a researcher at MINES ParisTech Centre for research on Risks and Crisis. His research objective is the resilience of individuals, organizations, and territories facing complex situations.*

Sébastien Travadel, *Chief Engineer at the French Ministry in charge of infrastructure, is currently researcher at MINES ParisTech / PSL Research University. He studies Artificial Intelligence applied to resilience engineering and risk management.*

Mihai Dascălu *is a Professor at University Politehnica of Bucharest and his main research areas are Intelligent Tutoring Systems, Natural Language Processing and the semantic web.*

Răzvan-Victor Rughiniș *is a Professor at University Politehnica of Bucharest interested in Artificial Intelligence and semantic web technologies.*

Integrating an Engagement Classification Pipeline into a GIFT Cybersecurity Module

Benjamin D. Nye[1], Mark G. Core[1], Daniel Auerbach[1], Aviroop Ghosal[1], Shikhar Jaiswal[2], Milton Rosenberg[1]
University of Southern California, Institute for Creative Technologies[1], Indian Institute of Technology Patna[2]

INTRODUCTION

Measuring engagement for learners is critical in computer-based training. Too often, when checkbox training is completed, learners have gone through the required motions but did not meaningfully engage with the material. However, it is challenging to develop generalized measures for engagement: different systems and learning material will result in different patterns of reaction times, responses, and scores. We addressed this challenge through machine learning in a prototype system called the Service for Measuring and Adapting to Real-Time Engagement (SMART-E).

SMART-E classifies engagement using a play-test methodology based on player personas; a small set of users act out engagement archetypes (e.g., diligent, distracted, racing). This archetype data is used as seed data for leveraging unlabeled user data for semi-supervised learning. In this work, we plugged SMART-E into a brief course on cyber-security topics built using the Generalized Intelligent Tutoring Framework (GIFT; Sottilare, Brawner, Goldberg, & Holden, 2012; Sottilare, Baker, Graesser, & Lester, 2018). We intentionally designed the two modules of this course such that one of them (HTTPS) should be less engaging than the other (Phishing) and adjusting for self-reported initial interest and expertise we would expect engagement scores to be higher for the Phishing module. In this paper, we describe the process of integrating the SMART-E pipeline into GIFT and our experiment to test SMART-E in the context of the GIFT cyber-security course.

BACKGROUND

Research has shown that engagement is essential for training and education, with impacts on both learning and persistence vs. attrition outcomes (Baker et al., 2010; Christenson et al., 2012; Lehman et al., 2011). Thus, it is critical to optimize engagement in order to promote training effectiveness. However, the learning sciences lacks standard measures of engagement. By comparison, fields where engagement has always been a top priority tend to present certain common metrics for comparing systems much like the learning sciences report learning gains (e.g., consumer engagement in marketing; web page analytics in human-computer interaction; Stavrakantonakis, Gagiu, Kasper, Toma, & Thalhammer, 2012; Haven, 2007).

One reason that standards for engagement in learning have been slow to develop is that educational settings vary widely in terms of available relevant real-time measures and longer term information (e.g., in a web-based system you can track repeat visits but in a museum you often cannot). Even for constrained slices of engagement (e.g., at the affective level), there are a wide range of possibilities (Baker et al., 2010): some systems interact with learners via dialogue allowing the text and potentially speech to be analyzed for cues of engagement; learner physical behavior and facial expressions can be analyzed by humans or machines, and some researchers use equipment such as eye trackers and pressure sensitive chairs. Given this variety, it is understandable that researchers have focused on supporting their particular configuration rather than building general purpose engagement measuring tools.

Moreover, at an analytics level, there are substantial barriers to developing general-purpose metrics and classification of engagement. In this work, we consider two main barriers: 1) Automating Pipelines and 2) Content-Relativity. These raise significant questions that must be answered before being able to develop a framework, including:

1) Automating Pipelines:

- Standard Format: How to record data from many systems?

- Modularity: How to re-use analytics with minimal change?

2) Content Relativity:

- Event Set: What actions occur across a variety of systems?
- Normative vs. Atypical Behavior: For a single measure such as response time, what is normative vs. atypical?
- Behavioral Patterns: What does "engaged" look like? How many common patterns exist?

The concern when developing a framework is ultimately: can it generalize over a broad set of common cases? Some of these questions have reasonable answers based on prior research. However, others remain open.

Automating Pipelines for Engagement Metrics

A variety of solutions already exist for implementing analytics workflows, including ones that allow calls to external third-party services, and ones that integrate with commonly-used statistical libraries. There are also a number of active research projects developing workflow analytics engines specifically for learning data, which include the NSF LearnSphere project (Koedinger, Liu, Stamper, Thille, & Pavlik, 2017) and the Collected Learning Analytics system (CLA; Bakharia, Kitto, Pardo, Gašević, & Dawson, 2016) which is designed to analyze DoD-standard xAPI Learning Record Stores (LRS; Advanced Distributed Learning, 2020). xAPI offers a *standard format* for recording data from many systems. While its logging remains too broad to directly use for generalized analytics, it offers a foundation for a set of common events that can be logged across a variety of systems. Research is also ongoing on platform-agnostic solutions for data mining models (e.g., Vartak et al., 2016). Moreover, commercial machine learning frameworks also enable analytics pipelines (e.g., Amazon SageMaker).

The above frameworks indicate a pathway toward *modularity*. Many systems currently collect event logs of user behavior. Based on these relatively raw logs, sets of metrics can be calculated (i.e., raw metrics). Next, intermediate metrics may be calculated and in some cases updated based on the raw metrics. This offers a pathway to modularity, since a standardized logging format enables evaluating a variety of raw and intermediate metrics.

Content Relativity

Differences in content between systems can mean that engagement manifests differently in behavior patterns. The first step toward addressing this challenge is to identify a set of engagement-relevant events that can be measured across a variety of systems. In general, we consider the following events and related metrics, and in this work, focus on the bold categories.

- Facial Expressions: A variety of measures based on analysis of facial expressions in video data. Raw metrics may involve high-level emotions (e.g., happiness, sadness, confusion, frustration) as well as patterns of facial action units (Ekman & Friesen, 1977). These metrics can be normalized and aggregated using different methodologies to study specific contexts or decision-points.
- **Time-on-Task**: Overall time-on-task as well as breakdown of time in different parts of the system and for different sessions.
- **Interaction Levels**: Metrics about quantity of learner contributions and inputs (e.g., mouse clicks, verbosity of speech input).
- **Decision Events/Correctness:** Question-answering and problem-solving metrics. May involve correctness of learner actions as well as combinations of events (e.g., facial expressions before/during/after a decision).

- Help/Support Levels: Metrics about usage of hints, feedback and other resources designed to improve performance or learning. Especially notable are metrics on gaming the system (e.g., requesting as many hints as possible).

- Self-Reported Constructs: Processed self-report and survey data (e.g., simple aggregation, more complex measures developed via factor analysis).

- Learning Gains: Learning gains between pre-test and post-test, based on question batteries that can be broken down by arbitrary categories (e.g., knowledge components or other taxonomies).

In this work, we focused on time, interaction levels, and correctness. Some metrics were also calculated across individuals (e.g., task difficulty as a factor which should interact with individual correctness and time). To search for *atypical patterns*, we used metrics which compared the user to the expected population behavior, such as z-scores (e.g., deviations from the population norm), and correlation or lack of correlation (e.g., if a users' time for tasks and their correctness was not well-correlated). These features are fairly general to calculate and may help distinguish between more or less engaged users.

However, a general engagement measurement framework must account for users having different behavior patterns despite having the same level of engagement. For example, one user might exhibit disengagement by racing through the material and receiving poor scores while another disengaged user may not pay attention and have long periods of inactivity between and during problems. Our approach to this challenge is to develop a set of "player persona" archetypes such that each archetype exhibits distinct behavior patterns and once a user is matched to their persona then automated metrics can be used to measure their engagement level.

Engagement Archetypes: Player Personas

Based on a review of behavioral engagement and disengagement, we posit that engagement has at least two dimensions: a) passiveness vs. activeness and b) avoidance vs. approach. The avoidance versus approach dimension captures the underlying engagement of the learner (i.e., not engaged with the learning experience versus engaged). Engagement and disengagement can be expressed as passive or active patterns of behavior. For example, the passive-avoidance combination results in low effort behaviors such as not paying attention or quickly clicking buttons to advance the system. By comparison, in the active-avoidance combination learners use short-cut strategies to cheat or cherry-pick tasks to minimize effort while still providing acceptable performance. Although approach learners are engaged (e.g., genuine effort toward problem solving), they may take a passive, low-effort approach compared to active learners who spend extra time on problem-solving (e.g., reflect on their choices) or self-regulate their learning (e.g., take advantage of freedom to choose material).

These latent engagement factors may result in different observed patterns. For example, while distraction and racing through material both represent passive disengagement, their data patterns will look very different. In considering these patterns, we developed the following archetypes which may be evident across a variety of systems:

- Diligent (Active Engagement): Spends somewhat more time on tasks and shows correspondingly better performance, and is more likely to complete optional tasks.

- Nominal Engagement (Passive Engagement): Completes tasks as recommended or assigned, with ordinary time-on-task and performance.

- Expert/Recall (Passive Engagement): Regardless of difficulty level, completes tasks very rapidly and with high performance. Possibly an expert on the content, but might also be shallow recall or lookup.

- Racing/Guessing (Passive Disengagement): Rapidly answers (potentially multiple times) despite relatively poor performance (Leiner, 2013).

- Distracted/Slow (Passive Disengagement): Uncommonly delayed or irregular answers, particularly when extra time does not appear to improve performance (Mattheiss et al., 2010).

- Self-Regulated (Active Engagement): Seeks out and spends greater time on harder tasks, but may skip or disengage on easier tasks (Janning, Schatten & Schmidt-Thieme, 2016; Weissgerber, Reinhard & Schindler, 2016).

- Cherry Picking (Active Disengagement): Seeks out easier tasks or abuses features to make tasks easier (e.g., hint abuse), and avoids harder tasks (Baker et al., 2006).

This set of engagement archetypes is not meant to be exhaustive but rather a set of common patterns which most instructors and system designers would understand if detected, and which, in many cases, can be emulated by testers. Not all patterns are relevant to all learning systems: in the work described here with GIFT, cherry-picking and significant self-regulated learning are not possible since learners cannot select or skip resources. In the next section, we consider how archetypes are used to classify learner performance.

APPROACH: SEMI-SUPERVISED LEARNING

To develop a generalized classifier for the behavioral archetypes, we developed an approach which is influenced by two techniques: 1) *semi-supervised learning*, which trains with a small set of labeled data and a larger set of unlabeled data and 2) video game play-testing, where testers sometimes act out multiple user personas/archetypes to test different situations (Chapelle, Scholkopf & Zien, 2009; Winn, 2009).

SMART-E Pipeline: Measuring and Classifying Engagement

While this paper focuses on a specific data set, the techniques applied here are designed to be generalizable and re-usable as part of the SMART-E pipeline. Data is required to be in a standard format, learning records that meet the xAPI standard (Advanced Distributed Learning, 2020). This raw xAPI data may either be sent directly by the system (e.g., through an API for logging) or by running a converter on system-generated logs after-the-fact. The first step in the pipeline is cleaning the raw xAPI data logs (partially system-specific) which corrects common data problems, such as sessions that terminated improperly or missing data fields that can be inferred from other data. The result is a canonical xAPI data store that does not have missing data.

SMART-E requires activities in logs to contain metadata allowing them to be structured into an activity tree, which represents the hierarchical structure of both sequential and parallel activities. While activities can be nested arbitrarily, four levels are analyzed to generate raw metrics tables: steps, tasks, lessons, and sessions. Raw metrics primarily concern time-based information (e.g., duration of a task, response time for first step, Laplace-smoothed logarithm of each task duration (i.e., $\ln(t+1)$), score-based information (e.g., numerical score and/or correctness), and support used (e.g., hint counts, retries of a problem). Metrics related to skills are not calculated, since the majority of systems do not tag their tasks with a consistent ontology of knowledge components. Intermediate metrics are generated using feature construction calculations based solely on raw metrics (i.e., without accessing the original xAPI logs). For this work, the most important intermediate metrics are averages across attempts (e.g., average scores, average task duration), the average difficulty for each task (inferred from first-attempt scores) and z-scores for task metrics (e.g., time-on-task for the learner relative to other users).

Based on these metrics, feature vectors are generated that represent each learner's performance in the system. In the current work, these vectors are calculated from the learner's task data for a single session, though one could generate similar features for specific tasks, across multiple sessions, or for recent tasks in a session (i.e., any collection of tasks). First, two simple features were calculated: average response time across tasks (Avg. RT) and average task performance (Avg. Score). These were considered the minimal information to potentially infer engagement. Next, a more complex feature set was developed to model interactions between task response time, task scores, and task difficulty. Based on z-score cutoffs, each variable was split into three bins (low, med, high) when possible, and into the most bins available when not (e.g., only medium if all values equal; only low and high if only two types of scores). Each scored task incremented an associated bin (e.g., fast answer with a high score on a hard problem).

To support this approach, a set of play-testers must perform runs through the system that follow different player persona behaviors. In this study, the archetypes that testers were instructed to attempt were: Diligent, Expert, Racing, and Distracted. In our semi-supervised approach, we align five clusters derived from unsupervised learning to the four play-tester archetypes. The cluster least matching the labeled data is assigned a fifth archetype (Nominal Engagement/Average).

GIFT Integration

To collect this data, the Generalized Intelligent Framework for Tutoring (GIFT) was instrumented to collect xAPI records relevant to SMART-E. This integration built on top of the GIFT Mulit-Agent Architecture (GIFT-MAA) framework (Nye, Auerbach, Mehta, & Hartholt, 2017; Nye, Thaker, Surana, Auerbach, & Brawner, 2018). Using the AgentContainer module, a new service was added which monitored all messaging traffic in a local GIFT instance. When messages that matched an appropriate pattern were detected, xAPI messages were constructed and sent to a Learning Record Store. The specific actions that were logged included Session Start/End, GIFT Course Lesson Start/End, Task Start/End (e.g., for any GIFT activity that could be authored), and scores for any multiple choice tests and dialog-based assessments.

All statements were timestamped, so the time to complete each activity was recorded. One challenge for this logging approach in GIFT was that messages were not generated for each attempt or selection for individual survey/test items, so reaction time for individual responses was not possible to collect. As such, question-level timing data was not available to analyze.

STUDY DESIGN: GIFT CYBERSECURITY MINI-COURSE

To evaluate this approach, a GIFT course was created which contained two modules: Phishing and HTTPS. Each module consisted of six activities: Text Introduction, Video Overview, Basic Questions (3 multiple choice), Intermediate Questions (3 harder multiple choice), an External URL, and A Dialog-Based assessment. The structure of the course is shown in Figure 1. Before and after the course, users were surveyed for their interest and experience level in the content.

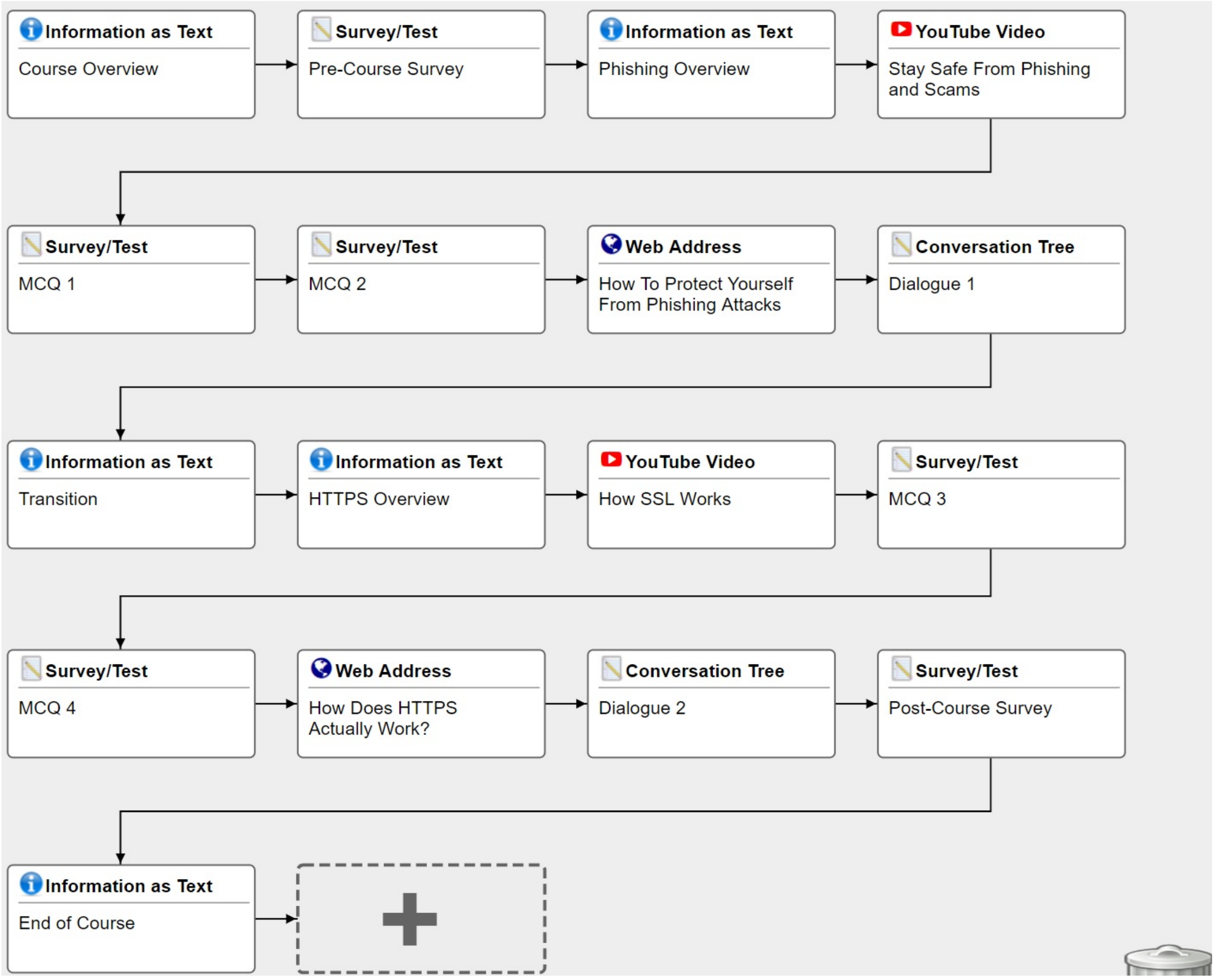
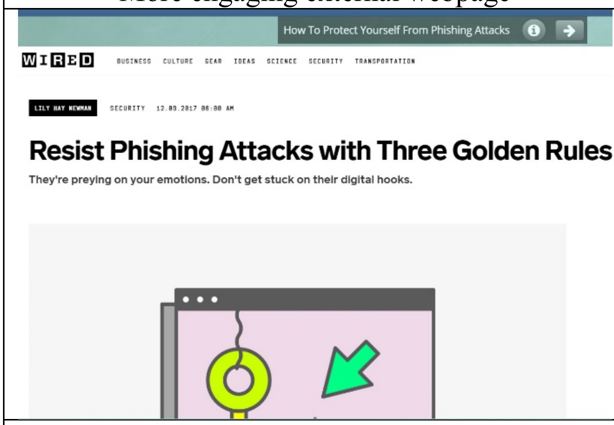
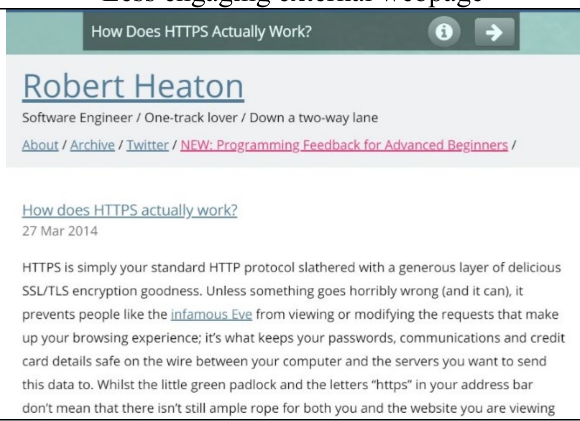


Figure 1: GIFT Cybersecutity Mini-Course

While both modules had an equivalent number and type of activities, the content was intentionally selected such that Phishing was likely to be more engaging than HTTPS (e.g., a long, dry video versus a shorter animated video; content that focused on technical details rather than personal outcomes). This design allows the comparison of metrics from individuals who are potentially engaged in the Phishing material and disengaged from the HTTPS material. Figure 2 shows a comparison between equivalent content from the two modules.

<table>
<tr><th>More engaging external webpage</th><th>Less engaging external webpage</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>More engaging, personally relevant questions</td><td>Less engaging, less personally relevant questions</td></tr>
</table>

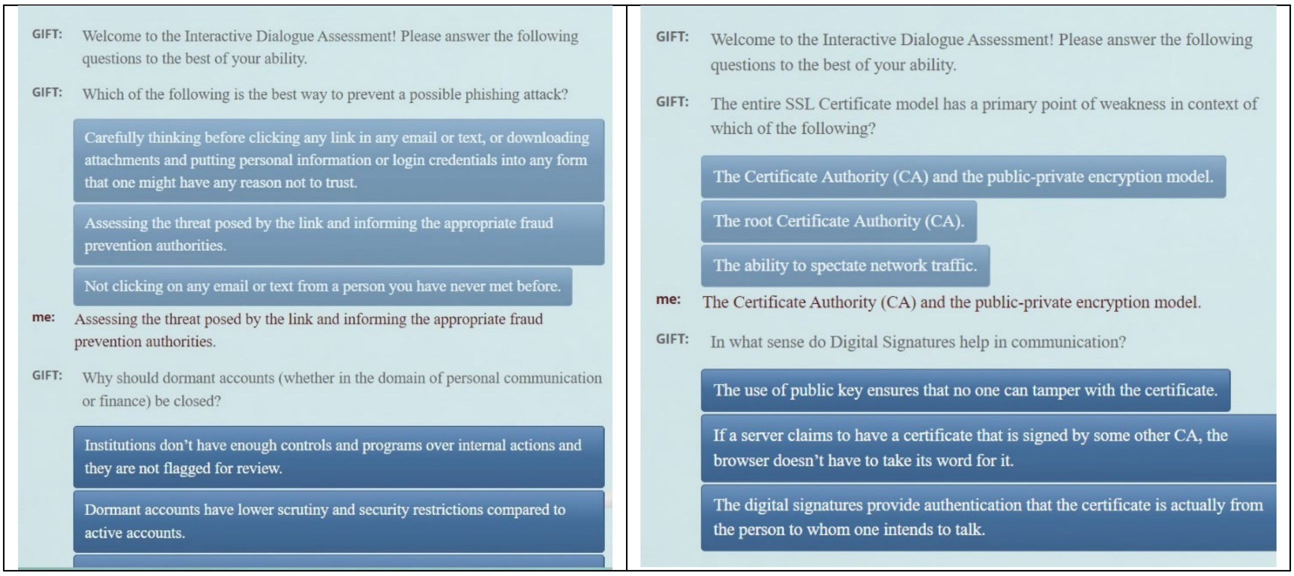


Figure 2. Example of Phishing (Left) vs. HTTPS (Right) Activities

Test users completed the course multiple times for a total of 17 archetype data points spread approximately evenly across each category. In most cases, testers could easily simulate the archetype (e.g., checking email when Distracted, and using an answer sheet when Expert). The exception was Diligent which had to be performed first before the user gained an unfair advantage having seen the material before. Data was then collected from 100 paid volunteers. A cleaning script was developed to handle logs from aborted sessions (e.g., accidentally started the wrong course) or other issues that led to duplicate sessions. The cleaned logs were transferred to a canonical LRS.

FUTURE DIRECTIONS

While data is being analyzed from this study, we can say that the SMART-E system integrated effectively with GIFT and this process was made substantially easier by leveraging the GIFT Multi-Agent Architecture AgentContainer to add the service for logging xAPI data. We also found that the process of developing a GIFT course using the new authoring tools is substantially faster and easier than using previous versions of GIFT. A novice student author was able to fairly quickly develop the mini-course, with only minor difficulty in associating concepts with certain assessment types. The data collection also ran smoothly, with a high level of stability such that valid data appears to be available from all participants demonstrating the success of the data collection components of our pipeline.

While initial statistics have been calculated, and data clustering performed, further analyses are required to interpret these results. Preliminary analyses on correctness vs. time-on-task patterns conform to expectations: *Distracted* (slow average response time, low average task performance), *Racing* (fast average response time, low average performance), *Diligent* (medium average response time, high average performance), *Expert* (fast average response time, high average performance) and *Average* having medium average response time and medium average performance. These results suggest that the semi-supervised classification approach is appropriate for this type of course. To investigate this further, we will be running the system across multiple random orderings of the subject data set to determine the consistency of SMART-E classification with relatively small data (e.g., 25-50 subjects) versus the full data set. The classification results will also be compared against users' self-reported engagement levels. The goal of this work will be to determine the cold-start performance of our engagement classification. As this approach has already been tested on a separate data set for scenario-based tutoring, positive results on this GIFT data set would verify the re-usability of this approach across multiple systems.

ACKNOWLEDGEMENTS

This research work was conducted under the USC ICT University Affiliated Research Center (UARC) and sponsored by the under U.S. Army Research Laboratory (W911NF-14D-0005). However, all statements contained of this publication are the work of the authors alone and do not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

REFERENCES

Advanced Distributed Learning (2020). *xAPI technical specification*. (https://github.com/adlnet/xAPI-Spec/blob/1.0.3/xAPI.md).

Baker, R. S., Corbett, A. T., Koedinger, K. R., Evenson, S., Roll, I., Wagner, A. Z., Naim, M., Raspat, J., Baker, D. J., & Beck, J. E. (2006). Adapting to when students game an intelligent tutoring system. *International Conference on Intelligent Tutoring Systems (ITS),* Springer.

Baker, R. S., D'Mello, S. K., Rodrigo, M. M. T., & Graesser, A. C. (2010). Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies,* 68(4), 223–241.

Bakharia, A., Kitto, K., Pardo, A., Gašević, D., & Dawson, S. (2016). Recipe for success: lessons learnt from using xAPI within the connected learning analytics toolkit. *International Conference on Learning Analytics & Knowledge,* ACM.

Chapelle, O., Scholkopf, B., & Zien, A. (2009). Semi-supervised learning. *IEEE Transactions on Neural Networks,* 20(3)*,* 542.

Christenson, S. L., Reschly, A. L., & Wylie, C. (2012). *Handbook of Research on Student Engagement*. Springer, New York.

Ekman, P., Friesen, W.V. (1977). *Facial Action Coding System*. Consulting Psychologists Press, Stanford University, Palo Alto.

Haven, B. (2007). *Marketing's new key metric: Engagement*. Forrester Marketing.

Janning, R., Schatten, C., & Schmidt-Thieme, L. (2016). Perceived task-difficulty recognition from log-file information for the use in adaptive intelligent tutoring systems. *International Journal of Artificial Intelligence in Education,* 26(3), 855–876.

Koedinger, K., Liu, R., Stamper, J., Thille, C., & Pavlik, P. (2017). Community based educational data repositories and analysis tools. *International Learning Analytics & Knowledge Conference,* ACM.

Lehman, B., D'Mello, S. K., Strain, A. C., Gross, M., Dobbins, A., Wallace, P., Millis, K., & Graesser, A. C. (2011). Inducing and tracking confusion with contradictions during critical thinking and scientific reasoning. *International Conference on Artificial Intelligence in Education (AIED).*

Leiner, D. J. (2013). Too fast, too straight, too weird: Post hoc identification of meaningless data in internet surveys. *SSRN Electronic Journal.*

Mattheiss, E., Kickmeier-Rust, M., Steiner, C., & Albert, D. (2010). Approaches to detect discouraged learners: Assessment of motivation in educational computer games. *eLearning Baltics (eLBa)*, 10(1), 10.

Nye, B. D., Auerbach, D., Mehta, T. R., & Hartholt, A. (2017). Building a backbone for multi-agent tutoring in GIFT (Work in progress). *Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym5).*

Nye, B.D., Thaker, R., Surana, N., Auerbach, D. and Brawner, K. (2018). Toward Self-Improving Multi-Agent Tutoring Architectures: Progress and Future Directions. *Design Recommendations for Intelligent Tutoring Systems*, *Vol. 8.*

Sottilare, R., Brawner, K., Goldberg, B. & Holden, H. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). US Army Research Laboratory.

Sottilare, R. A., Baker, R. S., Graesser, A. C., & Lester, J. C. (2018). Special Issue on the Generalized Intelligent Framework for Tutoring (GIFT): Creating a stable and flexible platform for Innovations in AIED research. *International Journal of Artificial Intelligence in Education*, *28*(2), 139-151.

Stavrakantonakis, I., Gagiu, A. E., Kasper, H., Toma, I., & Thalhammer, A. (2012). An approach for evaluation of social media monitoring tools. *Common Value Management, 52*(1), 52-64.

Vartak, M., Subramanyam, H., Lee, W. E., Viswanathan, S., Husnoo, S., Madden, S., & Zaharia, M. (2016). Model DB: A system for machine learning model management. *Workshop on Human-In-the-Loop Data Analytics,* ACM.

Weissgerber, S. C., Reinhard, M.-A., & Schindler, S. (2016). Study harder? The relationship of achievement goals to attitudes and self-reported use of desirable difficulties in self-regulated learning. *Journal of Psychological and Educational Research,* 24(1), 42.

Winn, B. M. (2009). The design, play, and experience framework. *Handbook of Research on Effective Electronic Gaming in Education*, IGI Global.

ABOUT THE AUTHORS

Benjamin D. Nye, Ph.D. *is the Director of Learning Sciences at USC ICT. He received his Ph.D. in Systems Engineering from the University of Pennsylvania in 2011 and previously served as a Research Professor at the University of Memphis. Ben's major research interest is to identify best-practices in advanced learning technology, particularly for frontiers such as distributed learning technologies (e.g., cloud-based, device-agnostic) and socially-situated learning (e.g., face-to-face mobile use). Ben's research tries to remove barriers development and adoption of intelligent tutoring systems so that they can reach larger numbers of learners, which has traditionally been a major roadblock for these highly-effective interventions.*

Mark G. Core, Ph.D. *is a Research Scientist at USC ICT. He received his Ph.D. from the University of Rochester in 2000, and was a Research Fellow at the University of Edinburgh until joining USC ICT in 2004. Mark specializes in artificial intelligence in education using virtual humans as role players for interpersonal skills training (e.g., negotiation, cultural awareness and leadership). His research focuses on how best to support learners using automated guidance, and support authors creating realistic and engaging virtual humans that promote learning.*

Daniel Auerbach *is a Research Programmer at USC ICT, who has developed and expanded a variety of intelligent tutoring systems and frameworks. He received a B.A. in Linguistics from Cornell University.*

Aviroop Ghosal *contributed to this work as a graduate student research worker and intern at USC ICT. He is currently a Software Development Engineer at Amazon Marketplace Web Services, working on enhancing the API experience for API authors and external developers. Aviroop has always been passionate about numbers and programming, and is often on the lookout for technological advancements in AI and Machine Learning.*

Shikhar Jaiswal *is a final year undergraduate student in the Department of Computer Science at IIT Patna. He is primarily interested in theoretical and applied research, in the broader areas of Artificial Intelligence and Systems. Shikhar collaborated with the Learning Sciences group on this project as a summer intern at USC ICT.*

Milton Rosenberg *is a Special Project Manager at USC ICT. He holds a BS in Business Administration with over 20 years of experience managing multiple projects, with a focus on learning technologies and simulation-based training. Earlier in his career, Milton also supervised the construction and global rollout of the Whirlpool manufacturing system and holds a patent in methods for maintaining computer software.*

THEME III:
AUTHORING AND DEVELOPMENT

Using GIFT to Develop Adaptive Remedial Courses for Graduate Degree Programs in Data Science

Hong Liu, Timothy Bernard, and Keshav Acharya
Embry-Riddle Aeronautical University

INTRODUCTION

This paper presents the instructional design and work-in-progress (WIP) using the Generalized Intelligent Framework for Tutoring (GIFT, Sottilare et al., 2012) to develop adaptive remedial mathematics courses for a graduate program, Master of Science in Data Science (MSDS). While existing computer-based applications for adaptive online learning are limited to serve K12 to low-level college mathematics subjects (Aleven et al., and Aleks https://www.aleks.com/), our adaptive learning courses are based on tested course materials and extend the educational technology to serve advanced mathematics topics - Matrix Algebra and Matrix Calculus. This paper also showcases how the interoperable, plug-and-play educational technologies centered on the GIFT system and xAPI data exchanges can ensure that our applications to have unlimited growth potential. The rest of the paper is organized as follows. First, we discuss the background of the MSDS degree program as well as the motivation of using GIFT to offer the courses. Next, we outline the pedagogical approach and instructional design. Then, we describe the design and tool configuration for GIFT system and other cloud-based services. Finally, we conclude the paper by considering future work.

MOTIVATION TO DEVELOP THE ADAPTIVE COURSES

As more data and analytic methods become available, work across nearly all domains is becoming more data driven. This recent increase in demand has attracted students, including veterans from a broad variety of undergraduate majors to graduate programs in data science. Embry-Riddle Aeronautical University (ERAU) offers a flexible MSDS program for students to specialize in one of the five tracks - High-Performance Computing, Cybersecurity, Aviation Safety, Aviation Business, and Homeland Security. These five tracks share five core courses in data analytics and differentiate from their elective and capstone courses. Each track offers a bridging course that helps students transfer a domain-specific problem into a data analytic problem. For example, the students in Aviation Safety Track will take a bridging course that studies how to use flight accident data to gain insight for improving fly instruction or aircraft maintenance. Each track has a capstone course or internship co-mentored by faculty and practitioners for students to solve a real-world data-intensive application problem.

The prerequisites of MSDS at ERAU include Matrix Algebra (MA) and Matrix Calculus (MC), which is necessary for an intuitive understanding of high dimensional data. Students with bachelor's degrees in natural science or engineering programs are expected to have this skill because their math courses cover most topics of MA and MC. However, students in the Aviation Science and Business tracks may have deficiencies in MA and MC because of their lower-level math requirements. Therefore, we need to prepare the remedial MA and MC courses so that we can offer conditional admission to these students by requiring them to take and pass these courses online before they enter the fall semester.

We believe that the deficiency of MA and MC is a common problem for the data science programs at many Universities. However, our ad hoc solution does not scale up for large enrollment. Massive Online Open Courses (MOOCs) are not only promising to make otherwise unaffordable education accessible to motivated learners but also turn teaching from a solo task for an instructor to a crowdsourcing developmental effort for researchers and educators. However, the lack of timely feedback and peer-support have been identified as two primary factors for high drop rates of MOOCs. To ameliorate this problem, adaptive Learning and Cognitive Tutoring (Aleven, V. et al., 2019) with online formative assessments (R. Sottilare et al., 2017) can provide immediate feedback based on learner models and domain knowledge profiles. Since the domain knowledge models of MA and MC are more challenging than those of Aleks (https://www.aleks.com/), our pragmatic approach aims to keep instructors in the loop. GIFT tutoring is used to replace manual work gradually and improve the automation of the adaptive learning MOOC iteratively over the next few years.

THE PEDAGOGY AND INSTRUCTIONAL DESIGN

Student-centered learning and data-driven learning assessment have dominated research literature as well as government-funded educational projects in the last two decades. We reviewed some articles that are relevant to this context. Hannafin, M. and Hannafin, K. M. 2010 addressed the issues of offering student-centered learning through web-based education. Chi 2009 defined a conceptual framework called iCAP to differentiate interactive, active, constructive, and passive learning pedagogies. While the lecture-homework-test approach still dominates college teaching, the two of the faculty co-authors Liu and Acharya explored a variety of active learning strategies in our traditional as well as hybrid learning courses (Liu et al. 2017). In this section, we present technology-independent pedagogy, learning objectives, and instructional design in current hybrid courses and future MOOCs for the two remedial math courses MA and MC.

Course Materials Adapted from a Hybrid Course

Both the MA and MC courses are organized as 12 lessons with 24 lessons in total. Six lessons of MC were developed recently for the remedial math course. The rest of the lessons are adapted from the course materials of Mathematical Modeling and Simulation (MMS http://modelsim.wordpress.com). The MMS course is one of six cyberlearning courses sponsored by the National Science Foundation (NSF) 2014-2020 for a STEM education program (IUSE 13940667 2014-2016, and 1626602 2017-2020). Two instructors Matthew Ikle at Adams State University and Liu at ERAU, who served as PI and Co-PIs of the grants above, took turns to develop, review, and teach the two courses; one is the MMS, and the other course is Data Mining and Visualization. Each of the 12 lessons includes lecture notes, 7-12 pages in e-book form, PowerPoint slides, and 4-6 short videos, each around 10 minutes, uploaded on YouTube. While it takes time to learn how to use GIFT to transfer course materials into adaptive MOOCs, we have almost done the most time-consuming task in developing course materials. Most of the course materials and other subject details can be found in cited references or the link above.

Student-Centered Pedagogy

As a form of hybrid synchronous learning with both local and online students from multiple colleges, these courses use queries to scaffold active learning and facilitate flipped classrooms (Liu et al., 2017). The MMS course was used for independent studies online for more than 20 students over the last six years. The first author mentored three to nine students per year by communicating with each student around half an hour per week. MMS uses constructive learning, e.g., using the learned concepts to model open-ended problems and using data to validate models, to promote students' problem-solving ability. Most importantly, all six courses use course-based undergraduate research experience (CURE) on relevant real-world problems, promote teamwork, and enable the use of computational tools to motivate students to gain a deep understanding of concepts. Recently, the authors Liu H, Spector JM, Ikle M. 2018 explored how to use computer technologies to facilitate model-based collaborative learning and help online learners gain peer-support.

Integrated Instructional Design for the Online Remedial Math Courses

The instructional design of MA and MC follows the integrated design principle proposed by the Online Learning Initiative (OLI, **https://oli.cmu.edu**) of Carnegie-Mellon University (CMU). The OLI emphasizes alignment among three main course components – learning objectives, assessment, and instructional activities - to ensure an internally consistent structure. For example, the learning objectives are organized as a hierarchical tree with four levels: (1) course at the top branching into three units, (2) unit next with three lessons, (3) lesson level, and (4) concepts and skills in each lesson at the bottom. The student-centered pedagogy and OLI design emphasize that a learning objective should be stated about the *competence* from the perspective of students. That is, the sentence should start with "the students" and then predicate the measurable learning outcomes. For example, a learning objective of MA states: "*The students should be capable of using the eigenvectors of a covariant matrix to identify principal components of a numerical data frame*." The assessment for each lesson includes homework assignments, 3-5 discussion questions for blog discussion, and 3-5 embedded quizzes. Each video is associated with a section of text focusing on either a concept or a procedure. It is then followed by either a conceptual quiz on a Google Form with immediate feedback or in-class exercises with the answers provided at the end of lecture notes. In general, each concept or procedure will be transformed into an xAPI statement of the competence framework. The outcome of its associated assessment will become a branching condition for a learner state transition of the GIFT EMAP. At the current stage of our rule-based adaptive learning design, concept progress maps are critical inputs to our adaptive learning strategies.

Transferring the Student-Centered Pedagogy to Adaptive Learning MOOC

The teachers of small face-to-face classes have the advantage of empirically evaluating the students in terms of three aspects: their cognitive competence, affective trait, and volition. For online education, student-centered pedagogy depends on learner models and competence models that are built on the data collected from students. Using the GIFT tutoring system and MOOC makes data collection easier. We started first by asking questions about what to measure, how to best measure, and what to do with the results. However, building student models that reflect the affective and motivational traits of students will be further explored. We identified a list of tasks based on the challenges of active online learning from Hannafin, M. and Hannafin, K. M. 2010. Table 1 below illustrates the problems, relevant data collection, and potential solutions.

Table 1: Problems, Data Collection, and Potential Solutions for Student-Centered Online Learning

Students lost in the hypertext, and disoriented	Use the GIFT tutoring to identify and recommend learning activities based on their learning objectives and data-driven formative assessment.
Misconception, canonical vs. individual interpretation of meanings	Build libraries for the concepts and misconceptions for each lesson and course, and use GIFT surveys to assess the understanding of students. Peer-learning and blog discussion will be encouraged.
Preknowledge, past experience, and belief,	The records in LMS and Learning Record Store (LRS) for each learner, and GIFT surveys will be used to build and update the learner models.
Adjust cognitive load	Students' surveys and future work to use sensor module of GIFT
Monitor the affective traits of students and intervene timely	Use web-bot called BotCaptain to collect data and monitor teamwork to assess student affective traits. (Liu, H, Warner, T., and Ikle, M, 2020)
Evaluate student motivation	Use the MUSIC (eMpower, Useful, Success, Interest, and Caring) Survey Inventory by Jones, B. D. 2020

TOOL CONFIGURATION AND PRELIMINARY SYSTEM DESIGN

Universal Data exchange and Learning Tool Interoperability (LTI) for long-term sustainability are the primary concerns in the development of platforms and tools. GIFT tutoring system is our top choice because it is an integrated component of the Future Learning Ecosystem (Duncan, A. G., 2019). Moodle is selected as a Learning Management System (LMS) for the two courses because Moodle is an LTI tool that also facilitates the xAPI (https://github.com/adlnet/xAPI-Spec) standard for data exchanges. Moodle and GIFT, can not only directly exchange data as consumers or producers but also share mutually accessible data through the xAPI data format (Hruska1, Medford, and Murphy, 2015) and Learning Record Stores (LRS). In this section, we present the tool configuration, content deployment, and use cases as well as the components that map the pedagogy and instructional design into the artifacts of GIFT tutoring system.

Tool Configuration, Content Deployment, and Use Cases

For all three tools and their services, we have to either install them in our local servers or use external cloud servers. Our retrospection indicates that three factors should be evaluated first: (1) The administrator privilege for connecting to other necessary tools and services, (2) allowable storage space and control of the file management, and (3) the cost for the size of potential needed services. We installed GIFT and Moodle in the local server of ERAU, and enrolled the GIFT and Moodle cloud services as well. In addition, we registered the SCORM (Sharable Content Object Reference Model) cloud for the LRS and configured the LRS at the EC2 server of Amazon Web Service (AWS) cloud. Since we cannot obtain administrator privilege to configure the LRS connection through XML files for GIFT or Moodle cloud, we have to use both tools installed in the server of ERAU. Next, we need to set up a public accessible end-point http://www.ecodolphin.org for our local server so that GIFT and Moodle can connect to each other and provide cloud services. Both of them are connected with the LRS hosted by the SCORM cloud and the AWS EC2 server for learner data exchange and storage. Shown in Figure 1, Moodle serves as the interface between students, teachers, and cloud services. The course contents are posted in the Moodle site mostly through linked pages, while the online exercises, formative assessment, and content recommendations are delegated to the GIFT. The student learning records and stable learner states are stored in LRS.

Domain Knowledge Files, Competence Model, and Learner State Transition

The adaptive features of MA and MC depend on the data collected in table 1 and the transition logic defined by the Merrill Quadrant of EMAP. The Domain Knowledge Files (DKF) are still under construction. The task of translating the quizzes and surveys from Google forms into GIFT surveys is to be done soon. We only consider the cognitive levels of students for the learner model at this stage and use the default pedagogical model. The learner state transition uses the three default levels - below, meet, or exceed, for the students to change state and move on to the next learning activity. Comparing other domains, the competence models of mathematical subjects have two characteristics. Firstly, the assessments can be easily standardized, and the results can be objective because mathematics is a well-structured formal language, and its statements need to be refutable in terms of right or wrong. The assessment of MA and MC for our first delivery are the quizzes and in-class exercises from the GIFT survey services. The second characteristic relates to how the concepts of the DKF are deeply nested and cross-referred due to the profound nature of advanced mathematics subjects. This is the primary cause of challenges to develop adaptive MOOC for MA and MC. To build a competence model, we need to specify the concept tree. Shown in Figure 2, the statement "*The students should be capable of using the eigenvectors of a covariant matrix to identify principal components of a numerical data frame*" depends on many parent learning objectives across many units and lessons. Figure 3 shows three referenced sample xAPI statements and a nested sub-statement for concepts related to principal component and eigenvectors along a path (4->4.1->4.1.2-> 4.1.2.1). A Bayesian model based on the concept tree in the learning space (Falmagne, J., et al., 2000) can be used to identify the most likely parent concepts until all the misconceptions are identified and remediated.

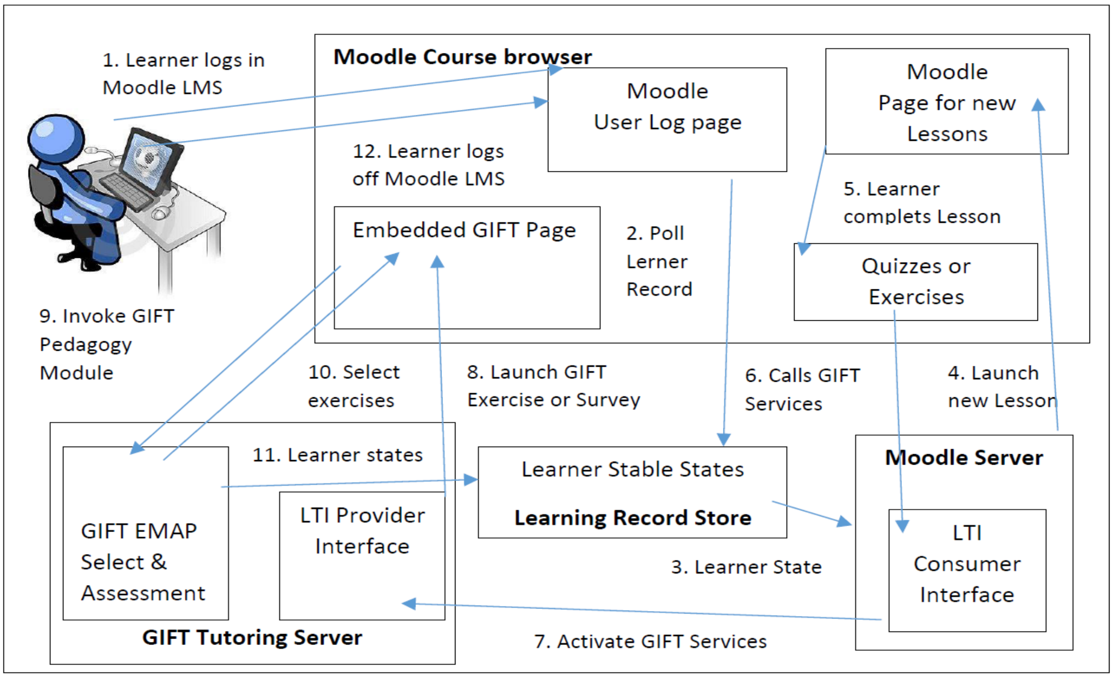


Figure 1: Use Cases and Services provided by GIFT, Moodle LMS, and LRS

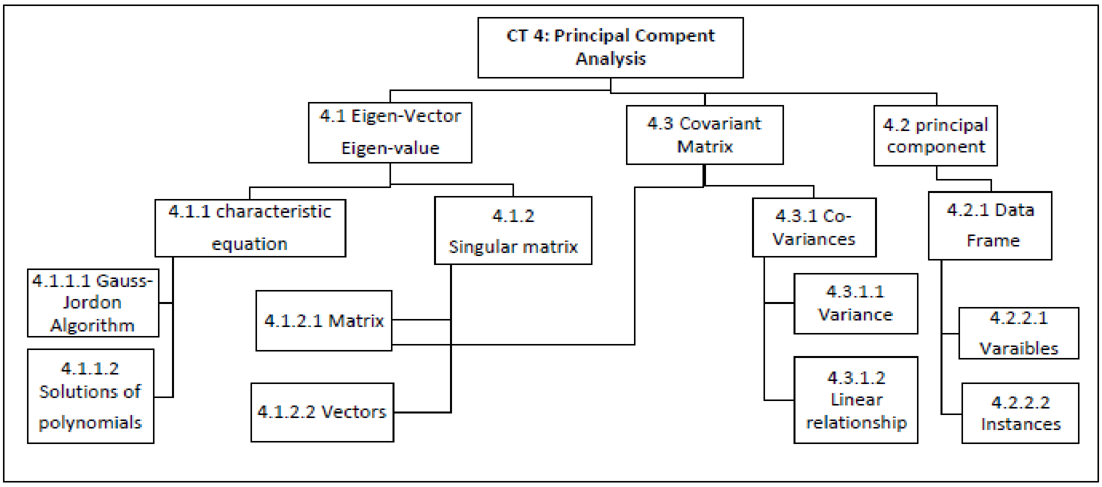


Figure 2: Concept Tree for Dependent Relationship for Principal Component Analysis

```
#Full xAPI Statements

{

    "id": "12345678-1234-5678-
1234-567812345678",

    "actor":{


"mbox":"bernart1@my.erau.edu",

        "Name":"Timothy Bernard"

    },

    "verb":{


"id":"http://adlnet.gov/expapi/v
erbs/experienced",

        "display":{

            "en-US":"Ranked"

        }

    },

    "object":{


"id":"http://example.adlnet.gov/
xapi/example/activity",

        "definition": {
```

```
#skipped the actor clauses.
{
    "actor":{......
    },
    "verb":{

"id":"http://adlnet.gov/expapi/verbs/expe
rienced",
        "display":{
            "en-US":"Distinguished"
        }
    },
    "object":{
        "objectType":"StatementRef",
        "id":"8f87ccde-bb56-4c2e-ab83-
44982ef22df0",
        "definition": {
            "name": { "en-US": "Singular
Matrix" }
          }
    }
}
{
        "actor": {......
        },
        "verb" : {

        "id":"http://adlnet.gov/expapi/ver
bs/experienced",
                "display":{
                        "en-US":"Computed
EigenVectors and EigenValues"
                }
        },
        "object": {
                "objectType":
"SubStatement",
                "actor" : {......
                },
                "verb" : {

        "id":"http://adlnet.gov/expapi/ver
bs/experienced",
```

```
            "name": { "en-US":
"Principle Components in Data
Frame" }

          }

    }

}

{
    "id": "8f87ccde-bb56-4c2e-
ab83-44982ef22df0",
    "actor":{

"mbox":"bernart1@my.erau.edu",
        "Name":"Timothy Bernard"
    },
    "verb":{

"id":"http://adlnet.gov/expapi/v
erbs/experienced",
        "display":{
            "en-US":"Computed"
        }
    },
    "object":{

"objectType":"StatementRef",
        "id":"12345678-1234-
5678-1234-567812345678",
        "definition": {
            "name": { "en-US":
"EigenVectors and EigenValues" }
          }
    }
}
```

```
                        "display":{
                                "en-
US":"Distinguished"
                        }
                },
                "object": {
                        "objectType":
"Activity",

        "id":"http://adlnet.gov/expapi/ver
bs/experienced",
                        "definition": {
                                "name" : {

        "en-US":"Singular Matrix"
                                }}}}}
```


Figure 3. Four Related xAPI Statements and Substatement for Principal Component Analysis

CONCLUSIONS AND FUTURE WORK

Developing effective cognitive tutoring and adaptive online courses is time-consuming and difficult. Closely related literatures can be dated back several decades. Nevertheless, the existing computer-based applications to adaptive online learning are still limited to serve K12 to low-level college mathematics courses. The progress of technology and applications are slow but steadfast to improve the quality of online courses and reduce the cost of human intervention. GIFT Tutoring system and its associated tools such as LRS and LTI LMS provide a platform for instructors and educational technologists to develop learner models and DKF that are reusable and extendable. The MA and MC remedial courses are built on tested hybrid learning courses based on student-centered pedagogy. Online courses have gradually increased features to automate repetitious tasks but will require instructors to answer verbal questions and grade final tests for many years. Our effort, though in an infant stage, extends the educational technology to serve advanced mathematics topics.

We envision our future work to include three major components. The first component consists of work on learning assessment. The online feedback for quizzes we have so far is limited to multiple choice answers in Google Forms. Such quizzes can help assess conceptual understanding but are inadequate in evaluating skills and procedural knowledge. The in-class exercises and homework problems are unlikely to be operationalized like those of the Aleks because their procedural complexity is intractable to computer algorithms. The reference answers for student to self-check are either numerical and symbolic answers or manually created procedures. We are working on adding Python and R code for the students to self-check. Soon, we will explore how to plug Juypter Notebook into the GIFT system so that the Python and R programs can be utilized to evaluate procedural knowledge and skills. Secondly, we aim to build a learner model with xAPI statements for the two courses. Because we do not have the competence models, we manually created a few xAPI statements to test the integration of tools. Next, we will explore an xAPI simulation tool from YetAnalytics to help design and test the data-driven assessment models. A tool to create xAPI statements from the concept tree is definitely desirable in future. Our third goal is related to the ongoing research of the first two authors in developing a web-bot to promote a peer-learning environment called iCycle (Liu, Warner, and Ikle, 2020). We are excited to know that the GIFT 2020-1 release will include more team modeling features. We aim to learn the GIFT team model and explore how to incorporate our web-bot app into GIFT to collect affective and motivational data based on teamwork. This WIP is a small step of our expedition in GIFT Tutoring and xAPI-based interoperable learning systems. By starting with these promising technologies within the architecture of future learning ecosystems, we can ensure that the growth of our applications and courseware are only limited by our committed time.

ACKNOWLEDGEMENT

This research was partially sponsored by the National Science Foundation (NSF) of the USA under the IUSE Grant 1626602 from 2017-2020.

REFERENCES

Aleven, V. et al., (2019), Towards Deeper Integration of Intelligent Tutoring Systems: One-way Student Model Sharing between GIFT and CTAT, Proceedings Cognition and Student-Centered, Web-Based Learning.

Aleven, V., Mclaren, B., Roll, I., and Koedinger, K., "Toward Meta-cognitive Tutoring: A Model of Help Seeking with a Cognitive Tutor", Int. J. Artif. Intell. Ed., Vol. 16, No. 2, pp. 101–128, 2006.

Falmagne, J., et al., (2000), Assessing Mathematical Knowledge in a Learning Space: Validity and/or Reliability, https://www.aleks.com/paper_psych/Validity_in_L_Spaces.pdf.

Hannafin, M., and Hannafin, K. M. (2010), Cognition and Student-Centered, Web-Based Learning: Issues and Implications for Research and Theory, Chapter 2, J.M. Spector et al. (eds.), *Learning and Instruction in the Digital Age*, DOI 10.1007/978-1-4419-1551-1_2, @ Springer Science Business Media, LLC.

Chi, Michelene, T. H., (2009), Active-Constructive-Interactive: A Conceptual Framework for Differentiating Learning Activities, Topics in Cognitive Science 1, 73-105. https://ineer2014.rtu.lv/sites/default/files/Paper_117.pdf

Duncan, A. G., (2019), The Biggest Problem We have is the lack of Connected Infrastructure Across Postsecondary Learning Systems, Ed., by Walcutt, J. J., and Schatz Sae, "Modernizing Learning, Building the Future Learning Ecosystem," 109 -128.

Goldberg, B., Hoffman, M. and Tarr, R. (2015). Authoring Instructional Management Logic in GIFT Using the Engine for Management of Adaptive Pedagogy (EMAP). *Design Recommendations for Intelligent Tutoring Systems: Authoring Tools and Expert Modeling Techniques*, 319-334.

Hruska1, M., Medford, A., and Murphy, J., (2015) Learning Ecosystems Using the Generalized Intelligent Framework for Tutoring (GIFT) and the Experience API (xAPI), http://problemsolutions.net/images/2015-GIFT-and-xAPI-AIED_FINAL.pdf.

Jones, B. D. 2020, User guide for assessing the components of the MUSIC® Model of Motivation. Retrieved from http://www.theMUSICmodel.com.

Liu, H., Ludu, A., Klein, J., Spector, M., and Ikle, M., (2017), Innovative Model, Tools, and Learning Environments to Promote Active Learning for Undergraduates in Computational Science & Engineering, the Journal of Computational Science Education, Volume 8 Issue 3 pp. 11 – 18, https://doi.org/10.22369/issn.2153-4136/8/3/2.

Liu H, Spector JM, Ikle M. (2018), Computer technologies for model-based collaborative learning: A research-based approach with initial findings, Comput Appl Eng Educ;1–10, https://doi.org/10.1002/cae.22049.

Liu, H, Warner, T., and Ikle, M, (2020) Harness Big Data by iCycle - Intelligent Computer-supported Hybrid Collaborative Learning Environment, to appear on International Journal of Smart Technology and Learning.

Ragusa, R., Hoffman, M., and Leonard, J. (2013) Unwrapping GIFT, A Primer on Developing with the Generalized Intelligent Framework for Tutoring http://ceur-ws.org/Vol-1009/0702.pdf .

Sottilare, R., Brawner, K., Goldberg, B. & Holden, H. (2012) The Generalized Intelligent Framework for Tutoring (GIFT). US Army Research Laboratory.

Sottilare, R., Graesser, A. C. , Hu, X., and Goldberg, B. S., (2017) Ed, "Design Recommendations for Intelligent Tutoring Systems, Volume 5, Assessment Methods," Adaptive Tutoring Series volume 5.

ABOUT THE AUTHORS

Hong Liu *was awarded a Ph.D. in Mathematics and MS in Computer Science at the University of Arkansas, Fayetteville in 2000. He serves as a professor in Mathematics and Computing at Embry-Riddle Aeronautical University. He served as PI and Co-PI of 14 sponsored projects, and published numerous articles in mathematics, data science, and STEM education. His current research interest is to develop a smart learning environment to promote peer learning.*

Timothy Bernard *graduated from Embry-Riddle Aeronautical University with a bachelor's degree in Cybersecurity. Currently, he is a graduate student in Master of Science in Data Science and serves as a research assistant to develop software apps and maintain software systems for learning technology.*

Keshav Acharya *was awarded a Ph.D. in Mathematics from University of Oklahoma in Norman, OK in 2013. He serves as an assistant professor in Mathematics at Embry-Riddle Aeronautical University. His research interests are in Mathematical Analysis and Applications and Mathematics Pedagogy. He has published several articles in Mathematics and Pedagogy.*

CbITS Authoring Tool in GIFT

Lijia Wang[1], Keith Shubeck[1], Genghu Shi[1], Liang Zhang[1], Xiangen Hu[1,2]
The University of Memphis[1], Central China Normal University[2]

INTRODUCTION

With the fast spread of the World Wide Web and the development of the hardwares, people released the power of the new technologies and tried to apply them to human learning and education. The concept of computer based instruction (CBI) was introduced to human education around the 1950s. CBI somehow defined the "non-intelligent" which required an individual to achieve the learning objectives and access to the tailored learning content with right difficulty under good efficiency. In the forms of the learning content, computer aided instruction (CAI) showed the high potential of multimedia during the learning (Murray, 1999). Unfortunately, neither CBI or CAI could make an intelligent system for their beneficiary. Advanced distributed learning (ADL) claims that the "intelligent", or the definitions of intelligent tutoring systems (ITSs) allows students to ask open questions and generate instructional material on the demand of the user (Fletcher, 2005).

ITS research and development have been a focus for generations of learning scientists since the 1960s. ITS applications are proven effective and efficient in delivering learning to all types of learners (Fletcher, 2005; Vanlehn, 2006). Despite the success of ITS in certain areas, "Intelligent Tutoring Systems (ITS) promised to the dream of a truly adaptive learning experience almost 30 years ago, but in spite of millions of dollars spent and promising student learning outcomes, they have not flourished" pointed by Bill Ferster (2017). He observed that one of the reasons is the lack of authoring tools. This idea is also supported by other groups (Sottilare *et al.*, 2015; Cai, Hu and Graesser, 2019; Hu, Cai and Graesser, 2019). Furthermore, the learning content is not only limited to the text but also consists of multimedia like images, videos, virtual meetings, and other online interactions. Authoring and managing this content is costly and requires a great deal of effort. Beyond the content's form, an ITS may also have different presentation and pedagogical strategies. Conversation based Intelligent Tutoring Systems (CbITS) is an excellent example here. It is easy to underestimate the amount of effort and expertise needed to author content for a CbITS. To make a good CbITS, content authoring will involve many experts like domain experts, linguists, instruction designers, programmers, artists, and computer scientists (Cai, Hu and Graesser, 2019). Content authoring is so time consuming and resource intensive, it should remain an important focus in the ITS field. Simply put, the training period cannot be ignored, either. These high entry requirements prevent ITS from being more widespread. Thus, a good authoring tool can not only help create the learning content, but also reduce the cost from both labor and time, as well as the cooperation among the teams.

There are many factors that can improve the authoring tool. A potential solution to this problem is through a general framework and systematic approach to authoring content in an ITS (Sottilare *et al.*, 2017). A general framework, or a standard can greatly reduce the cost of system integration when multi learning materials would be used. Under the standards, the learning objects can be well managed, shared and re-used among different groups in the same community. Therefore, it is very important to find such a framework. General 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*, no date). The "service-oriented" integrative capacity of GIFT makes it possible to integrate existing ITS systems, such as AutoTutor (Nye, 2013; Ventura *et al.*, 2015), into GIFT. Although such an "integration" capacity of GIFT makes it possible for other ITSs to be part of GIFT applications as add-on services, the issues of Authoring for the ITS services remain a headache for the content authors.

AutoTutor

AutoTutor (Graesser *et al.*, 1999, 2004, 2012; Person *et al.*, 2003; Graesser, 2016) is a computer tutor that helps students learn by holding a conversation in natural language. The main features of AutoTutor include a simplified student model, animated agents with text-to-speech, and semantic analysis services to hold conversations with learners. AutoTutor can track the cognition and emotions of the student and responds in a manner that adapts to the student. The applications of AutoTutor include but no limit to computer literacy (Graesser *et al.*, 1999; Person, 2003), physics (Graesser *et al.*, 2003; Rus *et al.*, 2014), mathematics (Nye *et al.*, 2015, 2018), electronics (Graesser *et al.*, 2018; Morgan *et al.*, 2018), adult literacy (Cai *et al.*, 2015; Graesser *et al.*, 2016), critical thinking (Wallace *et al.*, 2009; Graesser *et al.*, 2010), and biology (Olney, Graesser and Person, 2010). AutoTutor has been funded over $30M in the last two decades by NSF, IES, ONR, Army, DoD, and other government and local funding agencies. The effectiveness of AutoTutor in delivering learning is proven (Nye, Graesser and Hu, 2014; Nye *et al.*, 2015; Graesser, 2016). AutoTutor has many features and applications, such as AutoTutor's learner characteristic curve (LCC) (Hu and Martindale, 2008; Hu, Morrison and Cai, 2013; Sullins, Craig and Hu, 2015). Autotutor has been used in schools (Rus, Niraula and Banjade, 2015), part of the general framework (Nye, 2013; Ventura *et al.*, 2015), core component of major funded projects (Swartout *et al.*, 2016). Most recently, AutoTutor has been studied as part of a broader class of learning environments called Adaptive Instructional Systems (AIS) (*Adaptive Instructional Systems (C/LT/AIS) P2247.1*, no date). Researchers started to use AutoTutor as an example ITS that can self-improve (Hu, Cai and Graesser, 2019).

The AutoTutor module can be found in the GIFT on both desktop control panel and GIFT Authoring Tool (GAT) on cloud server (Figure 1). AutoTutor Script Authoring Tool (ASAT) is third-party based service which is not integrated in GIFT GAT directly (Hoffman and Ragusa, 2015). In an effort to further integrate AutoTutor into GIFT, our goal is to create AutoTutor scripts within the GIFT authoring framework. With this accomplished, the entire process from authoring AutoTutor content to its application can occur inside GIFT. To ease the integration of the AutoTutor authoring process into GIFT, we are exploring a new way for authoring AutoTutor's XML files, which operate as AutoTutor's content scripts. The proposed changes will allow us to explore collaboration during the authoring process within GIFT. To be more specifical, we would like to explore collaboration during the authoring process in the GIFT.

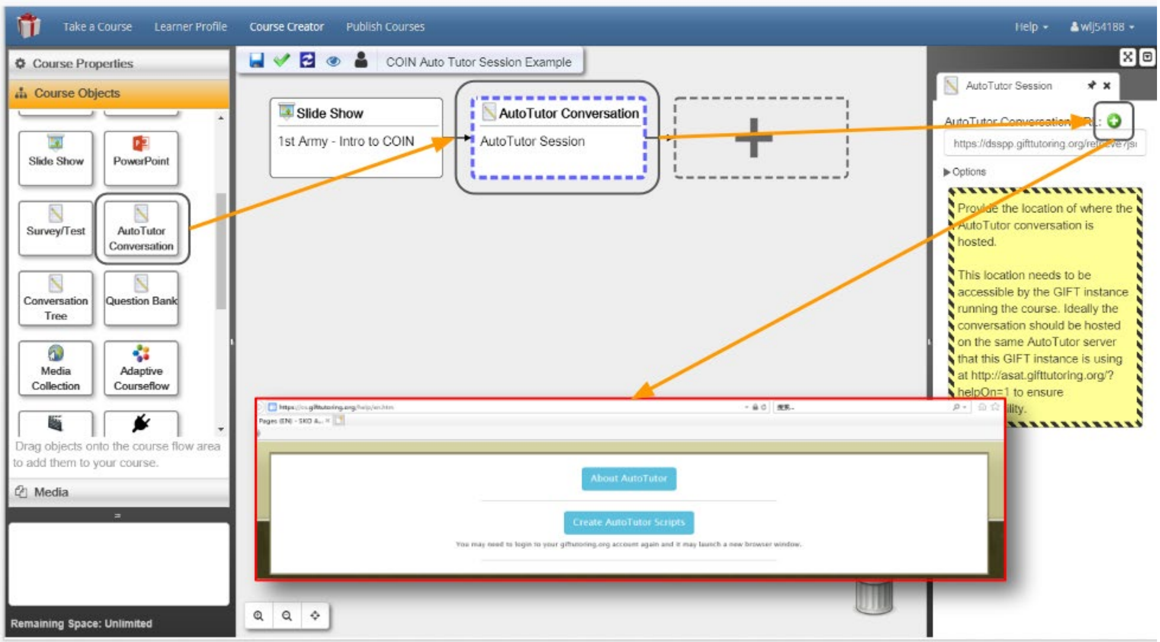


Figure 1. AutoTutor Script Authoring Tool in the GIFT Authoring Tool

IMPROVING AUTHORING TOOL

AutoTutor Script

An AutoTutor (Nye, Graesser and Hu, 2014) conversation session starts by loading an AutoTutor script into the AutoTutor Conversation Engine (ACE). The main job of ACE is to evaluate user input (including text, speech, or action), and calculate performance scores based on the content in an AutoTutor script. Figure 2 provides an example of an AutoTutor script, in terms of complexity. AutoTutor uses the design of the Expectation misconception tailored (EMT) dialogue to present knowledge and achieve learning goals. An AutoTutor script will contain multiple sections based on the modules setting. In this example, there are five core components in the AutoTutor script: Agents, SpeechActs, RigidPacks, TutoringPacks, and Rules, each with their own unique function that affects the learning conversation's performance.

- Agents: Define the personality of the avatars by stylistic languages used for feedback (positive, neutral, negative).
- SpeechActs: The SpeechActs section defines the regular expression used to capture certain user feedback types with strong keywords like "Yes" and "No".
- Rigid Packs: Specify pre-determined interactions such as opening or closing of tutoring sessions.
- Tutoring Packs: implement Expectation-Misconception Tailored (EMT) dialog of AutoTutor (Graesser *et al.*, 2005).
- Rules: A list of "if-then" that guide the interaction of AutoTutor when it interacts with learners.

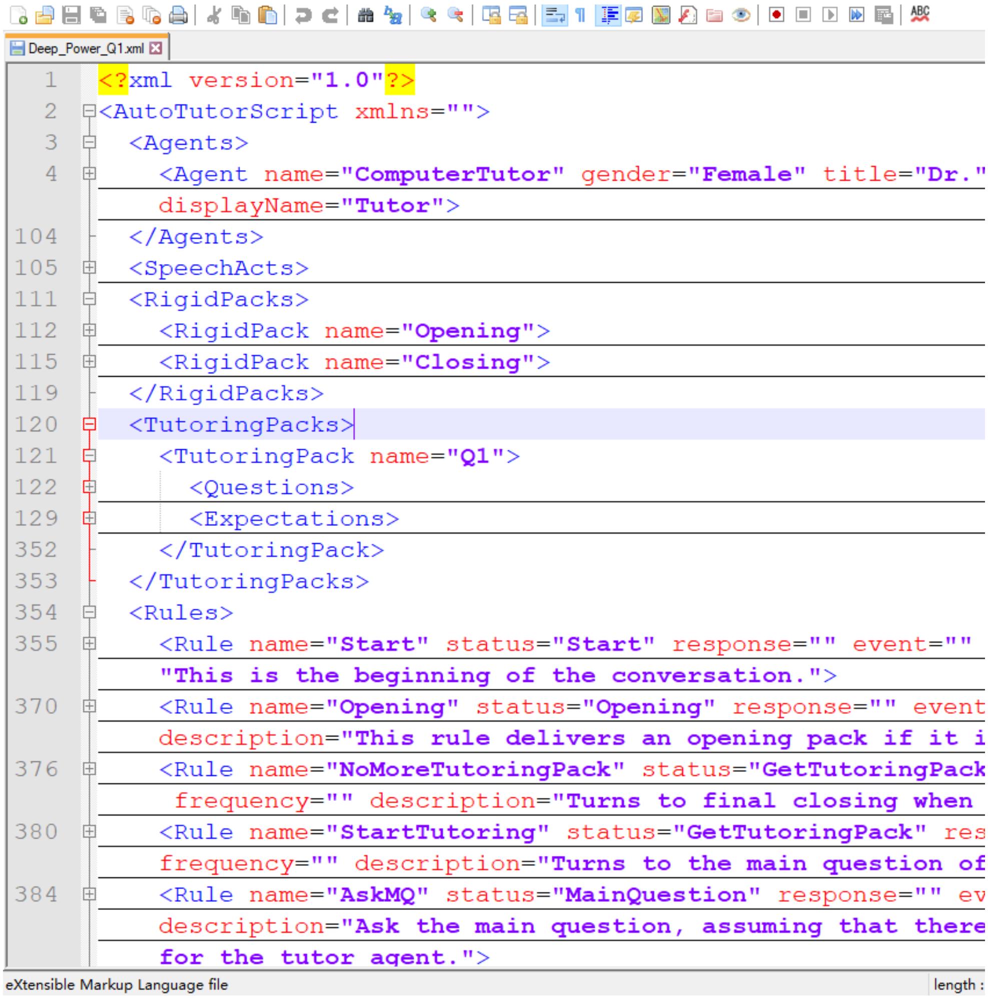


Figure 2. An ElectronixTutor AutoTutor Script Example with five major sections

In addition to these core components, an AT script can also include multimedia information, interaction triggers (e.g., mouse movements and click events), education metadata, and life cycle. Developing a quality AutoTutor script requires a good understanding of how these components relate to each other and to the tutoring interactions as a whole. This is understandably a challenge for newcomers. Ideally, domain experts, linguists, and instruction designers can work together as a team within the same authoring framework. In order to have people develop high quality scripts, much effort has been put into the AutoTutor Script Authoring Tool (ASAT). ASAT is currently using a Flash browser plug-in which no browser will support after December 2020. While collaboration is an important part of developing a quality AutoTutor script, the collaboration process within ASAT is limited at best. To simplify the process and to increase the approachability, cooperation, and overall usage of the ASAT, we are proposing a new iteration of AutoTutor's authoring tool by using the advantages the GIFT system provides.

GIFT XML Editor

There are many ways to handle learner data, course information, and learning content. They can be roughly divided into two categories (Damon Regan, Elaine M. Raybourn, and Paula J. Durlach, 2013): XML and RDF files, or HTTP, JSON, REST files. The major difference between these categories is the barrier to entry. XML and RDF relies on standards (e.g. W3C) which have a higher barrier while HTTP, JSON and REST have a better facilitation to communicate and change data.XML has been chosen to push the standards of learning with shareable knowledge objects. XML is used to build many application programming interfaces (APIs) in both human and machine readable format. GIFT XML editor can only be accessed by the desktop control panel. There are many XML-based applications available from the control panel. But all those applications will trigger the XML editor. The difference is the schema used. Each XML schema definition (XSD) will generate the unique content editor to create XML files for different learning modules in the GIFT system. And the tight integration between XML and XSD files can greatly improve the stability of the GIFT system by reducing the unnecessary errors and inconsistencies without changing coding (Hoffman et al., 2015). GIFT XML editor has the basic graphical user interface for authoring XML which usually occupies more than 50% of total resources during the development of other learning systems. The clear and simple outlook with tree-view can well show the relationship among each branch and their entities.

AutoTutor Script Schema Definition

Based on the existing GIFT XML editor, we develop the XSD file for a basic AutoTutor script. The schema defines the five major components related to content: Agents, SpeechActs, RigidPacks, TutoringPacks, and Rules. After applying the XSD file, the GIFT XML editor will look like Figure 3. A detailed explanation for each element and attribute can be found under the text box on the left bottom corner. The process of creating AutoTutor is also explored and documented (Franceschetti *et al.*, 2001; Cai, Hu and Graesser, 2019).

As Figure 3 showed, a new AT script created using GIFT XML editor was marked in red. It means the script was invalid compared to the schema. Any invalid elements are shown with red color while the elements highlighted with blue is a valid one. Only a valid script can be opened, or saved in this XML editor. This property makes it a good XML validator.

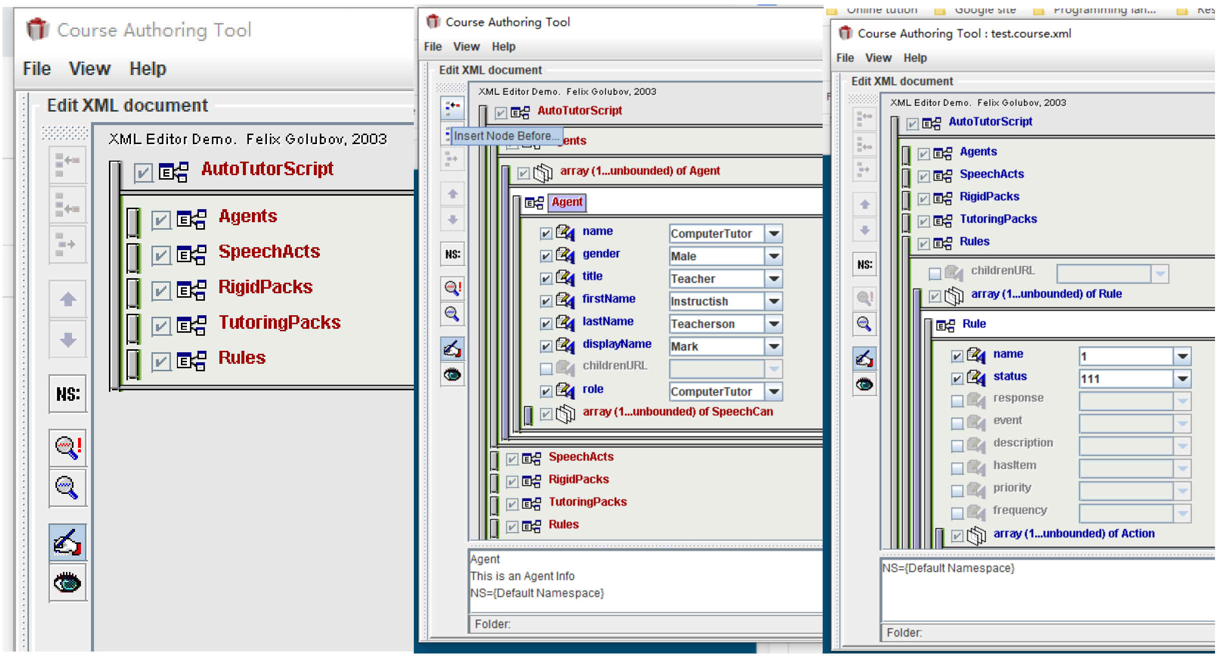


Figure 3. AutoTutor Script Authoring Tool in the GIFT XML editor using schema: Left image: Initial status after opening the GIFT XML editor; Middle image: Insert a new node and its attributes in Agents; Right image: Fulfilled AutoTutor script

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

The XSD of AutoTutor scripts provides a method to author the AutoTutor sharable knowledge objects. Further integration of AutoTutor into GIFT, will better serve the community by making authoring AutoTutor scripts more accessible and less time consuming for new users.

As mentioned previously, an AT script still has many sections that can be added into XSD to adopt other modules. Therefore, future work should add new modules that support additional essential functions including LCC, enhanced multimedia module, and education metadata. Additionally, more work is needed to improve the authoring of new customized rules. Rules may be visualized and modified using a similar mechanism, like the conversation tree tool provided in the GIFT. Finally, more work is needed to improve the collaborative process of authoring AutoTutor content. Improvements in this area should open up potential research opportunities concerning the collaborative content authoring process.

REFERENCES

Adaptive Instructional Systems (C/LT/AIS) P2247.1 (no date). Available at: https://site.ieee.org/sagroups-2247-1/ (Accessed: 12 May 2020).

Cai, Z. et al. (2015) ‘CSAL AutoTutor: integrating rich media with AutoTutor’, Intelligent Framework for …. books.google.com. Available at: https://books.google.com/books?hl=en&lr=&id=EaeDDAAAQBAJ&oi=fnd&pg=PA169&dq=graesser+nye+hu&ots=65XxFrKmRn&sig=g7kg-G4iiq7s3jpAxdmKm2zyLeE.

Cai, Z., Hu, X. and Graesser, A. C. (2019) ‘Authoring Conversational Intelligent Tutoring Systems’, in Adaptive Instructional Systems. Springer International Publishing, pp. 593–603.

Damon Regan, Elaine M. Raybourn, and Paula J. Durlach (2013) ‘Learner Modeling Considerations for a Personalized Assistant for Learning (PAL)’, in Robert Sottilare, Arthur Graesser, Xiangen Hu, Heather Holden (ed.) Design Recommendations for Intelligent Tutoring Systems. the U.S. Army Research Laboratory., pp. 215–225.

Ferster, B. (2017) Intelligent Tutoring Systems: What Happened? - eLearning Industry, eLearning Industry. eLearning Industry. Available at: https://elearningindustry.com/intelligent-tutoring-systems-what-happened (Accessed: 13 May 2020).

Fletcher, J. D. (2005) The Advanced Distributed Learning (ADL) Vision and Getting From Here To There. INSTITUTE FOR DEFENSE ANALYSES ALEXANDRIA VA. Available at: http://www.dtic.mil/docs/citations/ADA452053.

Franceschetti, D. R. et al. (2001) ‘Development of physics text corpora for latent semantic analysis’, in Proceedings of the 23rd annual conference of the cognitive science society. Erlbaum Mahwah, NJ, pp. 297–300.

Graesser, A. et al. (2010) ‘Critiquing Media Reports with Flawed Scientific Findings: Operation ARIES! A Game with Animated Agents and Natural Language Trialogues’, in Intelligent Tutoring Systems. International Conference on Intelligent Tutoring Systems, Springer, Berlin, Heidelberg (Lecture Notes in Computer Science), pp. 327–329.

Graesser, A. C. et al. (1999) ‘AutoTutor: A simulation of a human tutor’, Cognitive systems research. Elsevier, 1(1), pp. 35–51.

Graesser, A. C. et al. (2003) ‘Why/AutoTutor: A test of learning gains from a physics tutor with natural language dialog’, in Proceedings of the Annual Meeting of the Cognitive Science Society. cloudfront.escholarship.org. Available at: https://cloudfront.escholarship.org/dist/prd/content/qt6mj3q2v1/qt6mj3q2v1.pdf.

Graesser, A. C. et al. (2004) ‘AutoTutor: a tutor with dialogue in natural language’, Behavior research methods, instruments, & computers: a journal of the Psychonomic Society, Inc, 36(2), pp. 180–192.

Graesser, A. C. et al. (2005) ‘AutoTutor: an intelligent tutoring system with mixed-initiative dialogue’, IEEE Transactions on Education, 48(4), pp. 612–618.

Graesser, A. C. et al. (2012) ‘AutoTutor’, in P. M. McCarthy, &. C. B. (ed.) Applied natural language processing: Identification, investigation, and resolution. IGI Global, pp. 169–187.

Graesser, A. C. (2016) ‘Conversations with AutoTutor Help Students Learn’, International Journal of Artificial Intelligence in Education. Springer, 26(1), pp. 124–132.

Graesser, A. C. et al. (2016) ‘Reading comprehension lessons in AutoTutor for the Center for the Study of Adult Literacy’, Adaptive educational technologies for literacy instruction. books.google.com, pp. 288–293.

Graesser, A. C. et al. (2018) ‘ElectronixTutor: an intelligent tutoring system with multiple learning resources for electronics’, International journal of STEM education, 5(1), p. 15.

Hoffman, M. and Ragusa, C. (2015) ‘Unwrapping GIFT: A primer on authoring tools for the Generalized Intelligent Framework for Tutoring’, in Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym2), p. 11.

Hu, X., Cai, Z. and Graesser, A. C. (2019) ‘GIFT as a Framework for Self-Improvable Digital Resources in SIAIS’, of the 7th Annual GIFT …. books.google.com. Available at: https://books.google.com/books?hl=en&lr=&id=3MvKDwAAQBAJ&oi=fnd&pg=PA49&dq=GIFT+Framework+Self+Improvable+Digital+Resources+SIAIS&ots=yBd9x5Q1ll&sig=SnzhN4OkxnKmjIVTjzY8SmMU3ns.

Hu, X. and Martindale, T. (2008) ‘Enhance learning with ITS style interactions between learner and content’, in The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Hu, X., Morrison, D. M. and Cai, Z. (2013) ‘Conversation-based Intelligent Tutoring System’, Design Recommendations for Intelligent Tutoring Systems: Learner Modeling, 1, pp. 97–110.

Morgan, B. et al. (2018) ‘Electronixtutor Integrates Multiple Learning Resources to Teach Electronics on the Web’, in Proceedings of the Fifth Annual ACM Conference on Learning at Scale. New York, NY, USA: ACM (L@S ’18), pp. 33:1–33:2.

Murray, T. (1999) ‘Authoring intelligent tutoring systems: An analysis of the state of the art’. Available at: https://telearn.archives-ouvertes.fr/hal-00197339/.

Nye, B. D. (2013) ‘Integrating GIFT and AutoTutor with sharable knowledge objects (SKO)’, in AIED 2013 Workshop on GIFT. CEUR. gifttutoring.org, pp. 54–61.

Nye, B. D. et al. (2015) ‘Evaluating the Effectiveness of Integrating Natural Language Tutoring into an Existing Adaptive Learning System’, in Artificial Intelligence in Education. Springer International Publishing, pp. 743–747.

Nye, B. D. et al. (2018) ‘SKOPE-IT (Shareable Knowledge Objects as Portable Intelligent Tutors): overlaying natural language tutoring on an adaptive learning system for mathematics’, International Journal of STEM Education. Springer, 5(1), p. 12.

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

Olney, A. M., Graesser, A. C. and Person, N. K. (2010) ‘Tutorial Dialog in Natural Language’, in Nkambou, R., Bourdeau, J., and Mizoguchi, R. (eds) Advances in Intelligent Tutoring Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 181–206.

Overview - GIFT - GIFT Portal (no date). Available at: https://gifttutoring.org/projects/gift/wiki/Overview (Accessed: 21 February 2019).

Person, N. K. (2003) ‘AutoTutor improves deep learning of computer literacy: Is it the dialog or the talking head’, Artificial intelligence in education: Shaping the future of learning through intelligent technologies. books.google.com, 97, p. 47.

Person, N. K. et al. (2003) ‘Simulating human tutor dialog moves in AutoTutor’. telearn.archives-ouvertes.fr. Available at: https://telearn.archives-ouvertes.fr/hal-00197320/.

Rus, V. et al. (2014) ‘DeepTutor: towards macro- and micro-adaptive conversational intelligent tutoring at scale’, in Proceedings of the first ACM conference on Learning @ scale conference. ACM, pp. 209–210.

Rus, V., Niraula, N. B. and Banjade, R. (2015) ‘DeepTutor: An Effective, Online Intelligent Tutoring System That Promotes Deep Learning’. Available at: https://www.semanticscholar.org/paper/e0feeebed6d35740772958e96dfea219d4ce2d03 (Accessed: 12 May 2020).

Sottilare, R. et al. (2015) Design Recommendations for Intelligent Tutoring Systems: Authoring Tools and Expert Modeling Techniques. US Army Research Laboratory.

Sottilare, R. A. et al. (2017) ‘An updated concept for a Generalized Intelligent Framework for Tutoring (GIFT)’, GIFTtutoring. org. Available at: https://www.gifttutoring.org/attachments/download/2076/Updated%20Concept%20for%20the%20Generalized%20Intelligent%20Framework%20for%20Tutoring_9%20May%202017.pdf.

Sullins, J., Craig, S. D. and Hu, X. (2015) ‘Exploring the effectiveness of a novel feedback mechanism within an intelligent tutoring system’, International Journal of Learning Technology. Inderscience Publishers (IEL), 10(3), pp. 220–236.

Swartout, W. R. et al. (2016) ‘Designing a personal assistant for life-long learning (PAL3)’, in The Twenty-Ninth International Flairs Conference. aaai.org. Available at: https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS16/paper/viewPaper/12793.

Vanlehn, K. (2006) ‘The behavior of tutoring systems’, International journal of artificial intelligence in education. IOS Press, 16(3), pp. 227–265.

Ventura, M. et al. (2015) ‘NewtonianTalk: Integration of Physics Playground and AutoTutor using GIFT’, in AIED Workshops. pdfs.semanticscholar.org. Available at: https://pdfs.semanticscholar.org/a796/58178f4541d5bad17fddf3f734b69a9776a1.pdf.

Wallace, P. S. et al. (2009) ‘Operation ARIES!: A Computerized Game for Teaching Scientific Inquiry’, in AIED. niu.edu, pp. 602–604.

ABOUT THE AUTHORS

Lijia Wang *is Ph.D. candidate in the Department of Electrical Engineering and Computer Engineering at the University of Memphis. He received a M.S. in physics from the University of Memphis in May 2015. He has worked with project Personal Assistant for Lifelong Learning (PAL3) and ElectronixTutor from ONR under the supervision of his major professor, Dr. Xiangen Hu at the Institute for Intelligent Systems. His research interests include team interactions and modeling during learning, and content authoring.*

Keith Shubeck *is doctoral candidate in cognitive psychology at the University of Memphis. He received a Certificate of Cognitive Science and a M.S. in psychology in December 2015. He works with his major professor, Dr. Xiangen Hu, in the Advanced Distributed Learning Partnership Lab at the Institute for Intelligent Systems. His dissertation work compares interactive and vicarious tutoring frameworks in a conversation-based intelligent tutoring system for critical thinking and scientific reasoning.*

Genghu Shi *is a Ph.D. student in Experimental Psychology and also pursuing a master's degree in Statistics in University of Memphis. He works as a research assistant for Center for the Study of Adult Literacy in the Institute for Intelligent Systems. His research interests lie in Adult Literacy and the characteristics of conversations in intelligent tutoring systems. He is also interested in Data Science and Machine Learning. He earned a master's degree in Cognitive Psychology from Central China Normal University and a Bachelor's Degree in Computer Science from China University of Geoscience (Wuhan).*

Liang Zhang *is currently working toward the Ph.D. degree in the Department of Electrical Engineering and Computer Engineering, the University of Memphis, Memphis, TN. He also works as research assistant following Phil Pavlik in Optimal Learning Lab, at Institute for Intelligent System of University of Memphis. His research interests include educational data mining and learning analytics.*

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.*

Trainers and Fighter Pilots - Using GIFT with an OODA Loop Framework

Stacey Sokoloff, John Merrihew
Veloxiti, Inc.

INTRODUCTION

Trainers and Fighter Pilots

A good trainer, like a good fighter pilot, must react to changes in the environment. While a fighter pilot is concerned with enemy state, aircraft state, weather conditions, etc. a good military trainer pays close attention to learner performance and training scenario progression to adapt the training as necessary. Trainers are likely to benefit from applying decision making strategies that have proven successful for fighter pilots.

United States Air Force Colonel John Boyd developed a four-step decision making process called the Observe Orient Decide Act (OODA) Loop that describes a means of making timely and effective decisions in uncertain, dynamic environments. According to Boyd, a relatively large amount of data about the environment is *observed*, but much of the data is irrelevant. Decision makers use mental models to filter and process the data, enabling attention to be *oriented* on critical information that drives *decisions*, which often result in *actions*. Actions produce more data to be observed and the cycle continues.

Experienced trainers frequently apply the OODA Loop during training exercises. Field Manual (FM) 7.0 states "Once proficiency is achieved under the task's published conditions, leaders continually change the conditions". To achieve this, leaders/trainers must observe task performance, assess proficiency, decide how to change the task conditions, and act upon their decisions. The OODA Loop is highly applicable to adaptive training.

At first glance, integrating the Velox Framework, a Government Purpose Rights (GPR) framework for building OODA Loop based systems, with GIFT seems valuable and straight forward. In practice, integrating the two frameworks is possible and likely beneficial, but challenging to do well. The objective is to design an architecture that leverages the strengths of both frameworks. The GPR Simulation Monitoring and Automated Reporting Tool (SMART) project is working toward meeting the challenge.

SIMULATION MONITORING AND REPORTING TOOL (SMART)

Overview

SMART is an emerging architecture that processes data about the environment, trainee actions, and communication between team members to perform automated task performance assessment. The first version of SMART, FO (Forward Observer) SMART, provided automated individual performance assessment for adjust fire missions. It was integrated with Bohemia Interactive's Virtual Battle Space 3 (VBS3) simulation.

Since the verbal communication between a Forward Observer and a Fire Direction Center (FDC) is an essential part of an adjust fire mission, FO SMART used a Commercial Off The Shelf (COTS) Speech To Text (STT) tool and implemented a simple Natural Language Processing (NLP) capability to simulate and monitor verbal communication.

The second iteration of SMART, Squad SMART, is being developed to provide collective training to an infantry squad requesting indirect fire support. Squad SMART integrates the Android Team Awareness Kit (ATAK), a Government Off The Shelf (GOTS) Android app that promotes situation awareness, and, among many other features, enables calls for fire. Building as much of the capability around ATAK as possible promotes the axiom *train as you fight*.

As the SMART architecture matures, so will its model of the battlespace, which will ease expansion into new, but related military domains. Having a rich domain model will provide context that is necessary to accurately assess communication between team members.

Squad SMART Component Walkthrough

Squad SMART contains all of FO SMART plus additional functionality to incorporate an infantry squad requesting indirect fire support. The initial Squad SMART scenario assumes that the squad leader requesting indirect fire support uses ATAK for communication, the forward observer has access to both ATAK and a radio, and the fire direction center only has radio access (Figure 1). This configuration allows the system to assess both verbal communication and interaction with a tool that soldiers currently use.

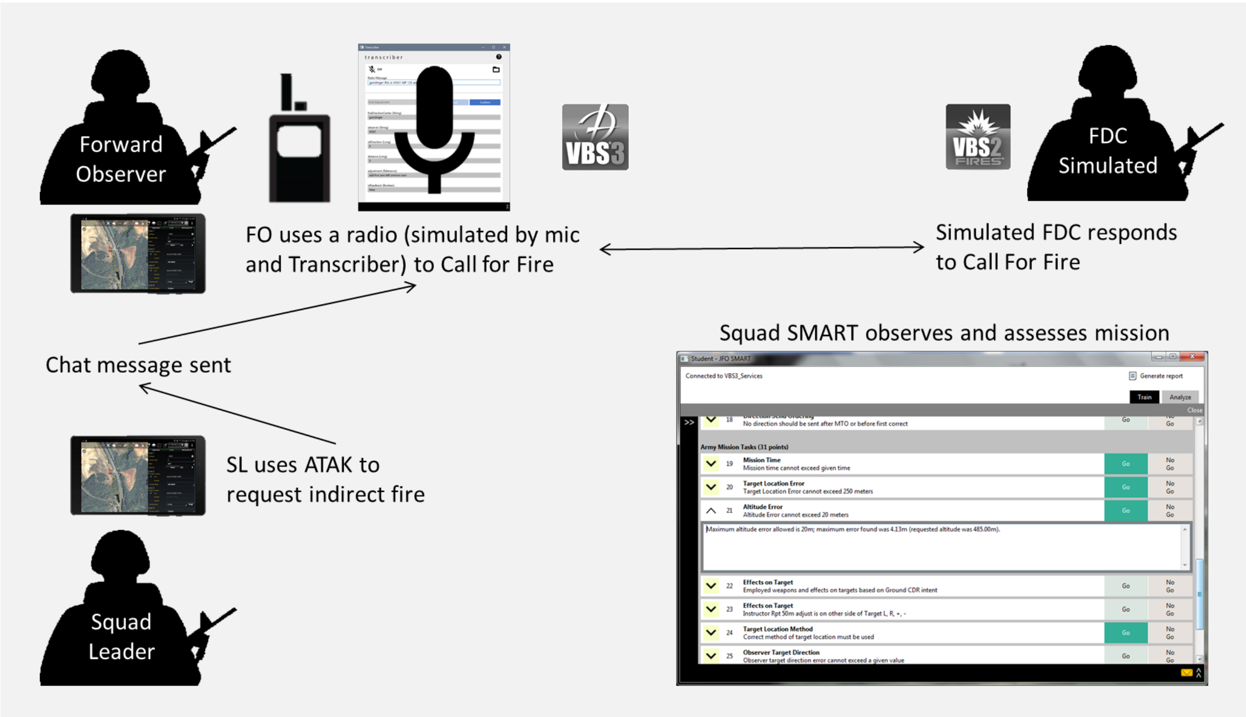


Figure 4: Squad Smart Overview

Squad Leader Requests Indirect Fire Support

A Squad Leader is on patrol and comes under fire from a bunker. After reacting to contact and providing initial guidance to his unit, the Squad Leader uses ATAK to request indirect fire. To streamline the reporting process, Veloxiti leveraged its GOTS Warfighter Associate ATAK plugin.

Report Small Arms Fire

The Squad Leader reports Small Arms Fire (SAF) either through ATAK chat or by using the ATAK point dropper (Figure 2). The Warfighter Associate detects the new event of interest, sends a notification, adds a map icon, and adds a C2 pointer (purple arrow). Future versions of Squad Smart may use the Warfighter Associate to deliver GIFT mediated intelligent training to a Squad Leader on the ATAK user interface.

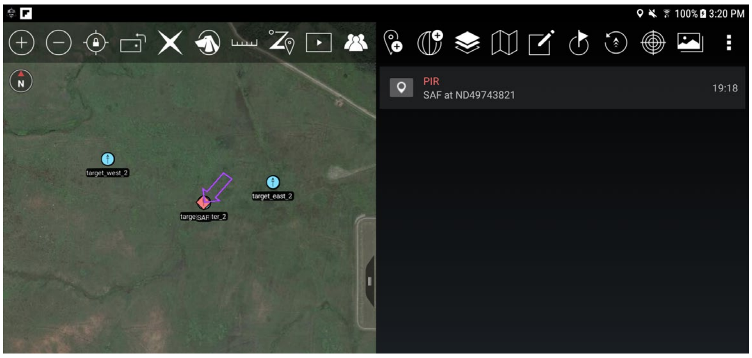


Figure 5: SAF Report

Submit Fire Request

The Squad Leader calls for fire using ATAK's standard indirect fire request mechanism. The request is sent through chat, observed using the Warfighter Associate chat monitoring service, and sent to Squad SMART, which parses the chat and assesses the Squad Leader's performance for accuracy and timeliness (Figure 3).

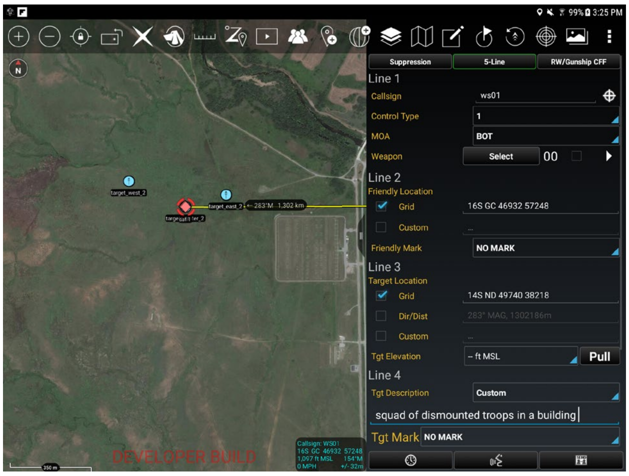


Figure 6: ATAK Fire Request (5 Line)

Forward Observer Calls for Fire

The Forward Observer receives the chat message from the Squad Leader on his ATAK device and contacts the Fire Direction Center using a radio. In Squad SMART, radio transmissions are simulated using a speech to text toolkit (for example, Dragon Naturally Speaking Home Edition) and the SMART Transcriber UI (Figure 4).

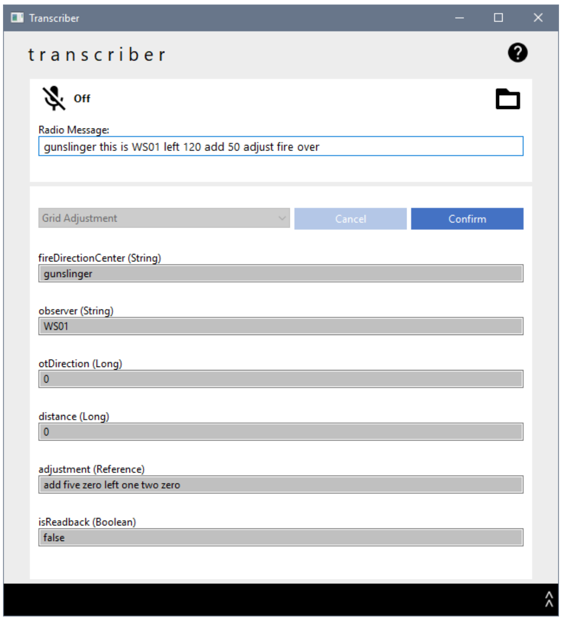


Figure 7: SMART Transcriber

A simulated Fire Direction Center processes the call for fire and the fire mission plays out in VBS3 (Figure 5). When using VBS 3.9.2, the Fire Direction Center is simulated by VBS2Fires. Squad SMART developed a simple Fire Direction Center simulator for use with simulations that do not provide an automated Fire Direction Center capability.

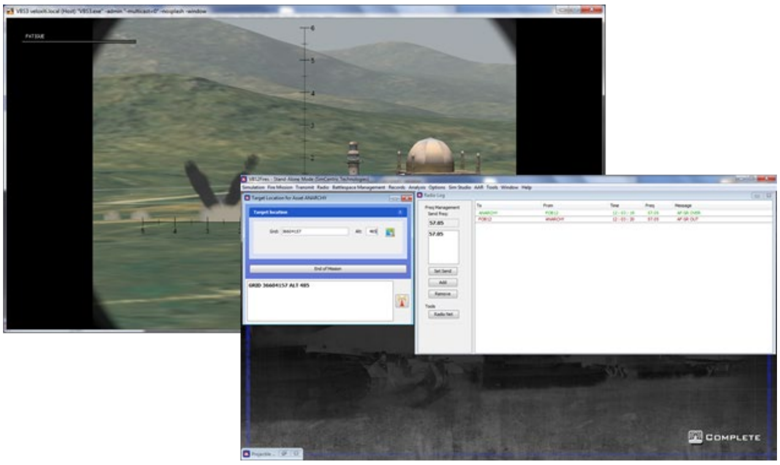


Figure 8: VBS3 and VBS2 Fires

Squad / FO SMART observes the fire mission, including natural language transmissions from the Transcriber and geographic location data from the simulation. The performance of the FO is assessed based on grading criteria provided by the Fires Center of Excellence at Fort Sill, Oklahoma. The real time grading user interface, shown in Figure 6, is optional.

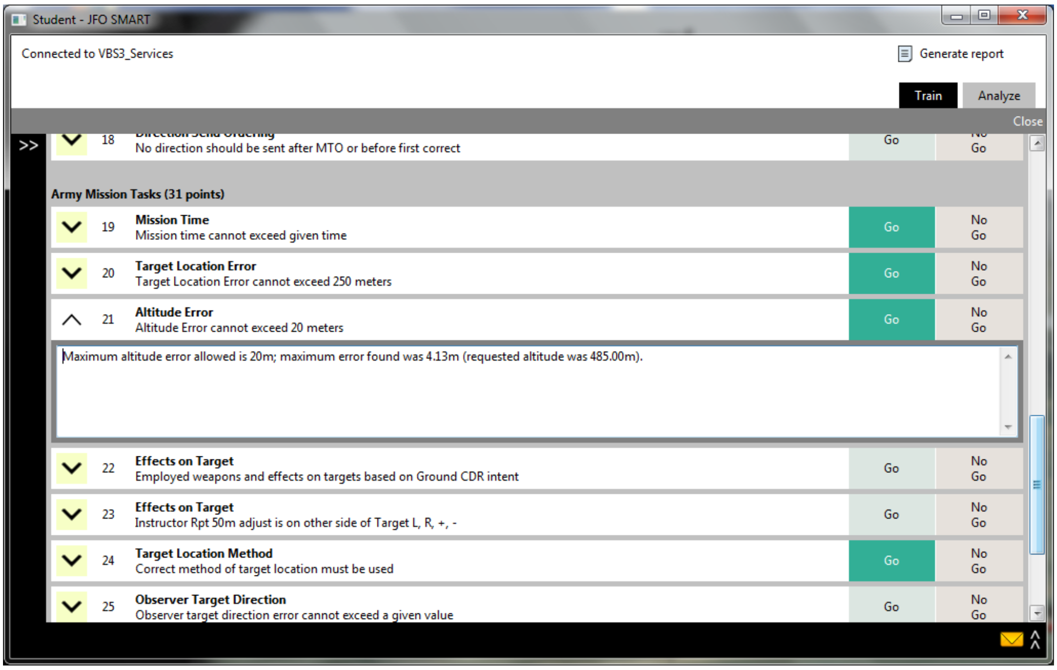
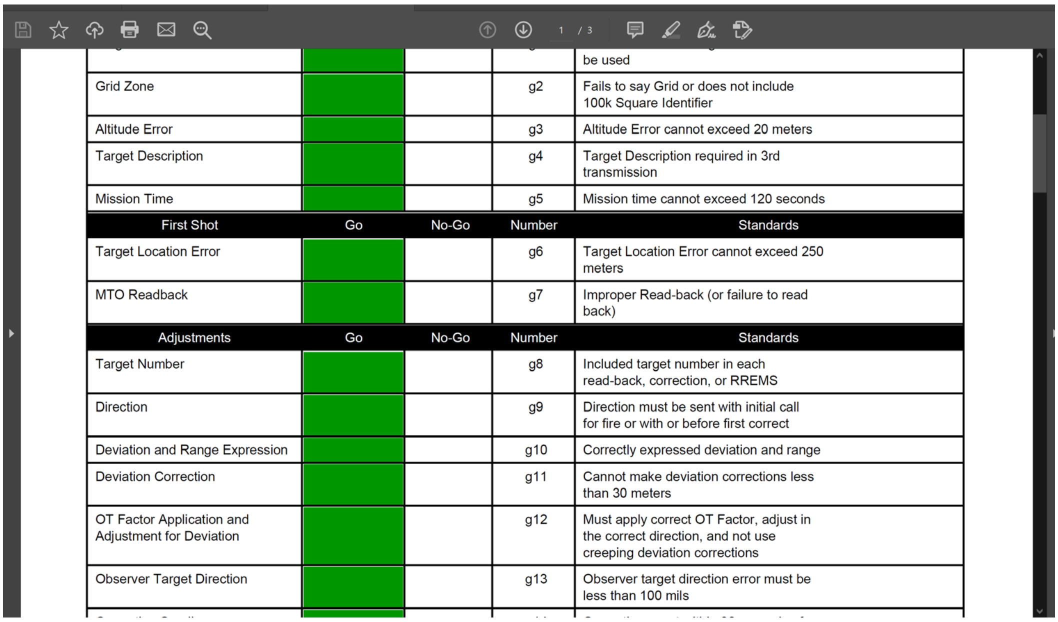


Figure 9: Squad SMART Real Time Grading

Performance assessments can be saved as PDF reports (Figure 7). This functionality is available in the Transcriber UI and the Real Time Grading UI.



	Go	No-Go	Number	Standards
				be used
Grid Zone			g2	Fails to say Grid or does not include 100k Square Identifier
Altitude Error			g3	Altitude Error cannot exceed 20 meters
Target Description			g4	Target Description required in 3rd transmission
Mission Time			g5	Mission time cannot exceed 120 seconds
First Shot	**Go**	**No-Go**	**Number**	**Standards**
Target Location Error			g6	Target Location Error cannot exceed 250 meters
MTO Readback			g7	Improper Read-back (or failure to read back)
Adjustments	**Go**	**No-Go**	**Number**	**Standards**
Target Number			g8	Included target number in each read-back, correction, or RREMS
Direction			g9	Direction must be sent with initial call for fire or with or before first correct
Deviation and Range Expression			g10	Correctly expressed deviation and range
Deviation Correction			g11	Cannot make deviation corrections less than 30 meters
OT Factor Application and Adjustment for Deviation			g12	Must apply correct OT Factor, adjust in the correct direction, and not use creeping deviation corrections
Observer Target Direction			g13	Observer target direction error must be less than 100 mils

Figure 10: PDF Report

THE VELOX FRAMEWORK AND MILITARY TRAINING

Overview

The Velox Framework was created to facilitate the development of OODA Loop systems. Velox enables Java developers to capture domain knowledge in two interacting graphs, an Observe-Orient graph, containing beliefs about the environment, and a Decide-Act graph, that models intent in the form of plans and goals. While Velox can be applied to a variety of system in a variety of domains, military decision aiding and training systems are a natural fit.

Observations about trainee performance can be captured as grounded beliefs in the Observe-Orient graph. These observations can be combined, processed, aggregated, and abstracted as necessary to create a domain model that informs the performance assessment of relevant tasks, driving the behavior of the plans and goals in the Decide-Act graph.

The Decide-Act graph is comprised of goals and plans. Goals model desirable states of the world and plans model techniques for achieving goals. Goals and plans at the top of a Decide-Act graph are typically highly abstract. These high-level nodes are decomposed into more specific goals and plans, eventually resulting in a plan that can be acted upon.

The Decide-Act graph of an adaptive training-based Velox agent that models the intent of a military trainer may use high level goals to capture trainee learning objectives. Lower level goals may capture secondary aims, such as keeping the training environment realistic or keeping the trainee engaged. Plans, which model means of achieving goals, may represent presentation forms, scenario modification strategies, etc.

Adaptive planning is fundamental to Velox. The Decide-Act graph closely monitors the Observe-Orient graph and updates plans and goals as necessary, enabling, for example, a system to request a specific scenario modification based on learner state (Figure 8).

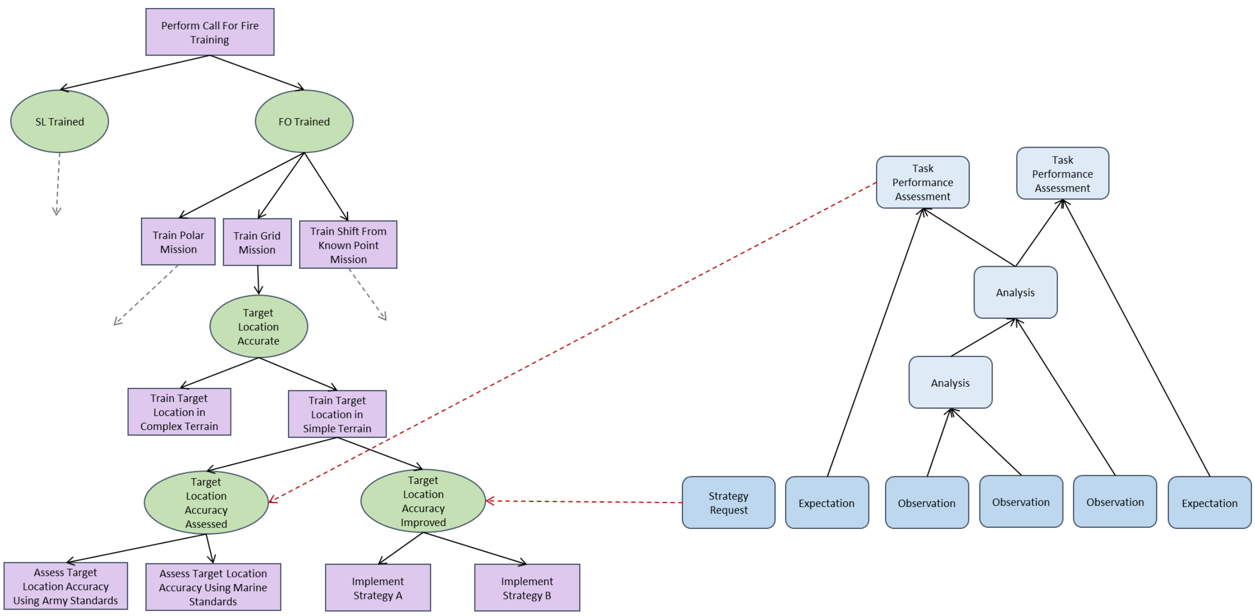


Figure 11: Notional Knowledge Base

Squad SMART and the Velox Framework

Squad SMART uses the Velox Framework for performance assessment. It combines observations about trainee performance with expectations in the Observe-Orient Graph (Figure 9). The current Decide-Act Graph contains goals and plans relating to basic skills to be assessed. This is appropriate because SMART has been focused on performance assessment. In the future, the Decide-Act graph will likely be enhanced to model more complex trainer intentions such as presentation mechanisms and scenario complexity. Outputs from GIFT will help enable more powerful Decide-Act functionality.

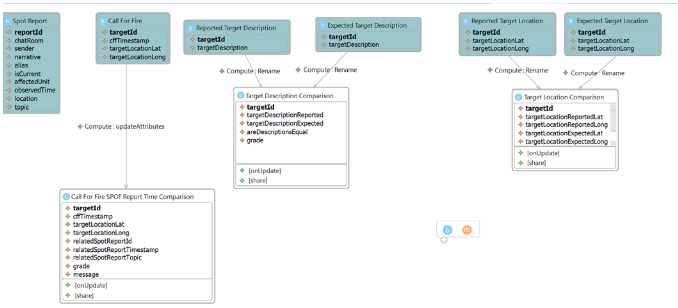


Figure 12: Simple Partial Squad Observe Orient Graph

SMART GIFT INTEGRATION

In addition to Squad SMART development, we are performing an initial integration between FO SMART and GIFT. Following an iterative development approach, we expect to update many aspects of this initial integration to meet the GIFT principles of reuse and standardization. FO SMART was a more appropriate initial use case for GIFT integration because it is more mature than Squad SMART and more dependent upon simulation data.

The current FO SMART integration with GIFT focuses on the GIFT Domain and Gateway modules. FO SMART uses the GIFT VBS Plugin and the DIS Interface Gateway module to interact with VBS3. Integration with DIS has been useful in making the SMART architecture simulation independent.

FO SMART has a generic condition class that drives assessment in the Domain module. Currently FO SMART grades tasks as Go or No-Go, which are mapped to GIFT assessment levels (Above Expectations, At Expectations, or Below Expectations).

The SMART architecture uses Veloxiti's GPR messaging framework, called Velox Messaging. Velox Messaging is a framework for developing Service Oriented Architectures that allows for code generation based on XML service and data contracts. Velox Messaging, which currently supports Java and C++, leverages Apache ActiveMQ and Google Protocol Buffers (ProtoBuf). Due to a Protocol Buffers version conflict with GIFT, FO SMART GIFT classes use standard ActiveMQ messaging to communicate with a proxy that that uses Velox Messaging to communicate with other SMART components.

FO SMART is primarily a performance assessment system. It performs real time task grading, including feedback as to why each task was marked Go or No-Go. A SMART task maps easily to a GIFT concept. Each task has defined data needs and an assessment procedure. Both architectures support hierarchical tasks. SMART is based on the Velox Framework, which uses hierarchical directed graphs. GIFT assessments contain tasks, which contain concepts, which contain conditions. The GIFT pedological model selects appropriate content based on learner state. While SMART has focused primarily on performance assessment, adapting the content that it provides a user based on output from GIFT is a natural use of a Velox Decide-Act graph.

SMART GIFT OODA LOOP COLLABORATION

Velox was developed to cover the entire OODA Loop and is very good at aggregating and abstracting data in a domain independent manner. GIFT is domain independent but focused on providing intelligent tutoring. GIFT and Velox are both relevant to each stage of the OODA loop, but perhaps should focus on different aspects. The following proposal is an attempt to leverage the strengths of each toolkit to develop an effective OODA loop based training capability (Figure 10).

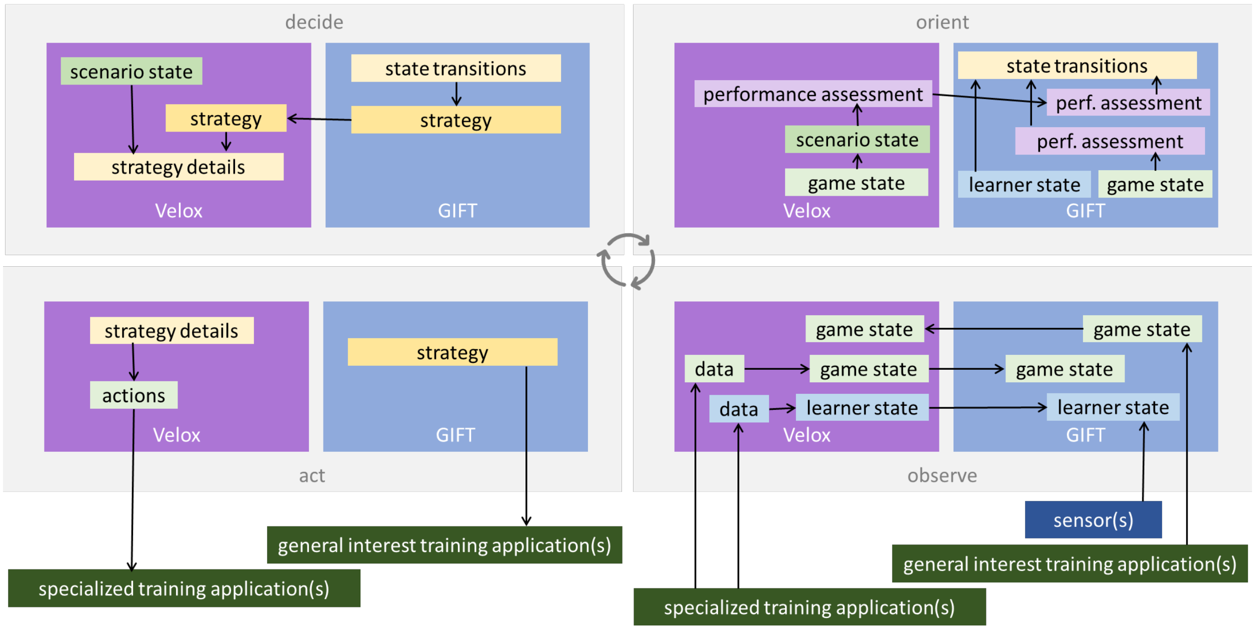


Figure 13: GIFT Velox OODA Loop Responsibility Thoughts

Observe: Both Velox and GIFT should contribute to the observe state. GIFT can leverage its sensors to observe learner state and its existing Gateway plugins to observe other relevant info. Small, modular Velox knowledge bases can be used to develop external sensors and send processed game state data to GIFT.

SMART Example: An external sensor module can be developed to assess a learner's stress level based on speech to text transcriptions and world state data.

Orient: The vital orient phase could also benefit from both Velox and GIFT. Velox is good at tracking progress through complex tasks for which performance assessment is context dependent. Unlike Velox, GIFT has user interface capabilities to enable tutoring creators to configure how performance is assessed and scored and has pedagogical expertise to inform instructional strategy selection.

SMART Example: Velox assesses observer target direction for correctness, timeliness (must be transmitted between certain fire mission steps), and necessity (only required for grid missions if adjustments are made). GIFT may consider below average performance on target direction reporting in context with learner affect to inform strategy selection.

Decide: GIFT has pedogeological expertise and should be responsible for high level instructional strategy choices. Velox is well suited for lower level decisions about how to implement instructional strategies based on the current scenario context and training application capabilities.

SMART Example: GIFT may recommend providing active tutoring to reduce target location error and Velox may request a specific ATAK map configuration.

Act: Both GIFT and Velox have roles to play in the act phase. Simple interactions can come directly from GIFT Gateway plugins. When necessary, Velox can be used to coordinate multistep processes or perform necessary calculations.

Smart Example: GIFT may request that a user is presented with an example polar call for fire. A Velox agent, which has been monitoring game state, can calculate the correct values, may send them to ATAK for display, and pan and zoom the ATAK map accordingly.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

Performing a basic, proof of concept integration between GIFT and FO SMART was a relatively straight forward software task. A more ambitious integration will benefit from updating the SMART architecture to better support GIFT. This includes better support for login-based startup, a more modular design (to support distributed re-use), and a clear division of responsibilities between SMART components and GIFT components. These changes would enable the development of a powerful OODA Loop based adaptive training system.

REFERENCES

Headquarters, Department of the Army (2016). FM 7-0 Train to Win in a Complex World. Washington, D.C., October 2016.

ABOUT THE AUTHORS

Stacey Sokoloff *is a Senior Knowledge Engineer with Veloxiti, Inc. Since joining the company in 2002, she has contributed to and provided technical leadership for numerous projects that attempt to advance the state of art by providing novel solutions to help warfighters train, maintain situational awareness, and perform their roles with increased expertise. Before joining Veloxiti, Stacey Sokoloff earned a Master of Human Computer Interaction (MHCI) from Carnegie Mellon University and a Bachelor of Science (BS) in Computer Science from Tulane University.*

John Merrihew *served as Principal Investigator on a dozen Artificial Intelligence projects in the US Army in the domains of data analytics, operations/intelligence fusion, UAV automatous operations and human terrain. John is a retired LTC Army Aviator who served 24 years in the US Army. He is a 4-time combat veteran of Operation Iraqi Freedom with 113 combat sorties and over 1300 combat hours over Iraq. He had command and staff positions at multiple levels in the US Army. He served in Infantry, Field Artillery, Aviation (Cavalry, Attack and Lift). John has extensive combat experience in intelligence fusion for warfighters at the tactical edge and is a qualified Joint Operational Planning.*

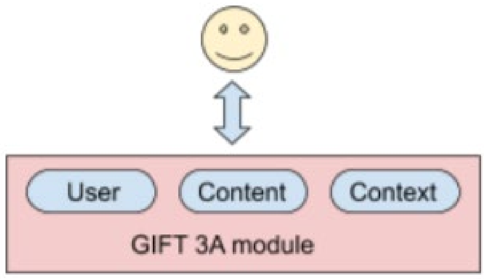
Towards a GIFT Enabled 3A Learning Environment

Faruk Ahmed[1], Keith Shubeck[1], Xiangen Hu[1,2]
The University of Memphis[1], Central China Normal University[2]

INTRODUCTION

Generalized Intelligent Framework for Tutoring (GIFT) (R. Sottilare et al., 2013; R. A. Sottilare et al., 2017) is a modular, domain-independent framework that is specially designed for authoring intelligent tutoring systems (ITS). In addition to other common features of ITS authoring tools, it has a sensory module, a special feature that most other ITSs do not have. The sensory module has a rich interface that can incorporate a variety of sensors (e.g., Emotiv sensor, SelfAssessment Sensor, Mouse). GIFT sensor interface opens up a gateway to analyze the cognitive state of learners. Proper analysis and utilization of sensor inputs will require an integrated framework that simultaneously uses data from the learner as a function of content and context. We propose to explore this potential of GIFT in this paper. We propose that the sensor module of GIFT can produce a 3A learning environment, namely, a learning environment that is user-aware, content-aware, and context-aware. Most existing ITS, including GIFT enabled ITS applications are already content-aware, and context-aware. We explored the "user-aware" component by utilizing the capabilities of modules of GIFT.

There are possible applications of the 3A environment for learners who have cognitive and/or behavioral disabilities (e.g., specific learning disability (SLD), intellectual disability (ID), emotional disturbance (ED), attention deficit hyperactivity (ADHD), autism spectrum disorders (ASD)) or sensory and/or physical disabilities (e.g., blindness, deafness, traumatic brain injury, cerebral palsy, muscular dystrophy) (Hallahan et al., 2020). We are in a process of producing a psychological foundation for the 3A learning environment. The first application would use the sensory input to capture facial expressions, the frequency of responses, and physical activities (e.g., speech, mouse activities, eye movements) to help learners that are visually impaired or have problems hearing. This work is based on the dissertation work of the first author, who had already piloted a mobile application that helps visually impaired individuals (Ahmed et al., 2018; Ahmed & Yeasin, 2017).



The proposed 3A learning environment prototype is narrowed down and designed using GIFT monitor and HTML5 media. Eventually it will be a GIFT extension that connects to the sensor module, a learning record store (LRS), and tutoring interface in real-time. We plan to use the implemented sensor module in the most recent release of GIFT prototype. The tutoring interface we are using is AutoTutor (Nye, 2013; Nye et al., 2014; Ventura et al., 2015).

This work concerns user-awareness in a 3A learning system (user-aware, content-aware, and context-aware). Several types of inputs are necessary to be aware of the learner such as sounds, facial expressions, and activities. We started with extracting emotions from the facial expressions as an initial step.

Analyzing facial expressions and emotions provides insight into the cognitive state of the learners. For example, meta-analyses on emotions in learning suggest that common emotions during learning include boredom, confusion, frustration, happiness, anxiety, and flow or engagement (Sidney D'Mello, 2013; Loderer et al., 2018). We can assess a learner's reaction to positive and negative feedback by detecting happy and sad expressions (Plass et al., 2019).

A "calm" expression might suggest that a learner is in a state of flow, which is positively correlated with learning (Csikszentmíhalyi, 1991). Additionally, anger and frustration are important to detect because these can lead to disengagement, which is negatively correlated with learning.

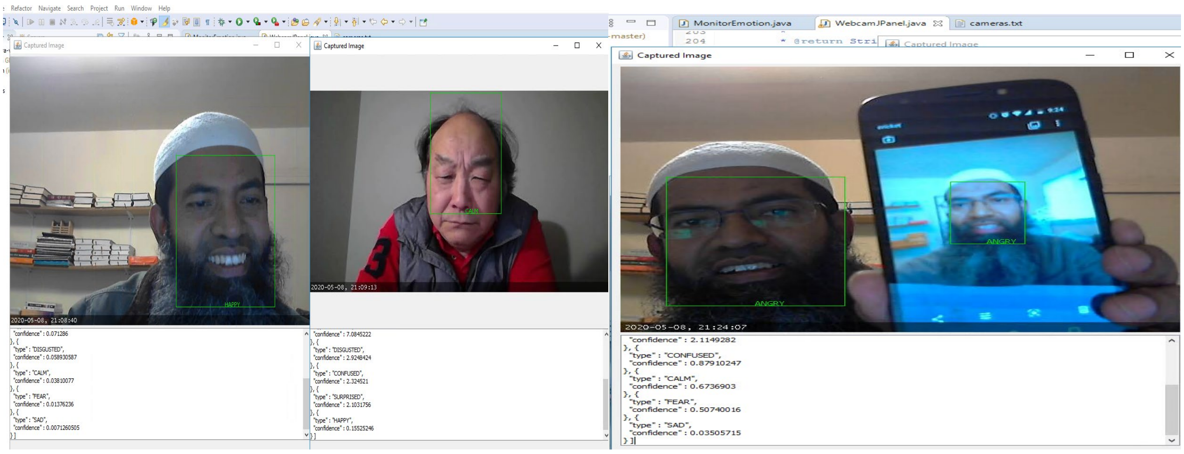
Monitoring emotions distributed over time provides an aggregated cognitive state of the learner. We can track distributed emotions that map to distributed cognitive states. Analysing those cognitive states would give deeper insight about the learning content. It may contribute to improving the design of the content.

THE PROTOTYPE

We have built two prototype implementations of finding discrete cognitive states. One is through GIFT's rich monitor module and another is through HTML5 media capability of modern web browsers. In both cases we used AWS Recognition for emotion recognition ('Amazon Rekognition - Developer Guide', no date).

GIFT Monitor

We added a new java class which receives an image and returns emotions with a confidence score. This class also returns the bounding boxes of the faces which help tracking faces. In the prototype we used a button to capture an image and send for emotion analysis. Once the emotions, confidence scores, and bounding box are available we let the canvas draw this information on the image. This scenario is for development but in production there will be no button to capture images. The process will run in the background and the emotions will be saved corresponding to any content. In the monitor module the GIFT looks for webcams in "cameras.txt" file and the IP address of the camera should be reachable by GIFT instance. Multiple cameras can be listed in that file and our prototype class is able to capture emotions from those cameras. In the learner's machine the "Yawcam" has to be running and streaming must be enabled. If the learner's computer is inside a private network the port 8081 should forward to that machine. This requires router port forwarding configuration. GIFT can find a learner's webcam with those few steps of configuration (*GIFT Operator Station Instructions 2020-1 - GIFT - GIFT Portal*, no date; D'Mello, Picard and Graesser, 2007)



(a) (b)

Figure 2. Emotion recognition in GIFT monitor module

Figure 2 shows an example picture of the emotion recognition in the GIFT monitor module. The picture (a) is for multiple cameras with emotions and the picture (b) is for a single camera with multiple emotions. In a classroom environment if we want to observe the emotions of learners that is also possible because the system is able to identify multiple faces in a single image.

HTML5 media capability

Another way to Capture Emotion in a learner's computer through html5 media capability. Most of the modern browsers support this except safari and internet explorer (MDN contributors, 2019). We programmed an html page with javascript to experiment. At the time of page load the page asks for permission to access media and it opens webcam if available. Once webcam video feed is available the javascript running in the background captures images and analyzes emotions. These emotions can be easily stored in LRS through xAPI calls and GIFT can retrieve those on demand. We can build a filter in GIFT to query those emotions. In the prototype we did not implement xAPI call yet. Figure 3 shows an example of an HTML version of emotion recognition.

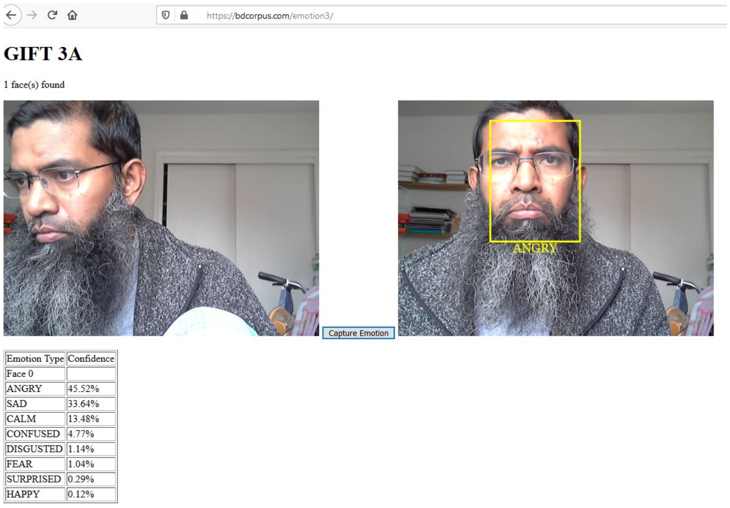


Figure 3. Javascript prototype of emotion recognition

Working principle of GIFT monitor and HTML5 media is depicted in figure 4. The GIFT monitor looks for a webcam in a public or reachable IP address which is preconfigured (e.g., the IP address is stored in the cameras.txt file). Whereas the HTML5 media feeds the information to LRS and GIFT filter fetches those when required.

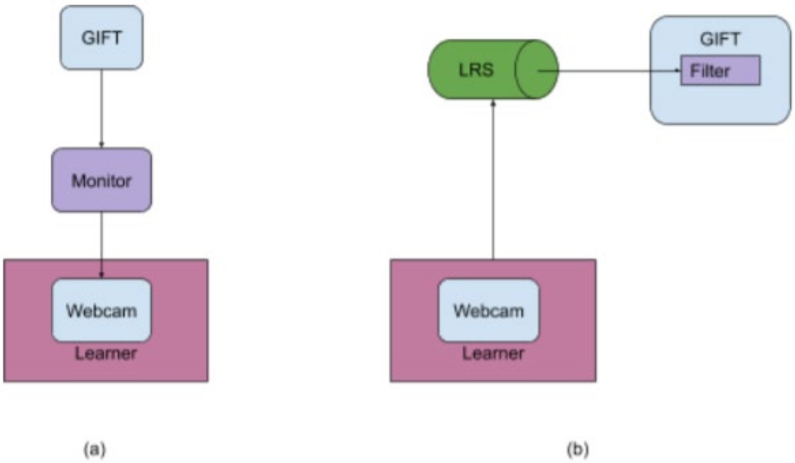


Figure 4. Working principle of (a) GIFT monitor and (b) HTML5 media

PROS AND CONS

We observed pros and cons of GIFT monitor and HTML5 media for capturing emotions. The GIFT monitor is centralized but requires configurations. This method will necessarily increase the server computing load if large amounts of image data from multiple learners are processed centrally. Moreover the webcam IP address should be reachable by GIFT and there should be a Yawcam application running. In addition, there are also issues of privacy when learner's facial images are captured and sent to the server in real time. The HTML media does not require any configuration and it is distributed. Moreover the computations happen in the learner's computer. The URL of the emotion recognizer can be embedded while authoring a course (e.g., while using "Course Creator").

EXAMPLE ANALYSIS

For simplicity, we sorted the eight emotions in alphabetical order shown in table 1. Using this order we generated artificial data that contains virtual content and imaginary emotions. The plot of content vs. emotions is shown in figure 5. By looking at the graph we can say that the learner was "surprised" , most likely while studying content 12, whereas he was "calm" while studying content 5 and so forth.

Table 1. Alphabetic order of the emotions

ANGRY	CALM	CONFUSED	DISGUSTED	FEAR	HAPPY	SAD	SURPRISED
1	2	3	4	5	6	7	8

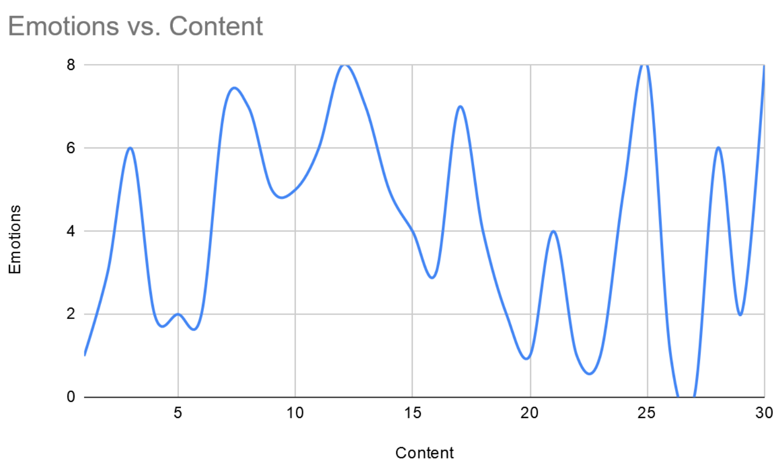


Figure 5. An example plot of artificially generated emotion data with highest confidence score

In LRS we can store the emotions along with . In an instance of emotion recognition there are eight emotions with a confidence score. The confidence score is the percentage. The moving mean and the moving standard deviation can be calculated using different algorithms (Joni, 2013). If we store the instantaneous values of emotions along with the moving mean and moving standard deviation we will have the emotion profile of a learner. Figure shows a real time plot of emotion with confidence score in a Chrome browser.

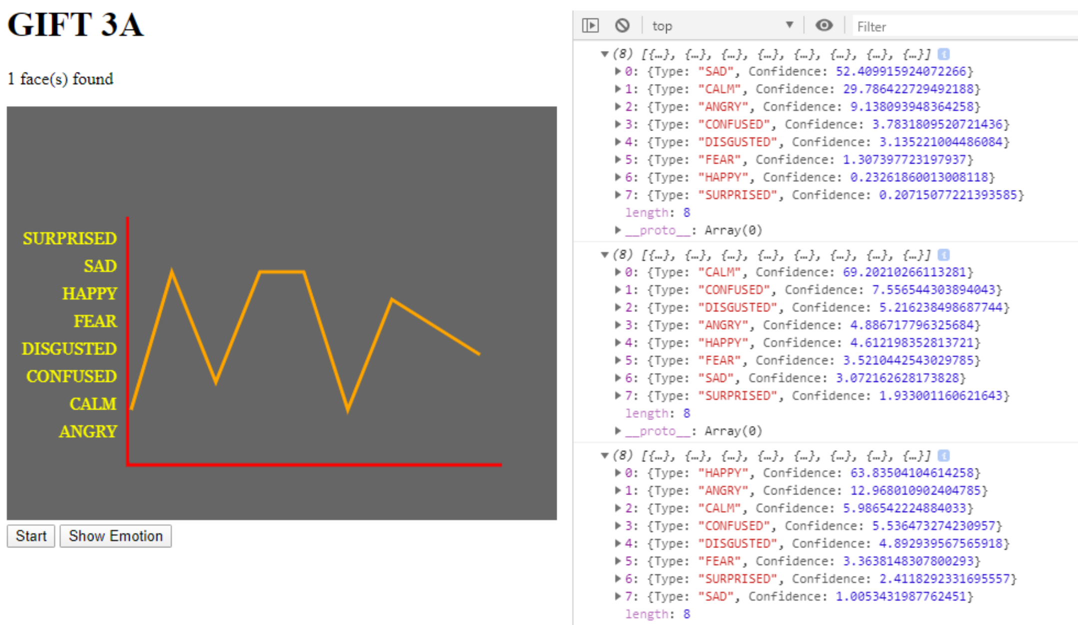


Figure 6. A plot of highest confidence score of emotion along with all confidence score.

CONCLUSION

The potential of enabling GIFT 3A is huge. GIFT can not only suggest a break or crack a joke to a bored learner but it can also provide statistics of a user's stance about the learning content (e.g., hypothetically 85% learner happy on taking “Test Course”). Previous research suggests that analyzing facial expressions works best in a multimodal system (D’Mello and Graesser, 2010) . GIFT provides easy access to other relevant input modes that can be used to improve our confidence in our assessment of a learner’s affective state. Future direction of 3A is to incorporate “sound”. In addition to incorporating “sound” we would like to experiment with asking learners to do something and monitor that activity. For example, GIFT will ask learners to grab an object and monitor that grabbing activity. From that activity it is possible to infer a few types of exceptions a learner may have (e.g., visual impairment or hearing impairment).

REFERENCES

Ahmed, F. et al. (2018) ‘Image Captioning for Ambient Awareness on a Sidewalk’, in 2018 1st International Conference on Data Intelligence and Security (ICDIS). ieeexplore.ieee.org, pp. 85–91.

Ahmed, F. and Yeasin, M. (2017) ‘Optimization and evaluation of deep architectures for ambient awareness on a sidewalk’, in 2017 International Joint Conference on Neural Networks (IJCNN). ieeexplore.ieee.org, pp. 2692–2697.

‘Amazon Rekognition - Developer Guide’ (no date). Available at: https://docs.aws.amazon.com/rekognition/latest/dg/rekognition-dg.pdf.

Csikszentmíhalyi, M. (1991) ‘Flow: The psychology of optimal experience In: New York: HarperPerennial’.

D’Mello, S. (2013) ‘A selective meta-analysis on the relative incidence of discrete affective states during learning with technology’, Journal of educational psychology. American Psychological Association, 105(4), p. 1082.

D’Mello, S. K. and Graesser, A. (2010) ‘Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features’, User modeling and user-adapted interaction, 20(2), pp. 147–187.

D’Mello, S., Picard, R. W. and Graesser, A. (2007) ‘Toward an Affect-Sensitive AutoTutor’, IEEE intelligent systems. ieeexplore.ieee.org, 22(4), pp. 53–61.

GIFT Operator Station Instructions 2020-1 - GIFT - GIFT Portal (no date). Available at: https://www.gifttutoring.org/projects/gift/wiki/GIFT_Operator_Station_Instructions_2020-1 (Accessed: 12 May 2020).

Hallahan, D. P. et al. (2020) ‘Exceptional learners’, in Oxford Research Encyclopedia of Education. oxfordre.com.

Loderer, K., Pekrun, R. and Lester, J. C. (2018) ‘Beyond cold technology: A systematic review and meta-analysis on emotions in technology-based learning environments’, Learning and Instruction, p. 101162.

MDN contributors (2019) Navigator.getUserMedia(), MDN Web Docs. Available at: https://developer.mozilla.org/en-US/docs/Web/API/Navigator/getUserMedia (Accessed: 12 May 2020).

Nye, B. D. (2013) ‘Integrating GIFT and AutoTutor with sharable knowledge objects (SKO)’, in AIED 2013 Workshop on GIFT. CEUR. gifttutoring.org, pp. 54–61.

Nye, B. D. et al. (2014) ‘Autotutor in the cloud: a service-oriented paradigm for an interoperable natural-language its’, Journal of Advanced Distributed Learning Technology. sites.google.com, 2(6), pp. 35–48.

Plass, J. L. et al. (2019) ‘Emotional design for digital games for learning: The effect of expression, color, shape, and dimensionality on the affective quality of game characters’, Learning and Instruction, p. 101194. doi: 10.1016/j.learninstruc.2019.01.005.

Sottilare, R. et al. (2013) Design Recommendations for Intelligent Tutoring Systems: Volume 1: Learner Modeling. Army Research Laboratory.

Sottilare, R. A. et al. (2017) ‘An updated concept for a Generalized Intelligent Framework for Tutoring (GIFT)’, GIFTtutoring. org. Available at: https://www.gifttutoring.org/attachments/download/2076/Updated%20Concept%20for%20the%20Generalized%20Intelligent%20Framework%20for%20Tutoring_9%20May%202017.pdf.

Ventura, M. et al. (2015) 'NewtonianTalk: Integration of Physics Playground and AutoTutor using GIFT', in AIED Workshops. pdfs.semanticscholar.org. Available at: https://pdfs.semanticscholar.org/a796/58178f4541d5bad17fddf3f734b69a9776a1.pdf.

ABOUT THE AUTHORS

Dr. Faruk Ahmed *is instructor in the Department of Electrical and Computer Engineering at The University of Memphis (UofM). Dr. Ahmed received his MS in electrical engineering from UofM and Ph.D. in engineering from UofM. His primary research areas include Assistive Technology Development, Human Computer Interaction (HCI), Machine Learning, Computer Vision, and Adaptive Instructional Systems (AIS).*

Keith Shubeck *is doctoral candidate in cognitive psychology at the University of Memphis. He received a Certificate of Cognitive Science and a M.S. in psychology in December 2015. He works with his major professor, Dr. Xiangen Hu, in the Advanced Distributed Learning Partnership Lab at the Institute for Intelligent Systems. His dissertation work compares interactive and vicarious tutoring frameworks in a conversation-based intelligent tutoring system for critical thinking and scientific reasoning.*

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.*

THEME IV: COMPETENCY-DRIVEN TECHNIQUES

Using GIFT, Competencies, and Virtual Reality Training to Provide Effectiveness Improvements for Nuclear Technicians

Mike Kalaf, Christopher Meyer, Lucy Woodman
Synaptic Sparks, Inc.

INTRODUCTION

Brawner and Sottilare define AISs as “computer-based systems that guide learning experiences by tailoring instruction and recommendations based on the goals, needs, and preferences of each learner in the context of domain learning objectives”, (Brawner & Sottilare, 2018, p. 55).

An ITS is an important type of AIS in such that the role of the ITS is to deliver and manage adaptive instruction. It is made up of four important parts: a learner model, an instructional model, a domain model, and an interface model (Brawner & Sottilare, 2018). GIFT combines the abilities of the AIS and the ITS to provide a framework that students and tutors can use to create and take courses.

As the first of three major parts of this demonstration - GIFT was implemented as the AIS/ITS and can be integrated with external applications through its services. Among these services are a standard approach for interfacing training applications, domain knowledge representation, performance assessment course flow, a pedagogical model, learner modeling, survey support, a learning management system, and a standardized approach for integrating sensors (Hoffman & Ragusa, 2015). This made GIFT an ideal choice for integration with the nuclear reactor training software to set up future experiments to improve the nuclear reactor training software’s effectiveness.

It was also important to this demonstration that the learner’s progress and abilities be quantifiable and that goals were clear. To accomplish this, the authors integrated CaSS into the GIFT which became the second major part of this demonstration.

CaSS is currently being integrated into GIFT at the time of this writing, and according to Gordan et al., “...the Competency and Skills System (CaSS) is a Total Learning Architecture (TLA) component that generates rich and traceable data about learning experiences and how they relate to skill proficiency, ultimately resulting in a certified set of credentials...” Thus, this effort contributed to the Use Case testing of the CaSS integration with GIFT through offering a chance to test verified competencies with learner performance and adaptive tutoring.

Finally, the nuclear reactor training software was made in response to part of a government multi-agency request to address challenges in U.S. Department of Energy (DoE) training methodologies. In order to help remedy this problem, The DoE has called for the modernization of nuclear training over recent years. So, to bring together the entire demonstration, the authors helped to create a nuclear training course in Virtual Reality (VR) utilizing GIFT, CaSS, and the Unity game engine.

Due to the unique nature of logistical challenges facing the world at the time of this writing, a live demonstration will be given to GIFT stakeholders at a later calendar date. And, the open source software artifacts used to integrate the disparate system together will be provided as part of a distributable package once physical co-location of coworkers and office equipment is again safe.

PARTS LIST AND DESCRIPTIONS

This section describes the parts and components that were used in the making of the prototype system described in this paper. Interested community members are encouraged to request further information or specifications from any of the authors if desired.

The authors broke down the nuclear technician's skills (originally created by nuclear station Subject Matter Experts, or SMEs) into their respective competencies. This gave the authors the materials necessary to create a competency framework on a public CaSS server. This server is currently online and available here: http://cassdev.gifttutoringclouddev.org. It is available for public use but should be considered volatile and for development purposes only - meaning user accounts and frameworks may be cleared at any time without notice.

The referenced framework was loaded into the CaSS database, and GIFT then was hard-coded to communicate with the CaSS server via REST calls to retrieve all competencies. This competency framework then became the Course Concept basis for gauging learner performance during the nuclear training lesson.

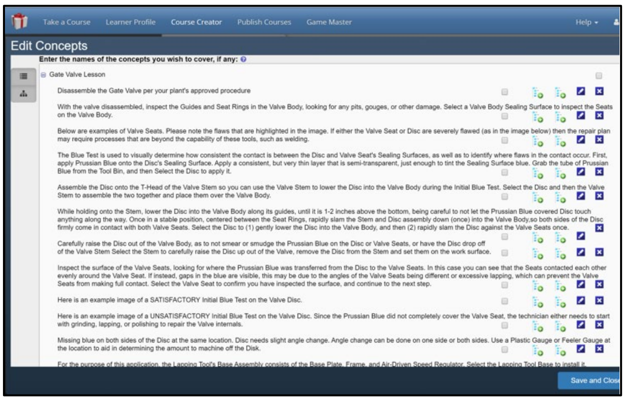


Figure 1: Screenshot of CaSS Nuclear Technician Competency Framework as GIFT Course Concept List

Each competency then became a performance measure in the lesson that could be assessed and then repeated as many times as necessary as a lesson step in the nuclear reactor training software. The GIFT course could then, through training application middleware, monitor the learner's progress and make sure that they were progressing optimally. These steps, through "Performing Below / At / Above" measures programmed into them, were finally able to provide GIFT, CaSS, and the nuclear reactor training software new training capabilities. Namely, the ability for GIFT to determine if lesson steps should be repeated or skipped based on the learner's acquisition of CaSS competencies as gauged by lesson step time duration and correctness only. The authors hope to be able to continue to add additional performance measures at a future time for additional experimentation.

For a guide on how to configure GIFT to operate with external applications in similar ways as to this particular demonstration, the reader is encouraged to read the documentation available online here as a starting point: https://gifttutoring.org/projects/gift/wiki/Developer_Guide_2020-1#Simple-Example-Training-Application.

GIFT Software Suite

At the time of this writing, GIFT 2019-1 has been released at www.gifttutoring.org/projects/gift/files. The reader may install this version of GIFT by creating a free GIFT account if they have not done so already. Once downloaded, the reader will be able to install their own local GIFT server and configure it to their liking. Readers can find configuration instructions included with their download as well as detailed discussions on the Forum tab at the www.gifttutoring.org homepage after making their own account.

CASS Online Test Database

A server maintained by the Advanced Distributed Learning (ADL) CaSS team can be found at https://cassproject.github.io/casseditor/. By following the https://cassproject.github.io/casseditor/ link, readers may explore the site to edit and configure their own competency and mastery framework. This server is maintained by the ADL CASS team, but any user has full permissions to create and edit their own framework. Readers may search for 'GIFTSym8' to examine the framework created for this paper's effort. Readers may also register with the CASS project at https://www.cassproject.org, download the open source code from the referenced GitHub project, and build/configure/maintain their own CASS server.

Note: Some modifications to the CaSS server and APIs were necessary in order for this paper's full demonstrable capability. These modifications will be provided back to the GIFT and CaSS communities as they reach maturity.

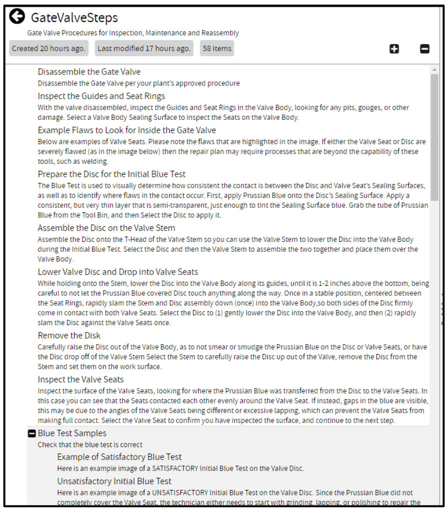


Figure 2: Screenshot of CASS Framework for GIFTSym8 in CASS Editor Web Tool

Unity

As GIFT continues to have cases of successfully integrating with Unity applications, the authors can add the additional software suite for nuclear reactor technician training to that list. As the nuclear reactor training software operates in both Desktop and Virtual Reality modes, GIFT was proven to operate in both domains as an ITS. A non-commercial version of Unity may be downloaded here https://unity3d.com/ if the reader would like to experiment building their own external applications.

Hardware

In order to further demonstrate that local GIFT servers can operate in conjunction with modern VR devices while performing adaptive training, a variety of hardware was chosen with cost and existing available equipment being the main driver in selection. Not included are the PCs and peripherals necessary to run GIFT locally, and powerful enough to run VR-capable applications.

- VivePro VR System: https://www.vive.com/eu/product/vive-pro/
- Nuclear Technician Gate, Globe, and Misc. Valve Training Software Suite Equipment
- Additional HTC VR System Sensors and Accessories
- 2 Servers and Custom Configurations (AWS Images Available to Community Upon Maturity)

The hardware listed above resulted in a combined system that was capable of monitoring a learner's progress through a nuclear reactor training scenario being monitored and managed by GIFT with reference to CaSS competencies.

METHODOLOGIES

This section further explains the configurations, interfaces between disparate systems, and the experimental framework that was created as a result of this paper's effort.

GIFT Installations, Configurations, and Modifications

It is assumed that the reader has a working knowledge of the GIFT software suite in order to fully understand this section. Special attention is given to the new areas of study that this paper explores, but only brief mention is given to basic GIFT topics. For more information, the reader may refer to GIFT documentation at www.gifttutoring.org, or How-To YouTube videos at https://www.youtube.com/channel/UCWtI_V8f2mN5XD6h2lCjsAA.

Prerequisites to this section include having downloaded GIFT 2019-1, fully having configured GIFT server, and having configured network communications and hosting to the point of understanding the reader's system being localhost vs. hosted online at a specific IP address or DNS web address. For this demonstration, the authors used a localhost GIFT server system, an online CaSS server, an online message passing server (ActiveMQ) acting together with translation middleware on the same virtual machine, and finally another physical server system hosting the nuclear reactor training Desktop and VR programs.

For this demonstration, it is important to note that GIFT communicated via RESTful calls to the CaSS server for competency information, and also passed messages back and forth through the middleware running on the additional ActiveMQ server to and from the nuclear reactor training software server.

CASS Server Installations and Database Entries

At the time of this writing, the authors have successfully installed a CaSS server on an Amazon Web Service (AWS) Elastic Compute Cloud (EC2) instance. The server can be accessed here http://cassdev.gifttutoringclouddev.org/. The reader must create a free user account in order to access the CaSS frameworks.

Virtual World Scenario in Unity

The nuclear reactor training environment was created using a combination of purchased assets from the Unity store and assets created by the authors. Among the assets created by the authors was a 3D replica of a gate valve, built to exact reactor specifications. One of the main objectives of the nuclear training course was to learn how to properly assemble and disassemble this particular gate valve. Therefore, the 3D model (displayed below in Figure 3) was created to be as detailed as possible so that the learner could practice assembling and disassembling as if they were working with the real one. The model was created using photos and measurements of the real valve and modeled using Autodesk Fusion 360 and Unity.

Figure 3: Screenshot of a Deconstructed Gate Valve in the Nuclear Reactor Training Software

The Complete User Story

Under optimal operational situations, the full demonstration is performed in a single room large enough for "room scale" virtual reality, usually no smaller than 10 ft. x 10 ft. square. One "instructor" launched the nuclear reactor training virtual reality software, while the learner equipped the virtual reality headset, controllers, and wireless monitoring devices. The instructor then begins the GIFT course, middleware server programs, and then allows the learner to proceed at their own pace.

The learner then began "Practice" mode in the training software, which is a step-by-step progression of tasks to disassemble and reassemble nuclear reactor valves modeled to the millimeter level of accuracy. An example of an instruction list is shown in Figure 4 below.

Figure 4: Screenshot of a Step-by-Step List of Tasks in Nuclear Reactor Training Software

The learner progressed through these steps, but if they progressed quicker than 5 seconds per step GIFT would register the series of steps as "Above Expectation" and proceed to skip sections of content. If the learner took longer than a certain duration (variable per lesson step), GIFT would register a "Below Expectation" and proceed to repeat a section of lesson content until more-acceptable lesson step durations were achieved.

While the original version of the training software is now deployed in the field, the authors look forward to future opportunities to determine if DoE representatives are as excited about these initial results from GIFT / CaSS creating new adaptive instruction capability as the authors have been to potentially improve the state of nuclear technician training even further.

CONCLUSIONS AND FUTURE RESEARCH

The combination of GIFT, CaSS, and Unity allowed for a powerful learner experience. Leveraging GIFT's capabilities as an ITS as well as its ability to integrate with various applications opens the door to countless possibilities for learning.

The authors' nuclear training application enabled nuclear technicians to train with the very same kind of valves that they would use in their everyday job. Increased opportunities for training in the virtual world should sharpen skills and will hopefully reduce accidents. Further, since GIFT allows learners to be almost completely autonomous and yet benefiting from adaptive instruction, the need for a human instructor can be reduced and therefore weaken the barrier of entry to training in the profession.

REFERENCES

Brawner, K.., Sottilare, R. (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 Intelligent Tutoring Systems (ITS) Conference, Montreal, Quebec, Canada.

Hoffman M, Ragusa C. Unwrapping GIFT: a primer on authoring tools for the generalized intelligent framework for tutoring. Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym2); 2015 Jun 12–13; Pittsburgh, PA. Orlando (FL): Army Research Laboratory (US); c2015. p. 11–24.

Gordon, J., Hernandez, M., & Smith, B. Competency-Based Learning (2018). Advanced Distributed Learning Initiative. https://adlnet.gov/resources/publications/2019/08/Competency-Based-Learning/

Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook Handbook, Nuclear Technicians, on the Internet at https://www.bls.gov/ooh/life-physical-and-social-science/nuclear-technicians.htm (visited May 24, 2020)

ABOUT THE AUTHORS

Mike Kalaf *has over 30 years of Modeling, Simulation and Training leading large scale efforts leveraging cutting edge technology. Mike has worked in the commercial and military aviation, training and simulation business. Mike has led several programs integrating "state of the art" technology and delivering highly successful technology and business innovation. Mike's formal education includes an earned Mechanical Engineering degree from Rochester Institute of Technology, RIT.*

Christopher Meyer *received his Bachelor and Master of Science degrees in Computer Science from Kansas State University, also receiving minors in Economics and Modern Languages. Chris also studied abroad for a year during a tour in Japan at Chukyo University dedicated to the specialized study of Artificial Intelligence. After completing traditional education phases, Chris was employed at Lockheed Martin for 10 years working 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.*

Lucy Woodman *graduated from Seminole State College with a Bachelor of Science in Information Technology. Lucy has supported Synaptic Sparks for two years in addition to a successful internship and transitioned to a be a major supporter of big data services within SSI in December of 2018. Lucy is a certified Amazon Web Service specialist in addition to being a certified System Architect.*

Matching Content to Competencies with Machine-Learning: A Service-Oriented Content Alignment Tool for Authoring in GIFT and Beyond

Benjamin Bell[1], Keith Brawner[2], Debbie Brown[1], Elaine Kelsey[1]
Eduworks[1], US Army CCDC STTC[2]

INTRODUCTION

Content Management as a Touchstone for Scalable Training Development

Authors of intelligent tutoring systems (ITS) face a common challenge of finding the instructional content best-suited to train the skills and competencies for which the ITS is intended. In modern times, an almost limitless array of web-accessible content can provide relevant instructional material to the author who has the time to search and curate online resources. A fortunate author has content which is organized, indexed, and discoverable; but most authors are not fortunate. Even a fortunate author does not have content which is indexed by learning objective or competency area, even if that were possible given the quickly changing nature of content and objectives.

A better way of indexing content on behalf of authors would be to arrange for ready alignment with the competencies around which the author is building a training package. The associations between content and competencies would be dynamic, and would be permanent attributes of neither the content nor the competencies. Further, it is possible to generate this dynamic association automatically.

This paper presents a solution to address the need for supporting training development, by automating the process of aligning information with competencies to be trained. At GIFTSym 7, we introduced the basic objective and technical approach of this effort, called *Machine-Assisted Generation of Instructional Content* (MAGIC). Since that time, we have refined the application of Machine Learning techniques to improve alignment results, and emphasized a services-oriented implementation to make MAGIC a service accessible to GIFT (Sotillare, *et al.*, 2013) as well as to any other training framework via an Application Programming Interface (API). In this paper we also highlight the interoperability of MAGIC with CaSS, an open-source competency framework management system (Robson & Poltrack, 2017) to facilitate how authors present MAGIC with the learning objectives for which they are seeking relevant content. We conclude with observations on how MAGIC can streamline and interoperate both with GIFT and with elements of major initiatives such as the Army's Synthetic Training Environment (STE).

The Need for Tools to Manage Instructional Content

Scaling virtual training for teams to fully address Army needs requires tools and techniques for efficiently creating team tutoring simulations. Locating and tagging content that aligns with desired competencies remains a labor-intensive process. Authors of team training must navigate complicated content management tasks related to distinguishing content that supports individual skills and content aligned with team skills. To help developers of team training find and tag relevant content more efficiently, automation is needed that supports analysis of content and its alignment with team and individual learning objectives.

In this paper we present progress on an automation tool first introduced at GIFTSym 7 that helps training developers find, organize, and curate resources aligned with desired individual and team competencies. ***Machine-Assisted Generation of Instructional Content*** **(MAGIC)** analyzes source documents and extracts content that aligns with competencies specified by the training developer. MAGIC additionally supports development of team training by performing this alignment for both individual and team competencies. Building on and extending existing artificial intelligence (AI) and natural language processing (NLP) techniques, MAGIC streamlines content alignment, distinguishing between individual and team content, and helps extend the reach of GIFT tutoring to meet Army team training demands (Gilbert et al., 2017; McCormack et al., 2018; Sinatra, 2018).

Scaling Synthetic Training with Intelligent Content Management

With growing demand for synthetic team training, GIFT tutors must be able to scale learning to meet training needs across broad content areas and offer instructional value for both individual Soldiers and teams (Sottilare et al, 2011; Sottilare et al, 2018; Salas et al, 2015; Sottilare et al, 2017; Fletcher & Sottilare, 2017). Scaling training development relies in part on efficient ways to find and maintain relevant content, and to assist training developers with discriminating between content supporting individual learning objectives and team learning objectives (Bonner et al, 2016). Supporting a scalable means of analyzing information and its alignment with competencies requires new automation tools. The remainder of this paper presents our recent work extending the underlying techniques used by MAGIC, creating a prototype interface for training developers, integration with an open-source competency framework management suite, and development of an API. A schematic depiction of MAGIC is shown in Figure 1, illustrating the one-time process of training MAGIC on a desired application domain (left pane, shaded blue) and its subsequent use by an instructional developer (right pane, shaded green).

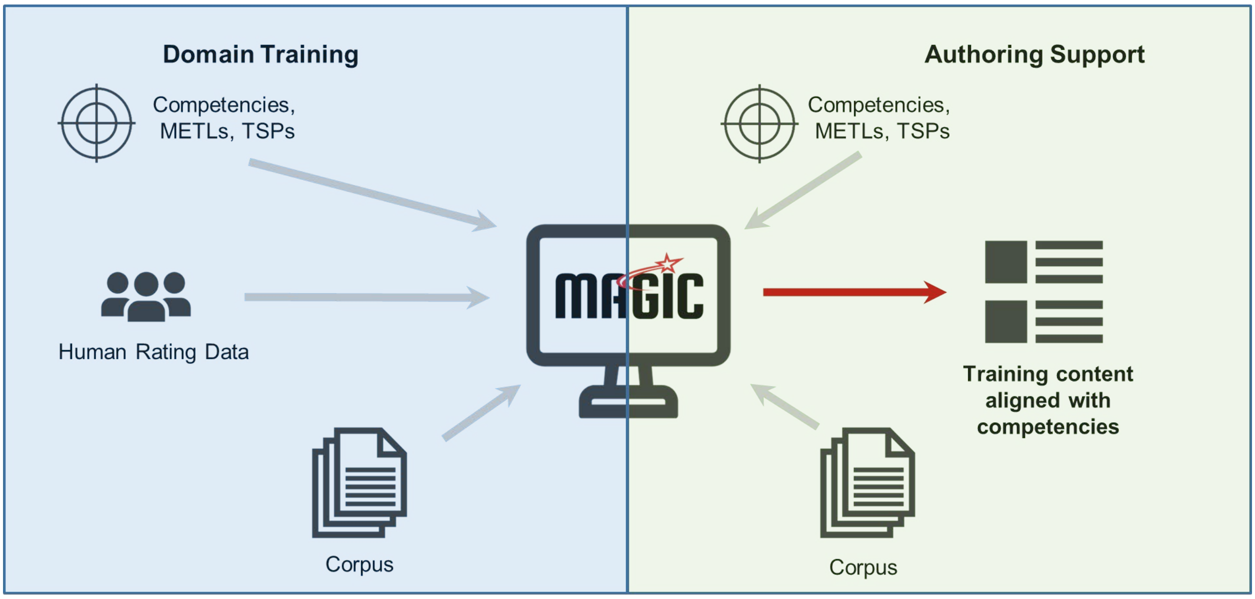


Figure 1. MAGIC at a glance.

STREAMLINING TRAINING DEVELOPMENT BY ALIGNING CONTENT WITH COMPETENCIES

Although structured to function as a service for a diverse range of training development contents, MAGIC uses the GIFT authoring process as a metaphor for organizing the workflow and user interface (UI). At the top level, MAGIC's content and results are organized into projects, each consisting of a document collection ("corpus"), a selected set of competencies, and the results, i.e., associations MAGIC generated between competencies and segments of content from the corpus. A training developer can save and manage multiple projects.

Within a project workflow, the training developer selects the competencies for which the training is being constructed. These competencies can be imported directly from CaSS, or created using the CaSS competency framework editor, which runs within MAGIC. The training developer also selects a target corpus of documents to be analyzed. This corpus can be modified (e.g., by adding additional document) via the UI. Selecting and modifying these two components (competencies and a corpus) are independent activities in MAGIC and be accomplished in whole or in part and in any order.

Once the competencies and corpus have been defined, the training developer uses MAGIC to generate a collection of content excerpts from across the selected documents, each excerpt tagged by the competency with which it is aligned, and labelled as aligning with either individual or team training. The user can navigate a graphical map view of the related competencies to select, view, filter, and, when available, compare MAGIC's rating with human rater results.

MAGIC's filters enable the user to select a specific group of competencies and/or a specific group of documents in the corpus from which to view the alignment results. While viewing results with any given set of filters applied, the training developer can select the content excerpts to be saved for use in the current project. Selected excerpts may be reviewed for original context and tagging details. These selections and results may be exported in JSON format.

ACHIEVING STRONG ALIGNMENT WITH MACHINE LEARNING

Approach

MAGIC uses ML and NLP techniques to train algorithms that associate content with learning objectives, and that tag content as having individual or team relevance. We developed three sets of ML models for our initial research and testing: (1) *unsupervised general* models trained using Wikipedia and the New York Times Annotated Corpus to map concepts; (2) *unsupervised domain-specific* models trained with military-sourced documents to define domain-specific concepts; (3) *supervised, domain-specific* models trained with human-tagged data from a team of instructional designers and subject-matter experts to enhance outcomes.

For our exemplar use case (creating training for maneuver battle drills), we manually created competencies outlined as hierarchical task procedures, based on original document text, and manually tagged content with task type as depicted in Figure 2. The manually tagged competencies were used to train the ML algorithms for task type detection.

COMPETENCY ID	PARENT ID	Standard or Competency	Team or Individual
BD10-S1		*All Soldiers don their protective mask within nine seconds (or fifteen seconds for masks with a hood).*	*Individual*
BD10-S1-T1	BD10-S1	Element dons their protective mask.	Individual
BD10-S1-T2	BD10-S1	Element gives vocal or nonvocal alarm.	Team
BD10-S1-T3	BD10-S1	Element uses the appropriate skin decontamination kit within one minute for individual decontamination, as necessary.	Team
BD10-S2		*Soldiers assume MOPP4 within eight minutes.*	*Individual*

Figure 2. Example competencies captured from an infantry maneuver battle drill.

To create the labeled data set, we used a team of three human raters with instructional design, research, and military backgrounds, led by an expert in instructional design. Raters were trained on the rating task, which included scoring relevance of sections of content to a learning objective and tagging with individual/team identifiers. The resulting labeled data set consists of 3,132 items and was segmented into two data sets: one for training the supervised learning models, and one for evaluating performance of all three ML model sets. The interrater reliability (n=3) was substantial for text selection and extraction (Fleiss' kappa 0.82, with a 95% confidence interval of 0.74-0.90) and nearly complete for distinguishing team and individual content (Fleiss' kappa 0.88, with a 95% confidence interval of 0.78-0.97).

A Novel Capability: Concept Embedding

MAGIC demonstrates a novel capability for matching content excerpts to a competency (typically a short text string) rather than to a topic (typically supported by larger amounts of descriptive text). Aligning content to a short text sample is especially difficult in specialized domains. While using pre-trained word embeddings, trained on large general domain corpora (e.g.Wikipedia), in conjunction with deep learning models yields good results on tasks in the general domain, they are a poor fit for tasks in highly technical or specialized domains. Moreover, for specialized domains, the available in-domain corpora are often too small, resulting in poor performance of embeddings trained on those corpora. To address this difficulty, we extended existing work in embedding techniques (e.g. Word2Vec, GLoVe) (Mikolov et al, 2013; Pennington et al, 2014, Bojanowski et al, 2017) and most recently ELMo (Peters et al, 2018), to develop a new technique we refer to as *concept embedding,* which extends the previous approaches in contextualized embeddings to improve performance in specialized domains.

The approach involves parsing an input corpus of general and adjacent domain documents to detect entities and relations as short phrases (rather than as individual words) using TensorFlow- or SyntaxNet-style dependency parsing along with traditional ontological approaches (Goldberg & Levy, 2014). In the next step, we build corpus models using the resulting dependency trees as input into distinct entity and relation embedding models, where 'concepts' are defined as tight clusters of phrases in the resulting vector spaces (Levy & Goldberg, 2014). By mapping entities and relations separately and linking them in a combined (modified W2V-SG) model, we instantiate concepts as tight clusters of phrases in the resulting entity and relation vector spaces. For example, we might instantiate the concept "*Santa Claus*" as associated with "*Jolly Old St. Nick*" and "*the man in the red suit*." (Li et al, 2016; Shalaby et al, 2018).

Whereas early iterations used static embedding approaches, we extended the approach by using recently developed dynamic embedding methods that consider context, specifically ELMo (Peters et al, 2018). We then construct a third embedding model, trained solely on a small in-domain corpus in the target specialized domain, and combine the representations using either concatenation or a variant of ELMo_{mix} (El Boukkouri et al, 2019). Finally, we apply this novel embedding approach to the matching and classification tasks described above. Specifically, we use the vectors of the labeled training dataset to build a model for each task (matching content to learning objectives and classifying training content as either team or individual).

We trained classifiers using traditional SVM approaches. To match content to competencies, we use a measure of similarity (e.g. MaLSTM) and differential weights for matching entity-concepts and relation-concepts (Mueller & Thyagarajan, 2016). We experimented with using direct comparison measures of vector representations of text to measure similarity without supervised learning techniques. While this approach produces results generally inferior to those obtained by running the vector representations through a machine learning model trained on labeled data, it was still surprisingly effective, suggesting that direct comparison metrics can serve as a viable alternative in the absence of labeled data sets needed for supervised learning.

Previous analysis examined results comparing the performance of MAGIC on a text selection task to human raters, under the three conditions described above: (1) generic domain model, unsupervised learning; (2) domain-specific model, unsupervised learning; (3) domain-specific model, supervised learning). The results demonstrated the algorithms performing slightly below human performance when using only the domain-general unsupervised model, at or near human performance when adding the unsupervised domain-specific model, and slightly above human performance when adding the supervised domain-specific model (Bell, Brawner, Robson, Brown & Kelsey, 2019).

INTEGRATION: MAGIC API FOR USE IN GIFT, STE AND BEYOND

MAGIC is envisioned as a service to be used seamlessly across a diversity of training development environments. To standardize how competencies are managed, we have designed MAGIC to import competency frameworks from the open-source CaSS environment, implemented as a MAGIC-specific abstraction layer in CaSS. For interoperability of MAGIC as an overall service, we are developing a RESTful API that supports detailed interaction with and review of the MAGIC data models and outputs. The MAGIC API uses a machine readable JSON format for all payloads. Currently, the API-accessible services include content alignment, discovery, selection, and organization tasks related to reuse and aggregation features of a generic authoring environment. This API will support the needs of open and proprietary authoring tools within the STE ecosystem and provide sufficient API services and documentation for future MAGIC integration.

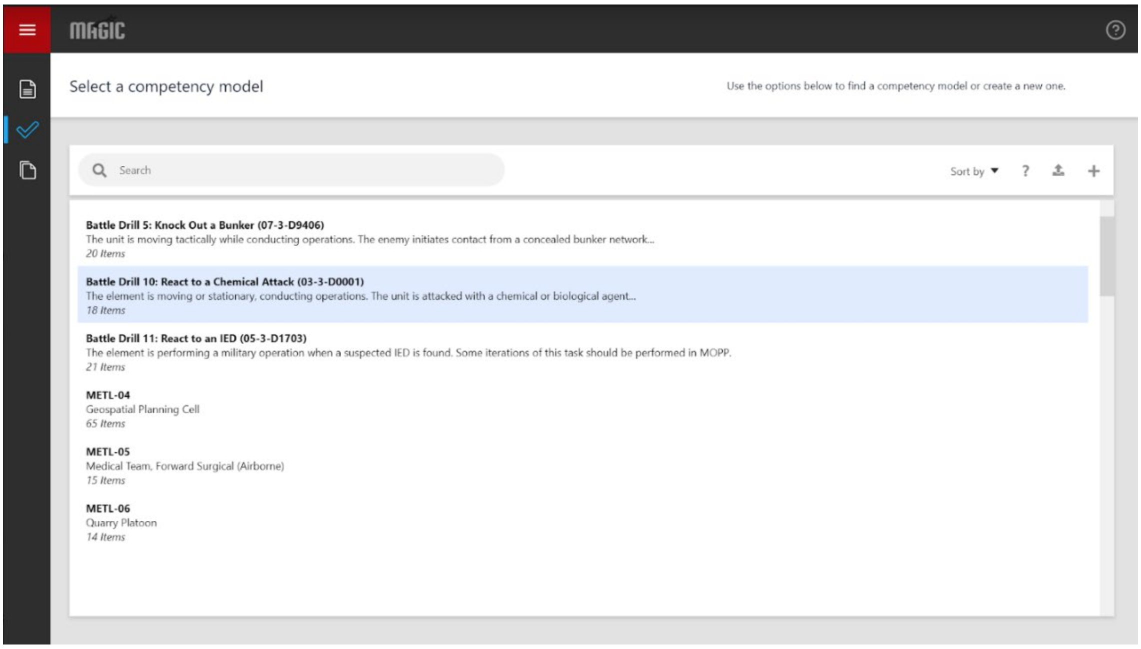


Figure 3. Selecting a competency framework using CaSS within MAGIC.

BRIEF EXAMPLE

Our demonstration exemplar uses competencies derived from battle drills in the infantry maneuver domain. The competency frameworks are represented in CaSS and available in MAGIC for review and selection (Figure 3). Within a framework, specific competencies can be selected or excluded (Figure 4).

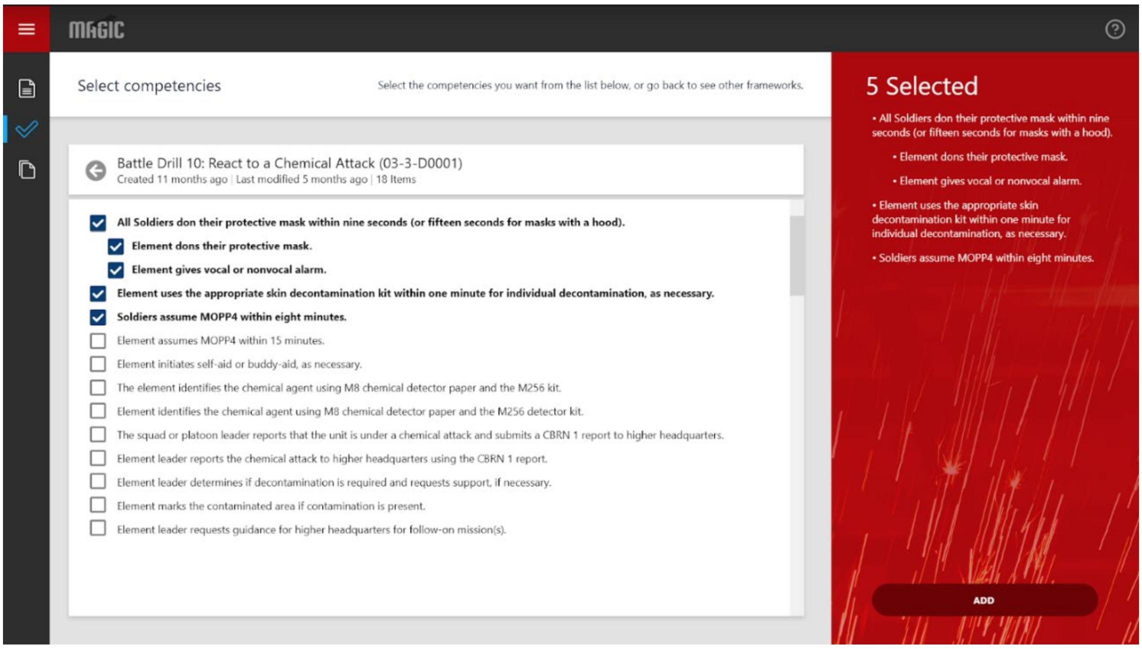


Figure 4. Selecting competencies from within a CaSS framework in CaSS in MAGIC.

To populate the collection of content from which a corpus can be created, we used the Central Army Registry (CAR) and the Milgaming Portal's Training Support Packages (TSPs), resulting in a collection of over 1,200 documents. Figure 5 shows the document management interface for tailoring the corpus.

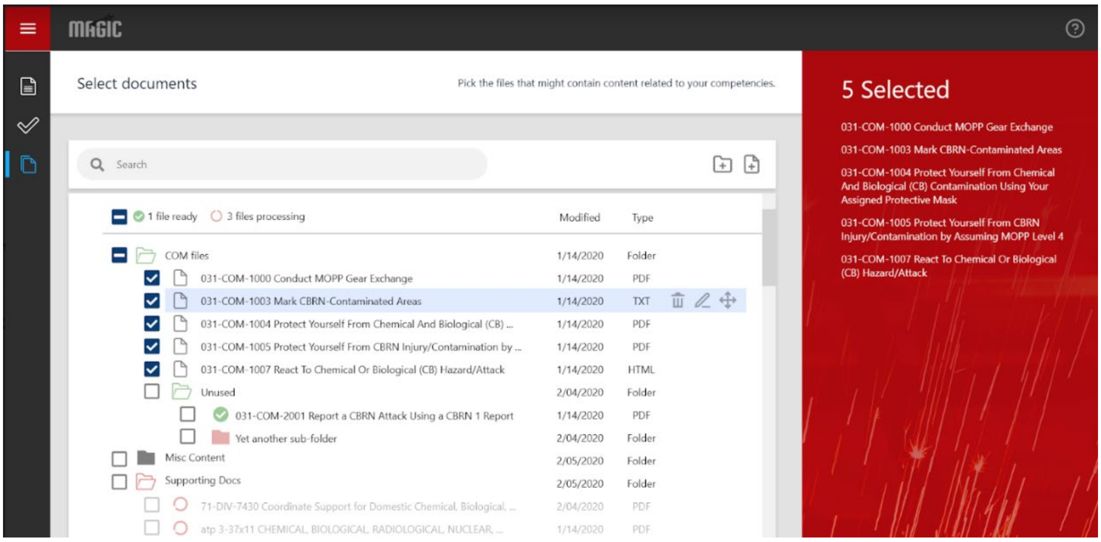


Figure 5. Selecting corpus documents that will be used in a project.

Once the competencies and corpus have been defined, the training developer uses MAGIC to generate a collection of content excerpts from across the selected documents, each excerpt tagged by the competency with which it is aligned, and labelled as aligning with either individual or team training. The user can navigate a graphical map view of the related competencies (Figure 6) to select, view, and filter the results (Figure 7). During this process, the training developer can select content excerpts to be included in the current project and review them for original context and tagging details (Figure 8).

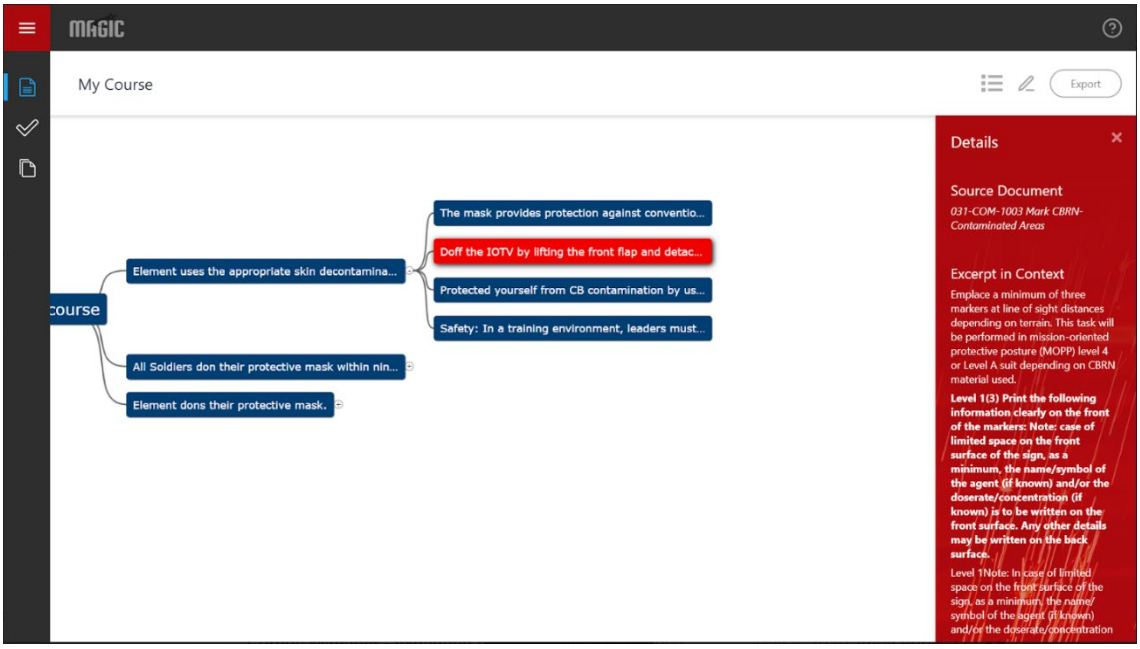


Figure 6. Using the competency map to review content alignments.

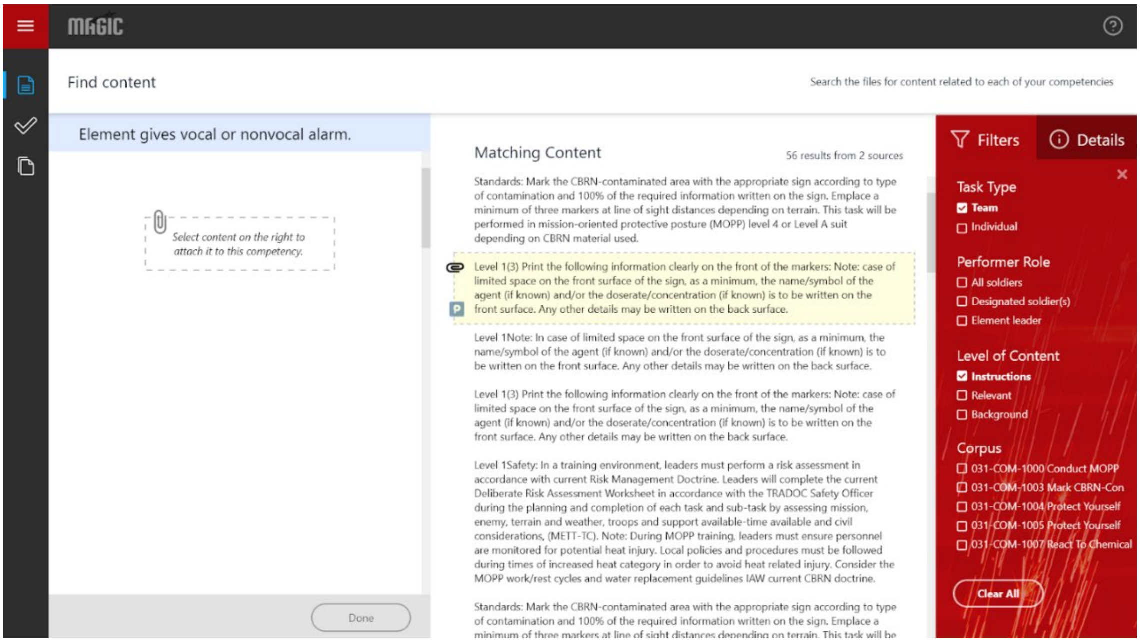


Figure 7. Filtering the excerpts found for each competency; adding content to project.

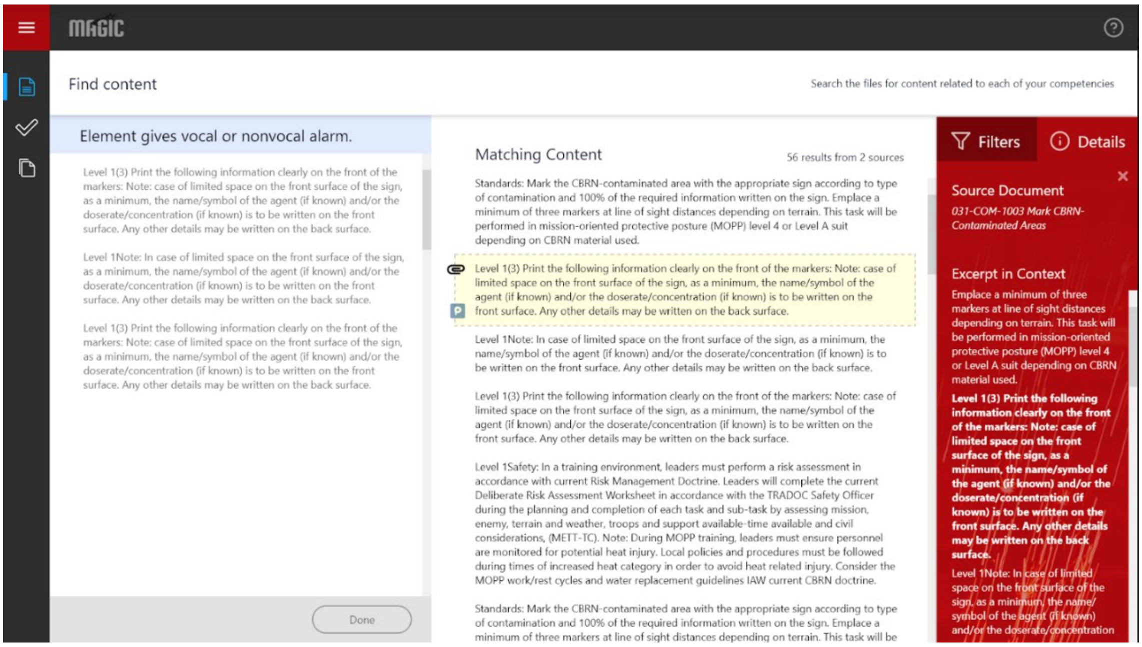


Figure 8. Reviewing the details for a specific content excerpt.

CONCLUSIONS AND FUTURE WORK

The work we have presented will advance the state-of-the-art in applying machine learning and NLP to authoring and development of training and in particular team tutoring, and will extend GIFT by supporting authors in collecting and aligning content with individual and team competencies. Preliminary results summarized in this paper show that MAGIC meets human rater levels of reliability using combined unsupervised general and domain specific models, and outperforms human raters with the addition of supervised domain-specific models. Based on these early findings, we see the potential for automated content discovery using competency auto-alignment and text extraction to yield faster, scalable team training development processes.

Integration of MAGIC services into the GIFT authoring workflows could propel reuse of training materials, while helping training developers overcome the challenges of distinguishing content supporting team or individual learning. The MAGIC services API described in this paper provides an integration path to support training development in broader contexts, including several STE-driven training environments. The interoperability of MAGIC with CaSS presented in this paper further positions MAGIC as an enabler for automatically tagging instructional content in a competency-based training landscape.

Next steps in this effort include testing and evaluation of MAGIC with authors of team training simulations; and the integration of MAGIC services with Army-selected authoring/CMS/LMS tools.

REFERENCES

Bell, B., Brawner, K., Robson, E., Brown, D., & Kelsey, E. (2019). Authoring Team Tutors in GIFT: An Automated Tool for Alignment of Content to Learning Objectives. In *Proceedings of the 7th Annual GIFT Users Symposium* (p. 140). US Army Combat Capabilities Development Command–Soldier Center.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135-146.

Bonner, Desmond; Gilbert, Stephen B.; Dorneich, Michael C.; Winer, Eliot; Sinatra, Anne M.; Slavina, Anna; MacAllister, Anastacia; and Holub, Joseph, (2016) "The Challenges of Building Intelligent Tutoring Systems for Teams". *Industrial and Manufacturing Systems Engineering Conference Proceedings and Posters*. 23. Retrieved from http://lib.dr.iastate.edu/imse_conf/23

El Boukkouri, H., Ferret, O., Lavergne, T., & Zweigenbaum, P. (2019, July). Embedding Strategies for Specialized Domains: Application to Clinical Entity Recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop* (pp. 295-301).

Fletcher, J. D., & Sottilare, R. A. (2017). Shared mental models in support of adaptive instruction of collective tasks using GIFT. *International Journal of Artificial Intelligence in Education*. https://doi.org/10.1007/s40593-017-0147-y.

Gilbert, S., Slavina, A., Dorneich, M., Sinatra, A., Bonner, D., Johnston, J., Holub, J., MacAllister, A., and Winer, E. (2017). Creating a team tutor using GIFT. *International Journal of Artificial Intelligence in Education.* https://doi.org/10.1007/s40593-017-0151-2.

Goldberg, Y., & Levy, O. (2014). word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722.

Levy, O., & Goldberg, Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (Volume 2: Short Papers) (Vol. 2, pp. 302-308).

Li, C., Wang, H., Zhang, Z., Sun, A., & Ma, Z. (2016, July). Topic modeling for short texts with auxiliary word embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 165-174). ACM.

McCormack, R., Kilcullen, T., Sinatra, A.M., Brown, T., & Beaubian, J. (2018). Scenarios for Training Teamwork Skills in Virtual Environments with GIFT. In Proceedings of the 6th Annual *Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym6)*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).

Mueller, J., & Thyagarajan, A. (2016, March). Siamese recurrent architectures for learning sentence similarity. In *thirtieth AAAI conference on artificial intelligence*.

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).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Salas, E. , Shuffler, M. L., Thayer, A. L., Bedwell, W. L. and Lazzara, E. H. (2015), Understanding and Improving

Robson, R., & Poltrack, J. (2017). Using competencies to map performance across multiple activities. In *Proc. of the Interservice/Industry Training Systems & Education Conference*, Orlando, December 2017.

Teamwork in Organizations: A Scientifically Based Practical Guide. *Human Resource Management*, 54: 599-622. doi:10.1002/hrm.21628

Shalaby, W., Zadrozny, W., & Jin, H. (2018). Beyond word embeddings: learning entity and concept representations from large scale knowledge bases. *Information Retrieval Journal*, 1-18.

Sinatra, A.M. (2018). Team Models in the Generalized Intelligent Framework for Tutoring: 2018 Update. In Proceedings of the 6th Annual *Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym6)*.

Sottilare, R., Holden, H., Brawner, K., & Goldberg, B. (2011). Challenges and emerging concepts in the development of adaptive, computer-based tutoring systems for team training. In *Proceedings of the Interservice/Industry Training Systems & Education Conference*, Orlando, December 2011.

Sottilare, R., Holden, H., Goldberg, B., & Brawner, K. (2013). The Generalized Intelligent Framework for Tutoring (GIFT). In Best, C., Galanis, G., Kerry, J. and Sottilare, R. (Eds.) *Fundamental Issues in Defence Simulation & Training*. Ashgate Publishing.

Sottilare, R. A., Burke, C. S., Salas, E., Sinatra, A. M., Johnston, J. H., & Gilbert, S. B. (2017). Towards a design process for adaptive instruction of Teams: A Meta-Analysis*. International Journal of Artificial Intelligence in Education*. https://doi.org/10.1007/s40593-017-0146-z.

Sottilare, R., Graesser, A., Hu, X., and Sinatra, A. (Eds.). (2018). *Design Recommendations for Intelligent Tutoring Systems: Volume 6 - Team Tutoring*. Orlando, FL: U.S. Army Research Laboratory. ISBN 978-0-9977257-4-2. Available at: https://gifttutoring.org/documents/ and on Google Play.

ABOUT THE AUTHORS

Dr. Benjamin Bell *is the Principal Investigator on the MAGIC project and president of Eduworks. He has been been leading funded research in education, training and simulation for over twenty years, with an emphasis on defense and national security applications. He holds a Ph.D. from Northwestern University in Artificial Intelligence, Master's degrees from Embry Riddle and Drexel University, and a Bachelor's degree from the University of Pennsylvania.*

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.*

Debbie Brown *acts as Learning Engineer, Instructional Designer, and Software Developer at Eduworks, and has been active in eLearning research and development for 20 years in academic, workforce, and government/military applications. Since joining Eduworks, she has served on multiple projects using GIFT.*

Elaine Kelsey *is a research engineer at Eduworks, focusing on applications of natural language processing and machine learning in topic and concept detection and automated text generation. She designed and led the development of Eduworks' automated question generation algorithm, and is currently developing solutions in human-interpretable machine learning. She has multiple Bachelors and Masters degrees in computer science, linguistics, and molecular biology, and is currently working towards a PhD in Computational Linguistics.*

THEME V: COLLECTIVE/TEAM BASED MEASURES

Intelligent Adaptive Training in the Synthetic Training Environment, 2020 Update

Gregory A. Goodwin[1] and Michael Hoffman[2]
U.S. Army Research Laboratory – Human Research and Engineering Directorate[1], Dignitas Technologies[2]

ABSTRACT

The Synthetic Training Environment (STE) is one of the Army's top 6 priorities for Army modernization. The STE is intended to provide a distributed, collective training capability to the Army. Initial operational capability of STE is projected for FY21 and final operational capability is planned for FY23. One key element of the STE is its training management tools (TMT). The TMT element of STE is supposed to enable leaders to plan, prepare, execute, and assess collective training with minimal effort and additional support needed. STE's requirements documents specifically identify intelligent tutoring as a capability that will be needed in the TMT. Because the STE is primarily being developed for collective training, there is a pressing need for the development of team level adaptive instructional capabilities. This year, GIFT's core capabilities will be integrated into the current STE baseline which should provide unprecedented opportunities to transition Science and Technology (S&T) products developed with GIFT to the STE. This paper will discuss emerging S&T challenges and needs for adaptive instructional systems as they relate to the STE.

BACKGROUND

In 2018, the U.S. Army stood up a fourth major command called the Army Futures Command (AFC). The AFC was established because our near peers have been gaining on us technologically over the last decade and Army leaders felt that this was at least in part due to a failure by the U.S. to effectively bring innovations to the warfighter (2019 Army Modernization Strategy). The acquisitions process which has been in place since WW2, with decades long schedules for developing, fielding, and sustaining new systems, is no longer adequate. Technical innovations in the commercial sector evolve so quickly that those timelines render many new systems obsolete long before they ever entered production.

To re-establish technological overmatch, the AFC identified six modernization priorities: long-range precision fires, next generation combat vehicles, future vertical lift, network, air and missile defense, and Soldier lethality. In addition, the AFC established cross-functional teams (CFTs) to accelerate the acquisition of specific key technologies under these modernization priorities. All of these capabilities are needed to engage in a type of warfare known as multi-domain operations or MDO.

Multi-Domain Operations

Russia is the current pacing threat to the U.S. Through operations in Crimea, Eastern Ukraine, and Syria, Russia has shown that it has developed capabilities to use information to manipulate proxies, deploy sophisticated cyber weapons, precision strike weapons, and is developing autonomous weapons with artificial intelligence (AI). China has also been investing heavily in AI tools, bio-engineering, cyber weapons, hypersonics, and space-based weapon systems.

These near peer competitors use and are developing multiple layers of standoff across all domains (air, land, sea, cyber, and space) to prevent the US from employing its conventional forces effectively. This allows them to have uncontested influence over regions that are of strategic importance to them. The multi-domain operations concept involves penetrating, disabling, and/or circumventing these layers of standoff without depending on armed conflict to enable the U.S. to engage adversaries on favorable terms in a conventional fight.

The MDO force will be a joint, multinational force that will consist of modernized, tailorable formations called force packages. These force packages will include networked teams of manned and unmanned platforms and will leverage significant electronic and cyber warfare capabilities. All of the Army's six modernization priorities are integral to the Army's plan to modernize for the MDO[4] fight.

Cross Functional Teams

As noted above, CFTs are charged with accelerating the acquisition process for key capabilities aligned with the Army's modernization priorities. CFTs were actually created before AFC existed through Army Directive 2017-24 (McCarthy, 2017). According to this directive the CFT's role is to reach out to industry, academia, and the warfighter in an iterative process to support a material development decision (MDD). The final product that the CFT must produce is a draft capability document.

There are eight CFT's aligned with the Army's six modernization priorities. Each CFT focuses on development of a single system with the goal of accelerating the acquisition process. There are two CFT's under the Soldier Lethality modernization priority: the STE CFT, and the Soldier Lethality CFT which is developing the Integrated Visual Augmentation System (IVAS) for dismounted soldiers. The IVAS system will be integrated with the STE to provide an augmented reality training mode for dismounted Soldiers.

As noted, CFTs are supposed to work across traditional divides of academia, industry, government, and warfighters to develop requirements informed by experimentation and technical demonstrations in support of the MDD. In the case of the STE CFT, the Information System (IS) rules of the Joint Integrated Capabilities Development System (JCIDS), also known as the Information Technology Box approach. This approach allows the government to use a more agile development process needed for software. The STE CFT is currently working to deliver initial operational capability (IOC) by the fourth quarter of FY21 and final operational capability (FOC) by fourth quarter of FY23.

THE SYNTHETIC TRAINING ENVIRONMENT

The Synthetic Training Environment will provide a collective, combined-arms training capability for squads through Army Service Component-Command (ASCC). The STE will provide this collective training across a common synthetic environment (CSE) for live, virtual, constructive, and gaming training in MDO.

The STE information system (STE-IS) includes three major components: 1) Training Simulation Software (TSS), 2) One World Terrain (OWT), and 3) Training Management Tools (TMT). These three components are described below.

TSS – this is the underlying simulation engine for STE. It also has architectural functions and is responsible for delivering STE training to the point of need. The TSS will have to solve problems of scalability from a nine-member squad to potentially millions of entities involved in brigade sized operations in dense urban terrain. It will provide intelligent automated forces that are able to employ battlefield tactics of any enemy we may face. TSS will also have to accurately simulate weapon systems and their associated battlefield effects of both U.S., coalition partners, and our adversaries.

OWT– this is a single solution for global terrain. Army simulators currently employ over 50 different terrain formats which creates a number of challenges in distributed simulation training. The concept of OWT is that a single source file will contain all layers of scenery to include elevation, land use, hydrology, sub-terranean, satellite imagery, as well as human terrain data like population density, political, social, and cultural data. The government will define standard data parameters for all layers. All applications will reference that common source to generate run-time representations for various applications (e.g., command post, dismounted, air, armor, etc.).

TMT – these tools will support the planning, preparation, execution, and assessment of training events. One of the key goals of TMT is to reduce the time and overhead associated with running large collective training events. To do this, TMT will have to leverage big data tools and AI to automatically design training scenarios that are

[4] Draft Pamphlet 525-3-X explains that at brigade and below, units engage in cross-domain maneuver, not multi-domain operations.

tailored to the training unit based on past experience and mission needs, automate assessments and provide data visualization both during and after training execution.

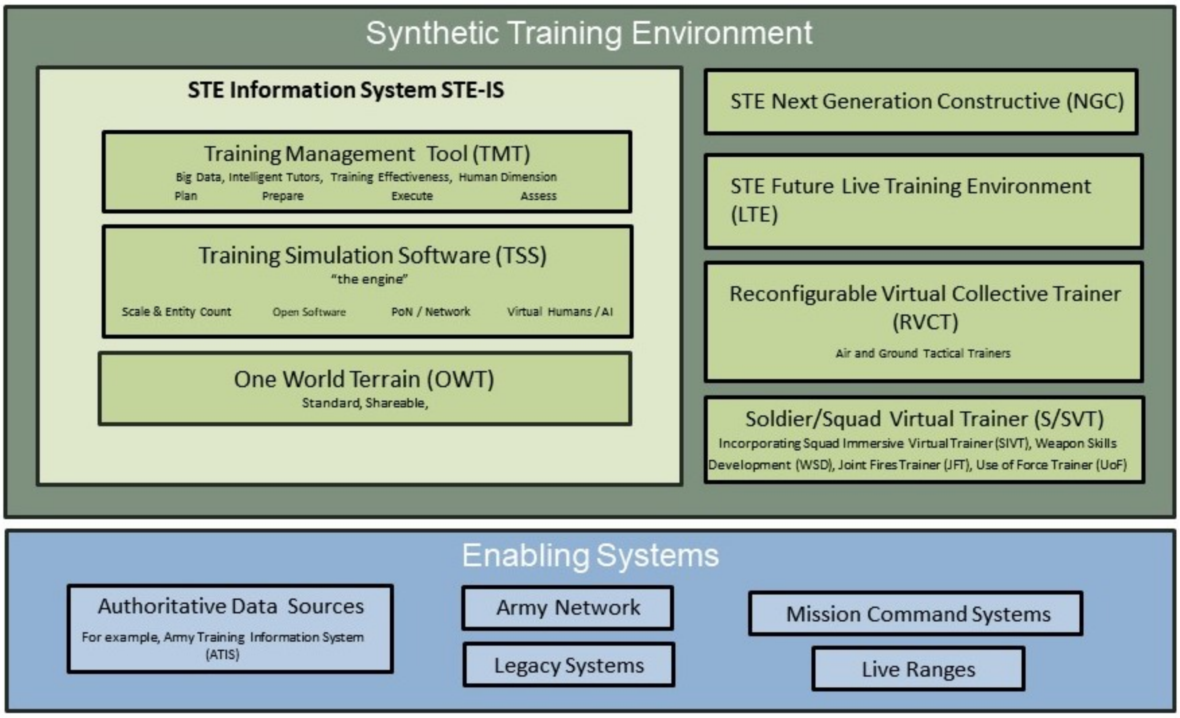


Figure 14 Synthetic Training Environment Framework

Other STE training systems including the future Live Training Environment (LTE), the future Next Generation Constructive (NGC) environment, the Reconfigurable Virtual Collective Trainer (RVCT), and the Soldier/Squad Virtual Trainer (S/SVT) will interface with STE-IS through standard API's and message protocols.

RVCT – This STE component is the next generation of combined arms tactical trainers (CATT). The design approach is to use a smaller footprint than the legacy CATT systems by leveraging mixed reality technologies to produce systems that can be reconfigured with the flip of a software switch to emulate a wide range of air and ground vehicles. These solutions contain a minimal physical representation of the air and ground systems.

S/SVT – this will enable Soldiers and squads to conduct squad, weapon, and joint fires training as well as use of force training. The individual elements are known as Weapon Skills Development (WSD), Joint Fires Training (JFT), and the Squad Immersive Virtual Trainer (SiVT). The goal is to be able to provide all of these in a system that is man-transportable and deployable worldwide.

NGC - Next generation constructive will provide a scalable representation of the battlefield for Brigade through ASCC leaders and staffs. Soldiers and units will use the NGC to train tasks that support Joint, Unified Land Operations (ULO) and MDO. The NGC will adjudicate the interaction of Soldiers, Units, computer generated forces (CGF), environmental effects, weather, weapons effects, and operations to ensure 'fair fight' outcomes.

LTE – The Future Live Training envirement seeks to overcome the shortcomings of current force on force live training systems. Current laser based engagement systems only work for line of sight engagements. Because of this, units don't have a way to simulate the effects of indirect fires, guided munitions, and other modern weapons in live force on force training. The LTE will also provide a more realistic determination of battlefield effects.

Finally, the STE-IS will need to access other authoritative databases for unit training and operational history as well as personnel and facilities management systems. It will also rely on Army networks, live training facilities and will need to interface with some operational systems like mission command systems to provide a common synthetic environment around live, virtual, and constructive training venues.

GIFT INTEGRATION WITH STE

Perhaps the most significant development in 2020 has been the integration of GIFT with the TMT. This year we received funding from the materiel developer to to provide the STE with an intelligent tutoring capability. To do this, we are building a so called "headless" version of GIFT, what we are calling a Run Time Assessment (RTA) version of GIFT.

In this RTA, the tutor-user interface and course authoring tools will be hidden/disabled/or simplified for the RTA so that TMT can maintain control of the scenario-based training being conducted in STE. Though disabled, the STE developer will be able to easily turn these features on to incorporate them into their front-end interface. GIFT will provide its Domain Knowledge File (DKF) functions to do real-time assessments and recommend remediation tactics, but the TMT will have the final say on which of those tactics to implement.

Several assumptions are implicit in this design approach. First, GIFT will primarily provide real time assessments of actions performed in a virtual environment but it will not present lesson material, surveys or other GIFT course objects. Because of this, there will be no need for the tutor-user interface (e.g., login, course selection, etc.). In addition, the GIFT course authoring tool UI will be simplified to allow creation and management of virtual training course objects and corresponding DKFs without the need to create a course. Finally, we assume that any pedagogical tactics like feedback or scenario adaptations will be sent to the TMT for adjudication. The general framework of the RTA is shown in figure 2.

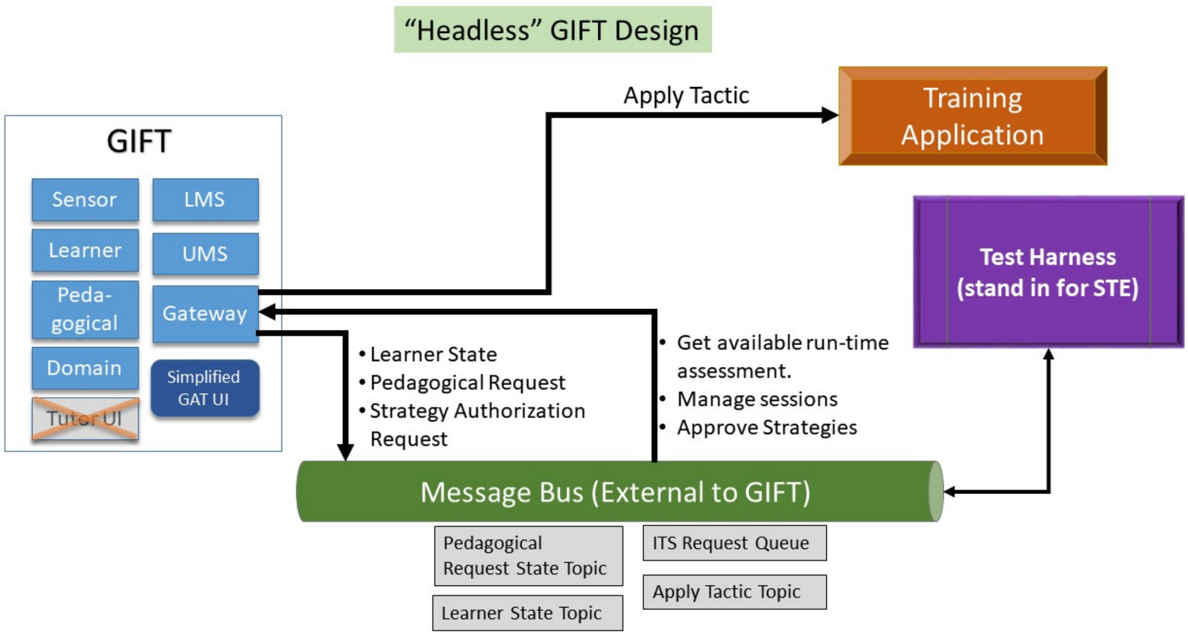


Figure 15. Headless GIFT Design

As has always been the case, our design approach is to remain training application agnostic. Therefore, communication with the TMT will be through the Gateway module that will provide the API for TMT to request and receive data from GIFT. The TMT will be able to request real-time session state for all current and past sessions including learner state, pedagogical requests, and suggested tactics. The TMT will also be able to start and stop GIFT, choose appropriate DKFs for use in a scenario, and ensure that GIFT is receiving state information from the training application in use. GIFT will also be able to implement tactics directly in the training application using its API (e.g. alter opposing force size/training/weapons to increase or decrease difficulty).

To further facilitate interoperability, a common message broker software will be utilized such as ActiveMQ or Pulsar to abstract the TMT from the RTA solution that GIFT is provided. The information sent over this message bus will leverage Google's Protobuf to create needed message classes in application appropriate languages that Protobuf supports. This approach has been used widely in industry and represents the current state of the art for messaging layers in networked microservice oriented architectures. It should be noted that this approach could be used to replace the use of DIS and HLA standards once a set of STE required messages has been defined.

The huge benefit of the RTA for the community is that GIFT can serve as a transition target for STE. As we as a community develop and validate new condition classes, DKF's, team and learner models, etc., we can implement them relatively easily in the STE through updates to the RTA version of GIFT. We are continuing to develop, test and release a full version of GIFT, but our plan is to maintain concurrency with the RTA through a common, core architecture.

Ares, the battlespace visualization tool, is also being integrated into the TMT. You may recall from last year that we have been developing an interface between GIFT and Ares (Davis, Riley, & Goldberg, 2019). GIFT leverages Ares as a more sophisticated battlespace visualization tool, beyond what the Game Master can provide. Ares uses GIFT to provide team organization structure/assignments, performance assessment, visual queue requests and altered training application data streams. In the future we hope GIFT will also provide other learner state information to Ares such as one of the many learner characteristics that can be ascertained in GIFT like motivation, Grit and competency to name a few.

There are two main ways GIFT communicates with Ares. The first is through the Ares REST API. Over this channel, GIFT sends learner state information regarding performance, team organization structure, learner names and their assignments to a team member role and corresponding entity in the training environment. Ares uses this information to alter the appearance of entities visualized in the battlespace such as using color coding scheme based on poor performance, showing a learner's name next to an icon or highlighting an entity to draw an observer's attention to its location. For the second form of communication, GIFT alters the data stream coming from the training application (e.g. Virtual Battlespace or VBS).

This data stream is typically DIS traffic. GIFT consumes the DIS traffic from the training application and then applies several layers of logic in order to manipulate what is visualized in Ares. These manipulations include ways to indicate whether an entity needs to be removed from visualization (because either the training application doesn't populate the correct information on its own, say when a lifeform mounts a vehicle or GIFT could be requesting that a higher echelon needs to be shown) and showing a team with appropriate echelon enumeration rather than an entity on the battlefield, something most DIS use cases don't support. The plan is to continue this integration across the various modalities that Ares provides such as AR/VR and to expose more information for Ares to present to the observer in a useful way that doesn't overwhelm their task load.

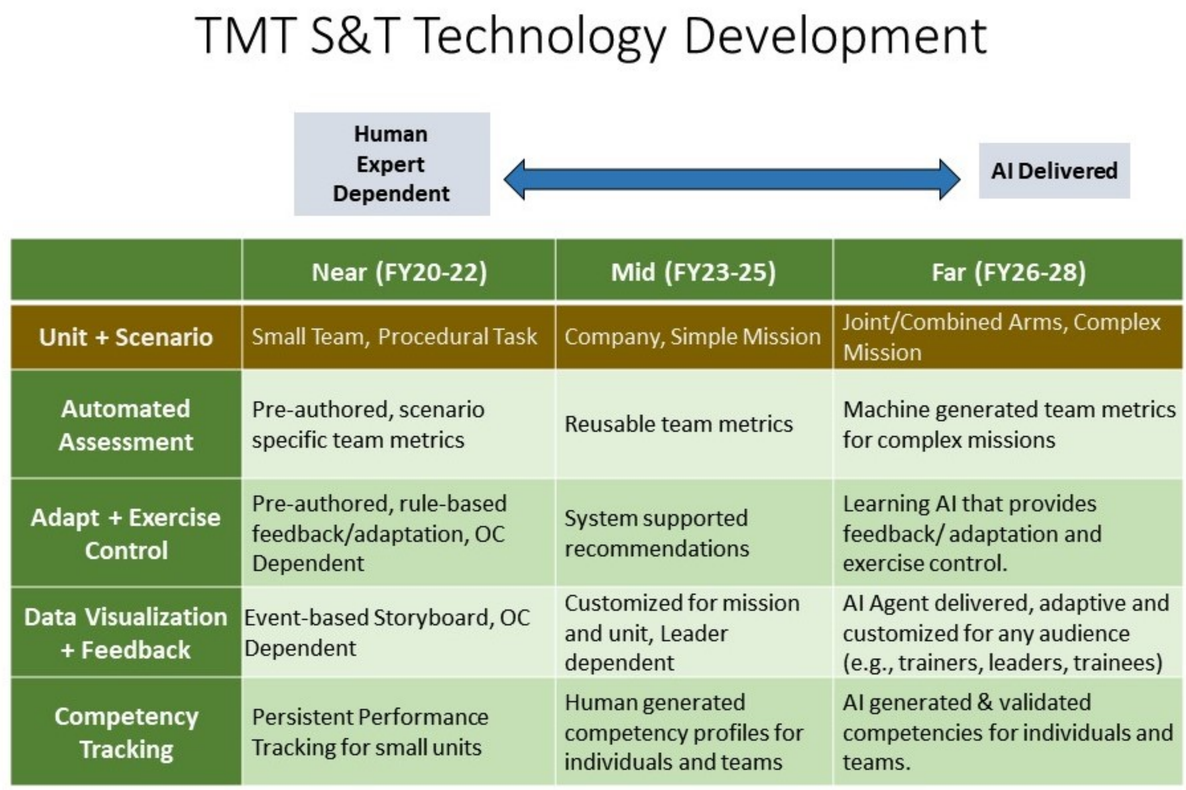
2020 RESEARCH GOALS

Back in November of 2018, we met with the capability developer, the materiel developer, and the capability manager and developed a list of technology gaps relevant to the TMT component of the STE-IS. For several days we discussed the capabilities needed for the TMT and the places where S&T was needed to close gaps. The three S&T focus areas we came up with are listed below in items 1-3. This year, we added a fourth focus area – competency tracking for teams.

1. Automated team assessments
2. Data visualization and feedback, especially at the team level
3. Intelligent adaptive training for teams
4. Competency tracking for teams

A couple of things to note about these priorities: first, as we have discussed, the S&T challenge is primarily at the team level. There are increasingly more commercial solutions that provide measurement, feedback, and adaptation at the individual level. While there are certainly some S&T challenges for individual training, we believe team training, which is core to any collective training event, is where we need to focus our efforts (Goodwin, Johnston, Sottilare, Brawner, Sinatra, & Graesser, 2015). Second, it should also be noted that these four challenge areas are not independent of each other. For example, automated assessments are necessary to feed automated adaptation or data visualization, or competency tracking.

To further illustrate how we think these capability gaps/S&T areas will change over time, we have identified near, mid, and far term descriptions of capability growth (see figure 3). For all of these, in the near term, we see a heavy dependence on human expert input but over time, we expect greater automation as machine learning and artificial intelligence play a greater and greater role. We also think that in the near term, these capabilities will be focused at the small unit/small team level during the execution of well-defined tasks, but over time these capabilities will work for larger teams performing complex, ill-defined tasks.



	Near (FY20-22)	Mid (FY23-25)	Far (FY26-28)
Unit + Scenario	Small Team, Procedural Task	Company, Simple Mission	Joint/Combined Arms, Complex Mission
Automated Assessment	Pre-authored, scenario specific team metrics	Reusable team metrics	Machine generated team metrics for complex missions
Adapt + Exercise Control	Pre-authored, rule-based feedback/adaptation, OC Dependent	System supported recommendations	Learning AI that provides feedback/ adaptation and exercise control.
Data Visualization + Feedback	Event-based Storyboard, OC Dependent	Customized for mission and unit, Leader dependent	AI Agent delivered, adaptive and customized for any audience (e.g., trainers, leaders, trainees)
Competency Tracking	Persistent Performance Tracking for small units	Human generated competency profiles for individuals and teams	AI generated & validated competencies for individuals and teams.

Figure 16. Technology Challenges for TMT

By presenting this vision for capability development across the near, mid, and far term, you can imagine how our research objectives and priorities will change over time. This is not to say that we assume progress will happen exactly according to this schedule, but certainly in the near term, we are focusing more on research that addresses smaller team, procedural tasks. Over time, you can see that we will increasingly look to address larger teams and teams of teams engaged in more complex mission types.

CONCLUSIONS

This year has seen some important developments for the Adaptive Training research program. One of the most significant developments has been the creation of the RTA version of GIFT for integration into STE. We are expecting to continue to develop RTA for at least the next couple of years so our hope is that this will become a transition target for much of our GIFT development.

We continue to see automated team assessments, automated data visualization and feedback for teams, and intelligent, adaptive training for teams as three foci of our S&T efforts. This year we have added a fourth category: competency tracking for teams.

Automating these four processes is necessary to reduce the substantial overhead cost associated with conducting large-scale collective training, one of the primary requirements of the new synthetic training environment.

We also see a couple of synergies that will happen across these lines of development. First, as we develop better automated team assessments, we will begin to have much richer digital data lakes of unit performance. These data lakes can then be exploited by machine learning and AI. To provide even better feedback and exercise adaptation. Eventually, these tools will be useful for managing not just individual training events, but training at a higher, enterprise level.

For example, machine learning and AI tools will be used to monitor training outcomes across schoolhouses, installations, and units so that effective training practices can be rapidly identified and implemented across the Army while ineffective ones can be eliminated. AI will also help leaders to make tradeoff decisions across domains of time, cost, and training effectiveness to optimize the return on investment of training resources (e.g., Goodwin & Niehaus, 2018).

REFERENCES

Davis, F.C., Riley, J.M., Goldberg, B.S. (2019). Extending GIFT Wrap to Live Training. Proceedings of the 7th Annual GIFT Users Symposium. Orlando, FL.

Goodwin, G., Johnston, J., Sottilare, R, Brawner, K, Sinatra, A., & Graesser, A. (2015). Individual Learner and Team Modeling for Adaptive Training and Education in Support of the US Army Learning Model: Research Outline. ARL Special Report 0336. U.S. Army Research Laboratory.

Goodwin, G., & Niehaus, J.(2018). Modeling training efficiency and return on investment for adaptive training: GIFT integration. Presented at the 6th *Annual GIFT Users Symposium* 9-11 May, 2018, Orlando, FL

McCarthy, R. D. (2017). Cross-Functional Team Pilot in Support of Materiel Development. Army Directive2017-24. Secretary of the Army, Washington D.C.

ABOUT THE AUTHORS

Gregory Goodwin *is a senior research scientist at the Army Research Laboratory-Human Research and Engineering Directorate, Simulation and Training Technology Center (STTC) in Orlando, Florida. His research focuses on methods and tools to maximize the effectiveness of training technologies. After completing his Ph.D. at the State University of New York at Binghamton in 1994, Dr. Goodwin spent three years in a post-doctoral fellowship at the Columbia University College of Physicians and Surgeons followed by a year as a research associate at Duke University Medical Center before joining the faculty at Skidmore College. In 2005, Dr. Goodwin left academia and began working at the Army Research Institute (ARI) field unit at Fort Benning Georgia and six years later, he came to the ARI field unit in Orlando, FL where he has been examining ways to leverage technologies to reduce the cost and improve the effectiveness of training.*

Michael Hoffman *is a senior software engineer at Dignitas Technologies and the technical lead for the GIFT project. He has been responsible for ensuring that the development of GIFT, meeting community requirements, and supporting production ITS systems, ITS research, and the growing user community. Michael manages and contributes support for the GIFT community through various mediums including the GIFT portal (www.GIFTTutoring.org), annual GIFT Symposium conferences and technical exchanges with ARL and their contractors. In addition, he utilizes his expertise in integrating third party capabilities such as software and hardware systems to enable other organizations to integrate GIFT into their training solutions.*

Authoring Collective Training Demonstrations in GIFT

Elyse Burmester
Dignitas Technologies

INTRODUCTION

"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", 2020) As one of the earliest descriptions for the Generalized Intelligent Framework for Tutoring (GIFT) it can still be found on the GIFT Wiki page (gifttutoring.org) and more importantly, GIFT is now even easier to use for authoring instructional training content. In the last eight years GIFT functionality, technical capability and compatibility has changed dramatically. The developers and engineers behind GIFT consistently maintained the functionality and ease-of-use as a high priority in the development of this software. The true constant is the ease of use for authors.

As a frequent developer of scenarios and demonstrations using the training applications integrated with GIFT, the author has developed a streamlined process to take advantage of this ease of use to create training content quickly and effectively.

The backbone of authoring training content in GIFT is the Domain Knowledge File (DKF) Authoring Tool (DAT) – a tool that some users might consider one of the more "advanced" or "complicated" aspects in GIFT. While this paper will specifically reference the VBS3 training application course object, the DKF authoring tool structure is similar for other training application course objects available in GIFT.

The streamlined process defined below is designed to benefit new and current GIFT course developers to embrace the full benefit GIFT offers when integrated with external training applications with minimal programming knowledge.

For a more in-depth description of the technical aspects discussed below, readers are encouraged to explore the documentation provided on gifttutoring.org.

PREPARING TO AUTHOR THE DKF

Before diving into the DKF, there are a few recommendations to help declutter this authoring process. The following assumes GIFT Desktop release version 2019-1 is being used and the training application plug-in has been configured according to requirements detailed at https://gifttutoring.org/projects/gift/wiki/Configuration_Settings_2019-1.

Automated Tasks

The first recommendation is to put together a list, whether it be in a text editor/spreadsheet/etc., that includes all of the tasks/performance measures that are required for your training exercise. These may already be outlined in a lesson plan or evaluation rubric for example, or it can be created entirely from scratch. Some questions to consider when creating this list:

- What tasks will the learner(s) be responsible for completing during the training exercise?
- What concept is being assessed within each task?
- What is the intended learning outcome/goal for each task?
- What are the thresholds for measuring a learners' performance for each task?

The more you can curate ahead of time, the easier it will be to input in GIFT.

Instructional Strategies

After you have your task list outlined, it is also recommended to consider how the system should react to varying levels of learner performance that may occur during execution. GIFT has a number of adaptation options that can be automatically applied by the system when a learner state is identified based on the defined parameters. These are called "Strategies" in the DKF architecture. Strategies are sent from the Pedagogical module and implemented by the Domain module based on changes in learner state. Figure 2 shows some of the high-level strategy options available. Some considerations recommended for this list include:

- What actions taken by the learner would be considered good performance for each task?
- What actions taken by the learner would be considered bad performance for each task?
- What instructional intervention would best correct each poor performance?
- What reinforcements, if any, should be applied when good performance occurs?

Having these branches mapped out ahead of time will speed up the process once you begin using the authoring tool. Now that an outline has been developed, the following sections will explain step-by-step how to implement all of these components into the authoring tool.

RECOMMENDED AUTHORING PROCESS

Once you have opened the DKF editor, also called Real-Time Assessment, your screen should look similar to Figure 1. The panel on the left-hand side of this window serves as the root menu of the editor. It has four tabs available at the top, located just under the words "Real-Time Assessment". Each tab contains different elements of the DKF. Starting from the left-hand side they are labeled: "Tasks", "Strategies", "State Transition", and "Assessment Properties". The green plus sign located to the right of this list is used to add new elements within each tab, excluding "Assessment Properties". We will explore each tab in detail in the following sections but for now we are going to start with the first tab, "Tasks".

Step One: The Task List

GIFT has a number of automated performance measure options, or Performance Nodes, that are available to use with most training applications. A concept is the lowest level performance node and is associated with a java class that contains logic to assess the learner's actions in the domain. A concept is assessed via conditions. The concept/condition hierarchy supports infinite nesting by specifying the "concepts" choice for a concept instead of "conditions".

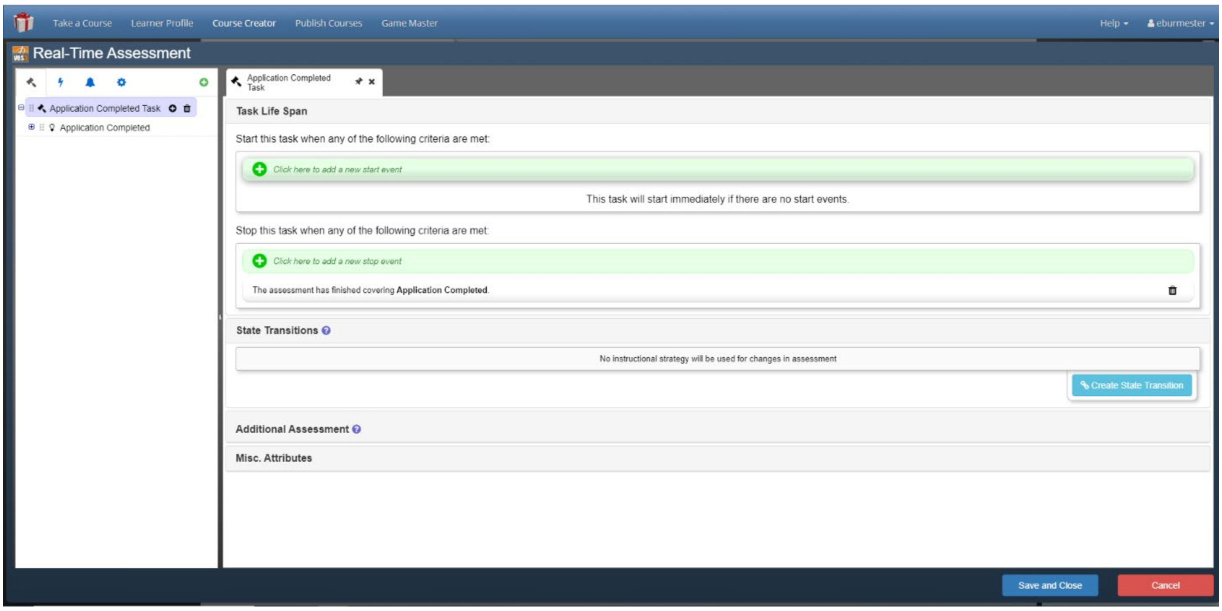


Figure 1. Task Authoring Panel.

Add an item to this task list using the green add button located next to the tabs at the top of the panel. This will add a parent task to the list (indicated by a hammer icon). To add a concept (indicated by a lightbulb icon) under a parent task, use the grey add button located next to the parent tasks' name.

When a concept is added, a second level node will appear below that concept and a list of condition classes will be shown in the panel on the right-hand side. Click through each condition class in this list to find detailed descriptions for the associated assessment logic and evaluation values that are used to drive that conditions' logic. After selecting a condition, enter the appropriate values in the "Real-Time Assessment" section to define the desired assessment logic you've identified in your drafted task outline. The "Overall Assessment" and "Advanced" sections within this panel contain additional options to customize assessments, but these options are not required for validation.

Parent tasks have a lifecycle that must be defined by Start and End triggers, found in the Task Life Span panel when a parent task is selected from the list. Use these triggers to structure the flow of your DKF – the concepts listed under each parent task will only remain active for the duration of the parent tasks' lifecycle. The Start and End triggers are defined using either: 1) the learner's location in the simulated environment, 2) the learner's completion of a task or concept, 3) the learner's performance on a concept, or 4) when a Learner Action is selected (explained further below).

Step Two: Strategy/State Transition Loop

Now that all of the tasks and concepts are defined, click on the Strategies tab (shown in Figure 2). Use the green add button at the top of the left-hand panel to add each new strategy. Unlike the task list though, I recommend adding strategies individually and linking each with a state transition before moving on to author the next strategy. After naming the strategy, click on the green bar in the "Activities" table to add an activity for the strategy to implement. There are two categories of activity options: Instructional Intervention, which includes either sending a feedback message or delivering a media resource to the learner; and Scenario Adaption, which includes options for changing the scenario or lesson in some manner.

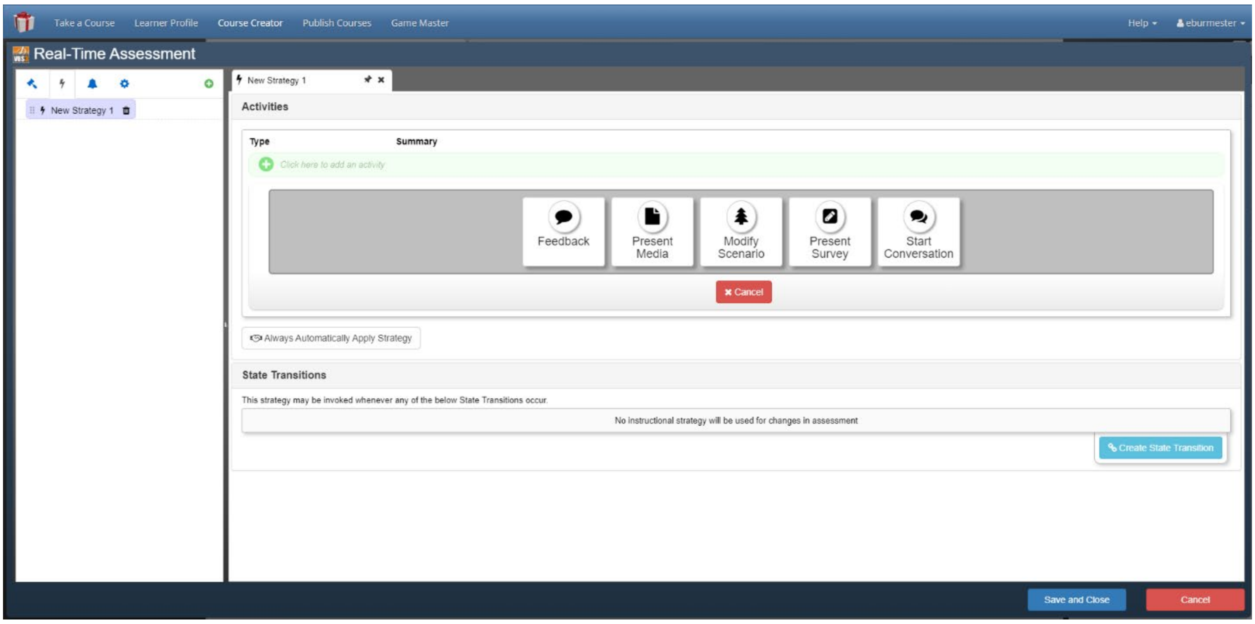


Figure 2. Strategy Authoring Panel.

To author a state transition from this strategy panel, click on the blue button labeled "Create State Transition" below the "State Transitions" section of this panel. This will open a new window, replacing the previous strategy authoring window, in the State Transition tab with the new state transition panel ready for authoring (shown in Figure 3). (Tip: select the push-pin icon in the right-side panel top tab to lock the panel, this will cause the new panel to appear in a new top tab while keeping the previous top tab open.)

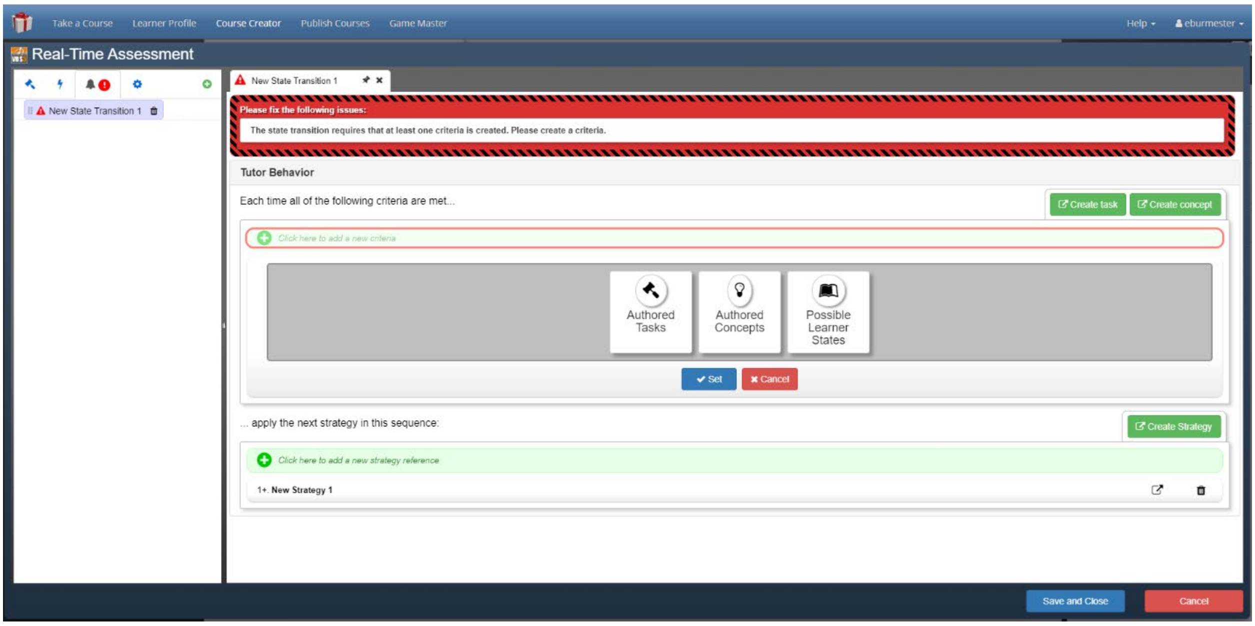


Figure 3. State Transition Panel.

State transitions cause instructional strategies to be sent to the learner based on assessment values defined in a Performance Node. A state transition is "activated" when a logical expression evaluates to true based on the specific learner state attribute(s) changes from the previous value to the current value ("Domain Knowledge File", 2018). To author the state transition logic, click on the green bar under the "Tutor Behavior" table. There are three evaluation criteria options to choose from: an authored Task, an authored Concept, or a possible Learner State. After selecting one of the options, you can then choose the specific attribute and learner state rules from the drop-down menus that appear.

Notice the strategy table at the bottom of the window is already populated with the strategy created from the previous window. You can add additional strategies that will be applied by this state transition using either the green bar in the table to select an existing authored strategy or by clicking the "Create Strategy" button in the top left-hand corner of the strategy table to create a new strategy. Strategies in this table will be applied in the order that they appear.

You have now authored a complete assessment loop for one task or concept, depending on which you chose for the state transition. Repeat these steps for each branch of your outline to complete the core assessment logic for your training scenario.

Step Three: Assessment Properties

The final pieces of the DKF left to add are found under the Assessment Properties tab in the left-hand panel. Two of these properties are necessary to tie your DKF together: Points of Interest and Team Organization. The other properties - End Triggers, Learner Actions, and Miscellaneous – provide additional customization options that are not required and will not be discussed in this paper.

Points of Interest

This section defines the global list of points, paths, or areas associated locations in a virtual environment, e.g. VBS3. These waypoints can be referenced throughout the DKF in tasks, concepts, and conditions. All waypoint name values must be unique within the DKF. Locations can be specified in either Geocentric Coordinates (GCC), Above Ground Location (AGL), or other coordinate systems depending on the need and application being used.

Team Organization

This section defines the hierarchy of teams and team members within your scenario. Both teams and team members can be referenced in various parts of the DKF, such as strategies and conditions. This is beneficial when assessing multiple teams at once, as well as assigning assessments or strategies to a specific team or team member to separate responsibilities. Each level of this hierarchy must have a unique name and this list must contain at least one team member for the learner to link to. Note: for the VBS3 application Entity Markings are recommended to define team members.

Step Four: Test, Test, Test

Now that you have all of the pieces of your DKF put together, it is very important to perform as many interactions of testing as necessary to achieve your desired instructional flow. Very rarely does a DKF work as expected the first time it is tested. For example, you may notice feedback that is not presented at the proper time. This could be due to incorrect values provided in the state transition associated with that strategy. Or, if that state transition is not being triggered by the expected learner performance state this could be due to incorrect values provided in the associated concepts condition logic. Keeping in mind where values are defined or what dependencies they are linked to will help pinpoint where in the DKF certain modifications are needed to reach the expected outcome.

CONCLUSION

The steps defined above have proven effective in authoring collective training content. While this paper only touches on a subset of GIFT's functionality and capabilities, it is intended to serve as a user-friendly introduction to the full-suite of tools available with this software.

REFERENCES

Authoring Guide (GIFT). (2018, August 15). Retrieved from https://gifttutoring.org/projects/gift/wiki/Authoring_Guide_2019-1.

Domain Knowledge File (GIFT). (2018, July 26). Retrieved from https://gifttutoring.org/projects/gift/wiki/Domain_Knowledge_File_2019-1.

Overview - Generalized Intelligent Framework for Tutoring (GIFT). (2020, April 6). Retrieved from https://gifttutoring.org/projects/gift/wiki/Overview

ABOUT THE AUTHOR

Elyse Burmester *is a Research Assistant at Dignitas Technologies. Since 2017, she has contributed as a member of the GIFT development team in addition to directly supporting the Creative and Effective Training Technologies Branch at U.S. Army Combat Capabilities Development Command - Soldier Center (CCDC-SC) SFC Paul Ray Smith Simulation & Training Technology Center (STTC). Ms. Burmester has a bachelor degree in Political Science from Florida Gulf Coast University.*

Automating Assessment and Feedback for Teamwork to Operationalize Team Functional Resilience

J.T. Folsom-Kovarik[1] and Anne M. Sinatra[2]
Soar Technology, Inc. [1], U.S. Army Combat Capabilities Development Command (CCDC) – Soldier Center – Simulation and Training Technology Center (STTC) [2]

INTRODUCTION

The Generalized Intelligent Framework for Tutoring (GIFT) is a framework for intelligent tutoring systems that has been developed by the U.S. Army (Sottilare, Brawner, Sinatra, & Johnson, 2017). GIFT has been designed with reusability in mind, and includes a set of tools that allow intelligent tutors to be efficiently created by individuals who do not have background in computer science (Sottilare, et al., 2017). Design decisions have been made in GIFT to allow for flexibility in both the topic area of the material (domain), as well as the design (there are a number of different approaches that can be taken for tutoring). Additionally, GIFT has been designed to support both creating tutors and research/experiments. The majority of the early work in GIFT was developing the functionality for use with individual training; however, a further goal of GIFT has always been to scale up to provide team tutoring (Sinatra, 2017; Sinatra, 2018). Moving from individual intelligent tutoring to intelligent tutoring for teams is a complex process which requires multiple considerations, including how to connect the team through the technology, how to assess the team, and how to establish authoring tools that give reusable building blocks to describe team tutoring. In the case of GIFT, there is an additional challenge as it is domain independent, which results in the authoring tools needing to be able to support many different team configurations and end goals.

Since the current development focus of GIFT is on teams, there have been a number of projects that have done some initial work in regard to team tutoring with GIFT. A meta-analysis of relevant team tutoring studies was performed, and as a result, a number of different behavioral markers were identified (Sottilare, Salas, Burke, Sinatra, Johnston, & Gilbert, 2018; Sottilare, Burke, Sinatra, Johnston, & Salas, 2020). The outcomes of the meta-analysis helped to guide the theoretical underpinnings of team tutoring in GIFT, with projects subsequently identifying relevant behavioral markers and determining ways to asses them in a computer based team environment (McCormack, Kilcullen, Sinatra, Brown, & Beaubien, 2018; McCormack, Kilcullen, Sinatra, Case & Howard, 2019).

The initial team implementations within GIFT focused on demonstrating the ability from a technology standpoint to assess multiple learners at the same time. Some early team tutoring projects utilized multiple domain knowledge files (DKFs) to assess interactions between the many combinations of individuals in a team (Gilbert et al., 2018; Ostrander et al., 2020), and others focused solely on the team level in order to reduce the amount of DKF authoring necessary (McCormack et al., 2018; McCormack et al., 2019). The current work described here aims to assess a team as a group of contributing individuals without the authoring burden of explicitly specifying interactions between each combination of team members.

In relation to past team assessment research and development in GIFT, two key opportunities to enhance the core technology are (1) linking individual contributions to team outcomes and (2) expressing team performance in a domain by differentiating between heterogenous roles and functions in a team. The current project aims to address these two opportunities with technical changes in the GIFT infrastructure.

First, linking individual contributions to team outcomes refers to assessing and reporting how individual and team assessments are defined, observed, and stored to drive feedback from GIFT. Team assessments include team outcomes, underlying process, and observable behavior markers indicating good or poor teamwork. The easily measured results of team performance are team outcome measures – did the team succeed in a task? But good or poor team outcomes can be achieved because of, or in spite of, good process and individual contributions. The individual actions should be recorded along with their impact on overall team performance.

In a training setting, team process is perhaps more important than team outcomes. Team process describes member interactions such as how information was shared and decisions were made (Mathieu et al., 2008). One framework for describing team process is team dimensional training, described in the next section. The team process antecedents of teamwork are a proper target of training because the same apparent outcomes can be reached through good team process or through poor team process compensated by individual efforts. Furthermore, team outcomes can also be mediated by environment and other factors not directly controlled by the team (Ilgen et al., 2005).

Finally, behavioral markers include actions and attributes that help uncover the hidden antecedents of teamwork. Observable markers should be captured and related to the teamwork inferences they help support. A number of team tutoring relevant behavioral markers for GIFT have previously been identified (Sottilare et al., 2018). However, one of the major challenges in applying these markers in GIFT is that many are traditionally evaluated by a human observer; in order to implement them in a computer-based tutoring system additional work needs to be done to adapt them based on what is possible in the system.

Second, expressing team performance in a domain refers to modeling how teams with different qualities may be expected to perform in that domain. While teamwork skills and markers may be general and reusable, they are typically expressed differently in each domain of instruction or performance. The GIFT DKF is the store of domain-specific information that applies reusable GIFT conditions to a specific simulation and setting. Linking individual contributions to team performance requires a flexible and expressive language for relating GIFT conditions to team assessment inferences. As part of the software implementation enhancing the DKF, additional structures in GIFT are also introduced to enable general remembering and sharing information across conditions. These let GIFT conditions make inferences that link several individual actions.

The team training enhancements should be designed and implemented in a reusable manner that is likely to generalize and apply to many team-training needs. They should also be scalable in the sense that the authoring and computational costs increase as little as possible with the size of the team. Otherwise, teams with many team members or with many relationships between team members could trigger exponential growth in the amount of domain knowledge required to define, capture, and interpret every possible combination of individual actions.

The current status of the work described in this paper at the time of writing is one of transition from design to implementation and refinement. As a result, this paper presents an opportunity for the GIFT community to provide feedback on the designs and ensure they meet the widest possible variety of team training use cases and stakeholder requirements.

The next section of this paper describes a motivating example and some concrete grounding for effective team assessment and feedback. The third section describes the technical approach being designed and implemented to meet the team training needs. Finally, near-term opportunities are discussed and some recommendations for longer-term extensions to GIFT are offered.

TEAM ASSESSMENT

Team tutoring is highly relevant to GIFT, as one of the major intended applications is for U.S. Army relevant training. Many military relevant skills include a team component, and it may also be relevant to train at the Squad level (or above). Implementing reusable, generalized tools that allow for this process is highly useful and applicable in multiple military relevant training domains. Care has been taken to look to the literature to define the general team approaches that are implemented in GIFT (Sottilare et al., 2018), and the current work extends this by framing the work within the concepts of team dimensional training and team functional resilience.

Team Dimensional Training

Team dimensional training (TDT) combines a prescriptive model of teamwork with a structured discussion format to guide after-action review (AAR) for teams (Smith-Jentsch et al., 1998; Smith-Jentsch et al., 2008). The TDT model of teamwork defines four dimensions along which teams with different levels of experience may be differentiated by observable performance. The TDT structured discussion comprises guidelines for effectively communicating about what to sustain and improve as a team.

TDT aligns with military training as demonstrated in settings that range from Army Infantry and tactical combat casualty care (Milham et al., 2017; Johnston et al., 2019) to Navy Aegis destroyer command and control (Smith-Jentsch et al., 2008) as well as joint or coalition training (Giebenrath et al., 2003). In addition, the Department of Defense has teamed with the Department of Health and Human Services to promulgate TDT for teamwork and safety in the medical community within the system Team Strategies & Tools to Enhance Performance and Patient Safety (TeamSTEPPS; Agency, 2013).

TDT dimensions help to motivate the team assessment challenges introduced above. The dimensions describe antecedents of teamwork, such as processes and attitudes, that GIFT must infer from observable behavioral markers. The four team dimensions have been validated in multiple studies (Smith-Jentsch et al., 2008): communication, supporting behavior, leadership or initiative, and information exchange. These are the dimensions which frame the technical approach below. As an example, each of the four TDT dimensions apply to a task in a demonstration scenario: movement as a team.

First, communication describes the manner in which information is delivered. Communication during movement to contact uses doctrinal phrases, gives commands or reports with required data in an expected order, chooses verbal or hand signals correctly, and uses audible voice and clear cadence. Second, supporting behavior describes attitudes and actions that help another team member with a task, such as by correcting an error or anticipating a need for help. During movement supporting behavior might involve flexing formation or zones of coverage to adapt in response to others' movement or loss of coverage. Third, leadership or initiative describes providing guidance to the team. Any team member can display leadership by offering a suggestion or calling out an important fact that may have been overlooked. An example might be the team leader giving orders and delegating tasks, or another team member calling out an observation of the enemy. Fourth, information exchange describes those skills associated with maintaining shared situational awareness, such as knowing what information to share and with whom, or understanding the team status and big picture. During movement this might include maintaining visual contact with the team, communicating planned movement with hand signals, or providing a report to higher.

Using the TDT dimensions, an AAR reviews the events during training in a guided discussion (Smith-Jentsch et al., 2008). A team leader provides AAR guidance, but the discussion is not didactic. Instead it is a forum for self-reflection and self-critique. The structure of a productive AAR is intended to help generalize beyond the specifics of what happened in one training scenario to reveal how the specific events are expressions of underlying team principles. As a result, it is important to highlight the key decision points and individual contributions during a discussion that the leader guides to be comprehensive: it includes all four dimensions, it includes both positive and negative examples, and it involves all the team members rather than a vocal subset.

Adaptive feedback in the form of support for guided AAR discussion is the end goal of the current research. Support is in the form of an adaptive list of discussion topics based on team performance. Experienced leaders can use this kind of support to help track all the split-second behaviors that suggest team processes they want to bring up. In addition, an output for leaders which structures the training events in terms of TDT can assist in how they provide their guidance to trainees. Some leaders could use the TDT structure to improve their discussion with trainees and keep it balanced, focused on key points, and engaging all team members rather than picking on one. By providing a theoretically based structure to the AAR, team assessment can support the leaders and facilitate their feedback in a quick and efficient manner. As a result, GIFT can help leaders with many of the existing challenges of an AAR.

Team Functional Resilience

Team functional resilience (TFR) is a complex cognitive team-level skill that describes a team's ability to carry out their missions when a team member is lost or performs a function below expectations (Neville et al., 2020). Although researchers sometimes consider team functions to consist of a list of team members' job tasks, expertise, or job titles (Bunderson & Sutcliffe, 2002), GIFT team assessment is well aligned with a broader definition. Team function is defined here to include the full set of teamwork antecedents and behavioral markers describing process and outcomes in a changing environment. When team members contribute different functions in this sense to the overall performance, the team as a whole must show resilience to recover from any loss or degradation of a function one team member provides. This resilience may be assessed and may motivate the selection of markers to assess.

Team functional resilience can be enabled or enhanced through TDT skills. Intuitively, application examples include information sharing to create shared situational awareness of functions and needs, joint anomaly detection to provide support for a lost function, communication abilities to enable coordinated action, and cognitive flexibility to respond to unexpected situations with leadership. TFR is a concept tightly related to linking individual contributions to team outcomes.

Team functional resilience is vital to mission success. Army teams at every echelon are composed of individuals providing different functions and capabilities to the overall enterprise – that is, the team members fill heterogenous roles. Functional differences are necessary for reasons such as differing assignments and specialized training and abilities; they also exist due to accidents of location and circumstance. Within a team setting, reasons that an individual might fail to provide a function the team needs might include casualty during a mission, longer-term illness or unit understrength, stress, cognitive load, lack of knowledge or skill, or team-level process failures that can result in momentary or ongoing poor performance evidenced in individuals.

The U.S. Army currently prepares for functional gaps by cross-training, designating a chain of personnel to assume command of important functions, and logistics planning for attrition (e.g., Anglemyer et al., 2018). Simulation training also helps teams prepare for contingencies by introducing failure conditions in a safe environment, followed by AAR and supported discussion to analyze the corrective actions taken and to drive home lessons learned. The TDT framework is one way to structure surfacing and addressing opportunities for a team to sustain and improve TFR when they use a training simulation. Feedback support structured by TDT will let GIFT collect, interpret, prioritize, and present team and individual actions for leaders who conduct an AAR.

A new GIFT support tool for team leaders, rather than for presentation to individuals, has been designed but not implemented at the time of this writing. Adaptive support as an outcome of TFR assessment will include both a chronological view of the exercise history and an interpretive view that combines individual actions into priority discussion topics about failures and corrective actions in each team dimension. This kind of support has the potential to help expert leaders to capture concrete actions that might take place in a very brief time window during a fast-paced simulation. The support can also help less-expert leaders to follow TDT guidelines for presenting a prioritized and balanced AAR, with positive and negative examples for each team dimension and content to engage all team members instead of focusing on a few. As a result of inferring team process from individual behavioral markers, the automated assessment of TFR will generate feedback that aligns with established Army best practices for effective AAR.

Team Training Example

A scenario is briefly described to make concrete the team concepts and technical approach. The scenario is presented to a team using GIFT and Virtual Battle Space 3 (VBS3; https://bisimulations.com/products/vbs3).

For the purposes of useful military training, it is important to understand who is using the training, what they know already, and what the team is expected to learn. The implementation and tasks that occur during the scenario were discussed with Subject Matter Experts and referenced to Soldier's manuals (Headquarters, 2009).

The example scenario is designed for an Infantry squad. The scenario supports team integration after one or more new members join the squad from advanced individual training (AIT). Collective training tasks at AIT do not include clearing a room or building, so the scenario will introduce that task. In such an example the team (other than the new members) is likely experienced, has deployed together, and now is focused on training for team integration. To focus training on team integration, the squad leader has previously shown the new members what to do in a one-on-one walkthrough and has talked through the tasks to accomplish the mission. Now, the GIFT VBS3 training is used to run a full-speed scenario with the new members and the existing team operating together. The scenario assesses the knowledge and skills associated with the collective task in the context of the full team performance.

The scenario depicts a squad on patrol receiving intel that a high-value target (HVT) may be present in a building nearby. The environment is cordoned off by a Company of U.S. Infantry to reduce variables in play during training. The squad acts as a team to carry out (1) movement to contact when approaching the HVT building and (2) entering a building to capture the HVT.

First, movement to contact in a town setting demonstrates coordinated action. The squad must carry out bounding overwatch, in which team members take turns moving to cover positions and providing the moving team members with security by overwatch (suppressing the enemy or preparing to react immediately if an enemy is encountered). A new GIFT capability to assess coordinated actions across individuals detects instances when the alternation is disrupted. The squad must also divide and visually clear several windows in surrounding buildings as they move. GIFT can determine not only whether all the windows were cleared in a timely manner, but also which team member may have contributed to any team failure.

Second, entering a building demonstrates scalable and reusable definitions of team concepts. In the training, the contents of the building could include an HVT, armed hostiles, both, or neither. The GIFT team conditions are defined to work in all cases. It is easy to imagine that the contents of the building could vary further—there might also be an explosive device present, or an open laptop, or an injured person requiring medical attention. One aspect of scalability is writing team conditions to work under all possible combinations of what the Soldiers encounter inside the building. This is important for reducing the authoring effort required to vary a training scenario and adaptively select scenario variants.

Finally, the team training example also contains a functional failure injection to assess team functional resilience. When a four-person fire team enters the HVT building, the second person can become a casualty immediately upon entering the room and fail to cover a sector of the entry building as a result. Different corrective actions are possible. If the third person to enter adjusts their direction of movement immediately, the room geometry can be covered with a balanced distribution of Soldiers. If the third person does not move to cover the area but the fourth person does, corrective action was still taken but less optimally. GIFT can combine observable facts to detect when a team function was not provided, and which individual corrected the lost team function, to provide differential assessment of team functional resilience.

TECHNICAL DESIGN FOR SCALABLE TEAM ASSESSMENT

This paper discusses three aspects of the technical approach: extending the definition of a GIFT Team or TeamMember with Role and Responsibility, extending GIFT Conditions to coordinate with world state memory and information sharing, and applying roles and world state for scalable reuse including adaptive feedback for team training variants.

Defining Individual Contributions with Team Roles and Responsibilities

Using code that was merged to the GIFT trunk within the year since the last GIFT Symposium, the publicly released GIFT DKF defines teams with a configurable hierarchy containing any number of Team levels and with TeamMember leaf nodes corresponding to simulation entities. TeamMembers include learners but also non-player characters and even vehicles. This data structure enables describing each learner as a member of nested teams. A learner may be a member of both a squad and a platoon, with the stipulation that the squad must be contained within the platoon. Conditions in turn can be configured in the DKF to assess performance for the platoon, the squad, or the individual.

Using new code that is being created for the current project, the experimental technical approach to expressing team functions meets the additional design requirement that functions may be carried out by many different team members in groups that cross team boundaries. In other words, a GIFT condition should be able to assess whether an observable action was carried out by an expected person, by another person as a corrective action, or by a suboptimal person.

The need to describe more flexible team relationships is met by defining a Role and a Responsibility. In the current work, the Role is a new property of a Team or TeamMember. A Role can have a many-to-many relationship, so that one TeamMember has several Roles and several TeamMembers can have the same Role. In the example scenario, a learner on the team may be in the role of a U.S. Soldier, a Sergeant, a squad leader, a member of the entry team, and the fourth person stacked up for entry into the HVT building. At the same time there may be another squad leader who is a learner sharing that same role, but different by virtue of being in a security role rather than an entry role.

Each Role may refer to one or more Responsibility entries defined in the DKF. The Responsibility is implemented with Conditions that test whether the corresponding team functions have been met or whether corrective action is needed. As examples, a squad leader may have a positive responsibility to communicate a plan to the team and a negative responsibility to limit the overall number of actions personally completed (because an effective leader should delegate some tasks).

To illustrate linking team performance to individual contributions, an example is given with a fire team of four Soldiers within the squad scenario described above. If GIFT uses a team condition to express that the four Soldiers must visually scan the four sectors around them, then a failure of the team can be recorded if two Soldiers look left and none look forward. Such a condition was developed prior to this work using a Team to specify four Soldiers and summing any gap in their visual scans. By defining a Responsibility, the condition can furthermore record, e.g., which Soldier should have been scanning forward based on their possibly changing team formation. Each position in a doctrinal formation can be modeled as a Role with the Responsibility to watch in a certain direction. As a result, GIFT can record both the team performance and the individual contribution in cases when knowing both makes feedback more effective.

Looking ahead to the assembly of AAR support from observed actions, a Role gives an alternative way to refer to individual contributions. It becomes possible to express categorical descriptors like “you should have sent a senior enlisted” instead of “you should have sent one of Jones, Smith, Harris, or Terrell.” In projected near-future work, Roles will also be assigned and changed during the course of a scenario. This will enable expressing situational descriptors like “the closest Soldier” or “the Soldier who received a certain order.”

Coordinating across Time and Conditions

GIFT conditions are currently implemented as Java classes and by convention are self-contained and modular. In other words, Conditions do not know about other Conditions and do not share information. Most Conditions also do not store information between events. Instead, a typical design pattern is to gather the latest information about the simulation world each time a Condition is evaluated. When Conditions store information from past events, it is kept local or private to the Condition. This has advantages such as freedom from unpredictable side effects. However, the implementation of Conditions without a shared store of past information limits the ability to make inferences based on more than the moment at hand or the single message being processed.

An experimental approach is suggested to enable Condition implementation classes to communicate and store memories. A world state model is introduced, which can be used or ignored by Conditions. The world state model functions as a whiteboard where Conditions can write information to be read later by themselves or others. In the future case where the world state model is widely used, it can be likened to a common operating picture which is assembled to coordinate what many units know about their views of a battlespace.

The demonstration scenario contains an example of processing that uses the world state to assess TDT support behavior. Support behavior requires a combination of a team function failure followed by a corrective action by another, supporting team member. In the example where one Soldier is a scripted casualty during building entry, the detecting Condition records the failure to cover a sector of the room in the world state model. Then, another Condition that detects coverage by a Soldier not normally responsible for the task can use the remembered fact to pair the two events and assign credit for supporting behavior.

Since a goal of the world state model is enabling inferences from observations made across time, like the delay interval or the learner who took a corrective action, the world state model combines the knowledge of current state with a history of past states. The history can be queried according to filters that are useful in the current use case, and may be extended in future work. As examples, it is possible to query a list of actions taken by a specific TeamMember or a Role. The history of past world states differs from memory of, for example, performance assessments because this information is not necessarily for presentation to the learners. Instead it is intended to enable inference by Conditions that are programmed to know about it.

The proof of concept implementation structures the world state model as a hash map with string keys and object values. The use of string keys enables Conditions to be configured in a DKF so that they store specific members in known locations. The keys can be chosen to match for sharing information with other Conditions, or to differ if keeping information separate. Additional exploration is needed to determine the range of use cases for a shared world state memory. It may be that other sharing schemes are preferable, such as explicitly linking Conditions when they should share information.

Examples of coordinating via shared information are numerous. During movement to contact, one Soldier should have established overwatch before another Soldier moves out of cover. When entering the HVT building, the second Soldier entering at a point should move in the opposite direction of the first Soldier. Both of these are accomplished by storing an object to memorize the implementation-specific information needed by newly created Conditions.

Existing Condition classes could also benefit from access to memory about the state of the world. Since the existing implementations have been demonstrated and work correctly, the benefit here is in capturing additional information about individual contributions or possibly computational efficiency of an alternative implementation. One example is the EliminateHostilesCondition. Its current implementation is, when evaluated, to check for presence of undamaged hostile simulation entities. This implementation checks the current world state, rather than memory of past states. It could be enhanced to record a memory of the time when each hostile is eliminated, or the learner who should get credit for eliminating each hostile. In this way individual contributions could enhance the overall assessment with information that one learner carried the day in eliminating these hostiles, or that the first hostile was quickly eliminated while the second took a long time. In potential near-future work, many team conditions could be enhanced to store world state information that was available during past event evaluations for later use.

Scalable Reuse

GIFT Concepts for evaluation have a useful property of being evaluated within a relevant subset of a scenario. This is accomplished by defining a Task with StartTrigger and EndTrigger properties to activate and deactivate the evaluation of all Concepts related to each Task. As implied by their names, the StartTrigger and EndTrigger implement an event-driven approach to determining which Concept is relevant.

An experimental extension to the DKF was added which extends the activation and deactivation ability to work for the many-to-many Role and Responsibility constructs. It is anticipated that a large number of Responsibilities may be defined. When future Roles are dynamically assigned, they may also need to update what Roles are relevant. With this enhancement, the same DKF can apply defined Responsibilities correctly to several variants of a scenario without rewriting. Separability of well-defined Roles may also offer reduced effort to author unit-specific standard operating procedures (SOPs) and updates for evolving tactics, techniques, and procedures (TTPs). The technical approach being implemented is to create overlays that let the DKF control the settings or world state that make Responsibilities relevant.

Overlays parallel the StartTrigger and EndTrigger properties with new properties called StartFact and EndFact. Facts are drawn from the world state model. Examples of facts include the presence of hostiles in a defined area or the presence of an HVT in the defined area. Recall that the world state is disconnected from the simulation environment, so that the presence of hostiles in the world state model is a fact put there by a Condition and not synonymous with the information in VBS3. A correctly implemented Condition would update the world state quickly to avoid desynchronization. The benefit of referring to the world state model is the added layer of semantic meaning that interprets data from simulator messages. The world state model can turn X and Y entity locations into meaningful facts like “hostiles are present.” The world state model can also retain facts from past simulator messages, so the corresponding facts do not vanish and, in this use, support activating or deactivating a Responsibility.

An example of activating only relevant Responsibilities shows how the experimental capability is needed for scalable reuse. When Soldiers enter the HVT building, they have Responsibilities to eliminate armed hostiles and to capture the HVT. Since variants of the scenario exist with and without each of these environmental facts, the DKF needs to specify that responsibility to capture the HVT only exists when the HVT is present (and alive). As an added complication, the responsibility to capture the HVT is of lower priority than the responsibility to eliminate the hostiles and it should not be activated if hostiles are present. This is accomplished by a list of override references that specify a partial ordering of active responsibilities. As a result, GIFT is able to differentiate scenario settings when a team function is not met, and requires corrective action, from settings where a team function is not relevant to evaluate.

Making world-state selection of relevant Responsibilities and Roles more computationally tractable than a long series of if-statements is part of potential near-future work. An approach that has worked in the world of production rule matching is the rete algorithm for pattern matching. There is also likely to be need for increased sophistication compared to a simple priority override system. A more sophisticated system would allow Responsibilities to combine in ways that change their processing or evaluation when multiple combinations of world facts are all true.

DISCUSSION

The experimental changes in GIFT now being implemented as proposed here will help to express key concepts in team assessment. Rather than making possible what was impossible, the changes make DKF authoring more scalable and reusable. Building blocks such as team roles, responsibilities, shared facts about the world, and differential processing based on the current state of the world all contribute to making a single DKF tell GIFT how to observe and infer teamwork antecedents and behavioral markers.

In the big picture, the proposed changes help GIFT to assess more about individual contributions to team outcomes. After a loss of any team function, GIFT no longer needs Conditions to spell out how individuals are expected to respond. Instead, Conditions record the state and cause changes in relevant responsibilities. GIFT can watch for any team member to take corrective action. Who took the corrective action? Was it the best available person? How long did it take? What did the short or long delay mean? All of these can be evaluated efficiently and need not be combined into a single Condition. Finally, what can be inferred from the many possible responses: information sharing, to remain aware of team status; supporting behavior, to step in and fill the missing role; leadership, to choose who best provides the coverage; and communication, to assign corresponding changes and reknit the team.

Some of the remaining challenges in the first year of the project will be addressed in the near term. Voice input (secondarily input from text chat, or simulated internet relay chat for command and control) is key to differentiating how functional loss became known and how corrective action was directed. Recent advances in natural language processing give a path forward to assessing this key communication channel (e.g., Tanaka et al., 2019). Additional changes will make the new roles increasingly functional for real-time assessment and feedback by building on the world state model to give semantic interpretation to role assignments as they change, rather than assigned at the start of training. It will be possible to understand roles learners play based on their movements or an order given. Finally, scalability in terms of efficiently storing and recalling the proliferation of memory elements from large teams and longitudinal tracking of individuals will enable real-time adaptive feedback in addition to support for AAR. The real-time adaptation can include selecting scenario variants based on the capability of roles and world state memory to assess under many variations of scenario presentation.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

In conclusion, the expected benefits to training based on the proposed changes in GIFT will include leader support for principled AAR based on the framework of team dimensional training. An example construct that demonstrates how GIFT can make useful inferences from combinations of behavioral markers is the assessment of team functional resilience. The technical approach to helping GIFT assess team functional resilience includes linking individual contributions to team outcomes, and expressing interactions among team individuals in a manner that is scalable and reusable.

The work described here is in the process of being implemented to demonstrate the proposed changes and their value to GIFT. It is being designed for possible integration with parallel GIFT efforts. This paper is offered as a tool to spur discussion among the GIFT community about the team assessment challenges that can help apply and extend the technical approach to meet as many use cases and requirements as possible.

A long-term recommendation might be extending the experience API (xAPI) standards to encompass teams using structures similar to Roles. GIFT is beginning to use xAPI as a tool for monitoring individual progress between sessions and across training tools. The work to use xAPI or xAPI-like structures to talk about teams or combinations of individuals in a collective training setting would represent a major advance in what xAPI and GIFT could naturally express for long-term experience tracking and sharing.

The work described in the current paper will add to the functionality and flexibility of team tutoring in GIFT. As GIFT continues to be developed, the work can be leveraged in order to support the discussed training examples, as well as implemented within GIFT to support new opportunities for GIFT authors.

REFERENCES

Agency for Healthcare Research and Quality. (2013). *Pocket Guide: TeamSTEPPS 2.0* (AHRQ Publication No. 14-0001-2).

Anglemyer, A., Speten, K., Shingleton, J., Alt, J., & Smith, A.D. (2018). *Predicting US Army First-Term Attrition After Initial Entry Training* (TRAC-M-TR-19-004). Monterey, CA: US Army TRADOC Analysis Center.

Bunderson, J.S., & Sutcliffe, K.M. (2002). Comparing alternative conceptualizations of functional diversity in management teams: Process and performance effects. *Academy of Management Journal, 45*(5), 875-893.

Giebenrath, J.L., Burns, J.J., Hockensmith, T.A., Hession, P.J., Brewer, J., & McDonald, D. (2003). *Extending the Team Learning Methodology to Coalition Training.* Paper presented at the Industry/Interservice Training Systems Conference.

Gilbert, S. B., Slavina, A., Dorneich, M. C., Sinatra, A. M., Bonner, D., Johnston, J., Holub J., MacAllister A., & Winer, E. (2018). Creating a team tutor using GIFT. *International Journal of Artificial Intelligence in Education*, *28*(2), 286-313.

Headquarters - Department of the Army. (2009, June). *Soldier's Manual of Common Tasks Warrior Skills Level 1*. (STP 21-1-SMCT). Washington, DC.

Ilgen, D.R., Hollenbeck, J.R., Johnson, M., & Jundt, D. (2005). Teams in organizations: From input-process-output models to IMOI models. *Annual Review of Psychology, 56*, 517-543.

Johnston, J.H., Phillips, H.L., Milham, L.M., Riddle, D.L., Townsend, L.N., DeCostanza, A.H., et al. (2019). A Team Training Field Research Study: Extending a Theory of Team Development. *Frontiers in Psychology, 10*, 1480.

Mathieu, J., Maynard, M.T., Rapp, T., & Gilson, L. (2008). Team effectiveness 1997-2007: A review of recent advancements and a glimpse into the future. *Journal of management, 34*(3), 410-476.

McCormack, R. K., Kilcullen, T., Sinatra, A. M., Brown, T., & Beaubien, J. M. (2018, May). Scenarios for training teamwork skills in virtual environments with GIFT. In *Proceedings of the Sixth Annual GIFT Users Symposium* (Vol. 6, p. 189). US Army Research Laboratory.

McCormack, R., Kilcullen, T., Sinatra, A. M., Case, A., & Howard, D. (2019, May). Teamwork Training Architecture, Scenarios, and Measures in GIFT. In *Proceedings of the 7th Annual GIFT Users Symposium* (p. 131). US Army Combat Capabilities Development Command–Soldier Center.

Milham, L.M., Phillips, H.L., Ross, W.A., Townsend, L.N., Riddle, D.L., Smith, K.M., et al. (2017). Squad-level training for tactical combat casualty care: Instructional approach and technology assessment. *The Journal of Defense Modeling and Simulation, 14*(4), 345-360.

Ostrander A., Bonner D., Walton J., Slavina A., Ouverson K., Kohl A., Gilbert S., Dorneich M., Sinatra A., Winer E. (2020). Evaluation of an intelligent team tutoring system for a collaborative two-person problem: Surveillance. *Computers in Human Behavior*, *104*, 105873.

Sinatra, A. M. (2017, May). Team Tutoring in the Generalized Intelligent Framework for Tutoring: Current and Future Directions. In *Proceedings of the 5th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym5)* (p. 123).

Sinatra, A. M. (2018, May). Team Models in the Generalized Intelligent Framework for Tutoring: 2018 Update. In *Proceedings of the 6th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym6)* (pp. 169-173).

Smith-Jentsch, K.A., Cannon-Bowers, J.A., Tannenbaum, S.I., & Salas, E. (2008). Guided team self-correction: Impacts on team mental models, processes, and effectiveness. *Small Group Research, 39*(3), 303-327.

Smith-Jentsch, K.A., Zeisig, R.L., Acton, B., & McPherson, J.A. (1998). Team dimensional training: a strategy for guided team self-correction. In J. A. Cannon-Bowers & E. Salas (Eds.), *Making decisions under stress: Implications for individual and team training*.

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.

Sottilare, R., Burke, S., Sinatra, A. M., Johnston, J., & Salas, E. (2020). *Toward a Scientifically Rooted Design Architecture of Team Process and Performance Modeling in Adaptive, Team-Based Intelligent Tutoring Systems*. CCDC Soldier Center, Simulation and Training Technology Center Orlando.

Sottilare, R. A., Brawner, K. W., Sinatra, A. M., & Johnston, J. H. (2017). An updated concept for a Generalized Intelligent Framework for Tutoring (GIFT). Retrieved from *GIFTtutoring.org*.

Tanaka, A., Stensrud, B., Welch, G., Guido-Sanz, F., Sciarini, L.W., & Phillips, H.L. (2019). *The Development and Implementation of Speech Understanding for Medical Handoff Training*. Paper presented at the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), Orlando, FL.

ACKNOWLEDGEMENT

This material is based upon work supported by the U.S. Army Combat Capabilities Development Command Soldier Center (CCDC SC) SFC Paul Ray Smith Simulation & Training Technology Center (STTC) under Contract No. W912CG-19-C-0009. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the CCDC-SC STTC or the U.S. Government.

ABOUT THE AUTHORS

Dr. J.T. Folsom-Kovarik *is a Senior Scientist with Soar Technology, Inc. His research focuses on computational tools for adaptive learning that work under real-world constraints such as efficient representation of complex real-world environments, efficient use of limited data for modeling, and transparency and control for end users.*

Dr. Anne M. Sinatra *is a Research Psychologist, and part of the adaptive training research team within CCDC Soldier Center – STTC. She works on the Generalized Intelligent Framework for Tutoring (GIFT) project. Her background is in Human Factors and Cognitive Psychology.*

Toward Data-Driven Models of Team Feedback in Synthetic Training Environments with GIFT

Jonathan Rowe[1], Randall Spain[1], Benjamin Goldberg[3], Robert Pokorny[2], Bradford Mott[1], and James Lester[1]
North Carolina State University[1], Intelligent Automation, Inc.[2], U.S. Army Combat Capabilities Development Command Soldier Center - STTC[3]

INTRODUCTION

Adaptive instructional systems (AISs) will serve a central role in the Army's future training capability. Using advanced simulation technologies, unit leaders will be able to conduct collective training and mission rehearsal exercises that are tailored to meet the needs of individuals and collective units. A key component of AISs is their capacity to deliver feedback and coaching that supports the acquisition of knowledge, skills, and abilities that enable teams to function effectively. It is widely accepted that feedback is critical for learning (Swart, Nielen, & de Jong, 2019; Wisniewski, Zierer, & Hattie, 2020), because it allows learners to evaluate their progress and performance, identify knowledge gaps, and repair faulty knowledge (Johnson & Priest, 2014; Wouters & van Oostendorp, 2013). However, determining when to present feedback, what type of feedback to deliver, and how it should be realized is a critical challenge, particularly for designing AISs. The presence of multiple team members in collective training exercises presents new opportunities for delivering feedback to learners. For example, during a team training event, feedback can be directed at the team level, the subgroup level, or the individual level (Goldberg, Nye, Lane, & Guadagnoli, 2018; Sottilare, Burke, Salas, Sinatra, Johnston, & Gilbert, 2018). Feedback content and feedback timing can also vary, which has significant implications for team learning and performance.

Developing data-driven models that automatically determine when and how feedback and coaching are provided to teams is critical for realizing the potential of team AISs. It is critical that team coaching models be generalizable across a broad range of tasks and domains to meet the requirements of Army training. There is growing evidence that machine learning techniques, including reinforcement learning (RL) and those based on Markov decision process (MDP) frameworks, provide effective data-driven approaches for modeling pedagogical coaching and feedback in AISs (Chi, VanLehn, Litman, & Jordan, 2011; Rowe & Lester, 2015; Shen, Mostafavi, Barnes, & Chi, 2018; Doroudi, Aleven, & Brunskill, 2019). RL has shown promise for automatically inducing tutorial policies that optimize student learning outcomes without requiring pedagogical policies to be manually programmed or demonstrated by expert tutors. To date, work on RL-based models of tutorial planning has largely focused on individual learners. Extending these methods to deliver coaching and feedback at the team level is an important next step for the community.

In this paper, we describe a new collaborative effort between North Carolina State University, Intelligent Automation Inc., and the U.S. Army Combat Capabilities Development Command Soldier Center to investigate a generalized data-driven tutorial planning framework for automatically delivering run-time feedback to support team performance and learning during collective training tasks in simulation-based training environments. We will leverage the Generalized Intelligent Framework for Tutoring (GIFT) as a platform for developing data-driven models of team coaching and feedback. Recent enhancements to GIFT introduce the opportunity to support team training by providing an observer controller dashboard, also referred to as the GIFT GameMaster, that enables the delivery of tailored assessments, feedback, and scenario adaptations within simulated collective training missions. Our project builds upon these developments by investigating the creation of automated data-driven models of coaching and feedback to support team learning and performance and to augment feedback provided by observer controllers.

We examine several important requirements for supporting generalized data-driven models of team coaching and feedback in GIFT. Distinct from prior work on RL-based tutorial planning outside of GIFT, data-driven team coaching models in GIFT will need to be designed in alignment with the Learning Effect Model (Sottilare et al., 2018), accounting for distinctions between domain-dependent versus domain-independent feature representations within induced coaching policies. In addition to developing best practices for devising input representations for RL-based team coaching, it will be critical to establish a corresponding run-time data pipeline for processing data on the states of learners, the training environment, and adaptive tutor to serve as input to the coaching model and translate coaching strategies into pedagogical tactics for delivery to distributed teams of learners. Another requirement will be balancing between when to follow recommendations of the coaching model versus when to follow coaching strategies that have not yet been fully tested; this is related to the exploration-exploitation tradeoff that is intrinsic to RL. Finally, enhancements to GIFT's Event Reporting Tool will be necessary to provide data filtering and transformation functionalities to support offline investigation of RL-based team coaching models with external machine learning libraries and tools. By addressing these requirements, GIFT will be well positioned to support generalized data-driven team coaching capabilities that show significant promise for enhancing the effectiveness of AISs for collective training within simulation-based training environments.

FEEDBACK AND COACHING IN SIMULATION-BASED TRAINING ENVIRONMENTS

Feedback and coaching play a critical role in supporting learning and skill acquisition. Providing feedback to learners can help learners evaluate their progress, motivate learners to perform better, and reinforce appropriate responses so that individuals are more likely to perform a task correctly in the future (Johnson & Priest, 2014; Kulhavy & Stock, 1989). AISs offer a number of affordances for automatically delivering feedback to support learning in simulation-based training environments. AISs guide learning experiences by tailoring instruction and recommendations based on the goals, needs, and preferences of each learner in the context of domain learning objectives. By leveraging advances in artificial intelligence and machine learning, AISs are envisioned to replicate the tasks of effective human coaches or tutors, monitoring and tracking trainees' learning needs, assessing and diagnosing problems, and providing coaching and assistance as appropriate (TRADOC, 2017).

Several studies have examined how, when, and what type of feedback to present to learners in AISs to improve performance (Serge, Priest, Durlach, & Johnson, 2013; Billings, 2012). To date, much of this research has focused on providing feedback to individual learners. The presence of multiple team members in collective training missions presents new opportunities for delivering feedback to support unit and team training. For example, a squad leader might provide feedback to the entire squad (alright squad, you need to…), to a subset of the squad (alpha team you should...), or to a specific team member. The message could be directed so that everyone hears the message or only a subset of the team receives the message. Thus, the direct targets (i.e., those receiving feedback) and secondary observers of feedback can differ, which has significant implications for team learning and performance. The timing of feedback can also vary. In some cases, an instructor may give feedback while the team is in the middle of performing a task (i.e., immediate feedback). In other cases, a trainer may stop the task and give feedback (i.e., chunked feedback), or alternatively, wait until after a task has been completed to provide feedback (i.e., after-action feedback). Further, the content of the feedback message can vary. In some instances, minimal feedback that informs a learner of whether he or she performed a task correctly will suffice, whereas in other situations the learner may need detailed feedback that provides clear explicit instructions on how to correct errors or perform the task. These design choices for feedback can dramatically affect both the pedagogy of the learning environment and the volume and complexity of feedback that team members receive (Sottilare et al., 2018).

In addition to considering how and when feedback should be delivered to team members, the team training literature suggests that a team's development phase may influence the type of feedback and coaching team members should receive (Kozlowski, Watola, Jensen, Kim & Botero, 2009). In the initial phase of team development, when teams are focused on team orientation and team formation, team members should receive coaching that promotes team identity, team norms, and team expectations. Feedback should help team members learn team norms, expectations, and values. As the team members become more familiar with one another and the team shifts into the next phase of development, feedback and coaching should support individual-level task mastery. Feedback should acknowledge what team members did correctly, address their deficiencies, and promote task self-efficacy and team coordination (Koslowski et al., 2009). Once team members have mastered their individual taskwork skills and understand their roles, coaching should shift towards promoting team performance. Feedback and training should provide insights on how to improve team communication, cooperation, and coordination skills, and promote shared mental models among team members. In the final phase of development, when team members are focused on continuous team improvement, team members should be able to monitor and correct their own performance based on their understanding of the task, team roles and responsibilities, and team performance expectations. Thus, feedback becomes self-directed.

Understanding these phases of team development as well as the competencies and skills that team members must acquire to develop into an expert team is critical for developing AISs that can automatically and continuously improve when and how teams receive feedback.

DATA-DRIVEN TUTORIAL PLANNING IN ADAPTIVE INSTRUCTIONAL SYSTEMS

Recent years have seen growing interest in data-driven approaches to tutorial planning, including reinforcement learning (RL) techniques, which show promise for enabling effective, adaptive feedback across a range of training tasks and learning environments (Chi, VanLehn, Litman, & Jordan, 2011; Rowe & Lester, 2015; Shen, Mostafavi, Barnes, & Chi, 2018; Doroudi et al., 2019). RL provides a framework for automatically inducing feedback policies from observations of learner behavior and outcomes. It reduces the need for pedagogical rules to be specified through manual programming or demonstrations by expert tutors, and it sets the stage for the creation of self-improving instructional systems that can refine their instructional strategies and tactics over time (Sinatra et al., 2019). RL and related techniques have been employed to model tutorial planning in narrative-centered learning environments for middle school science education (Rowe & Lester, 2015; Sawyer, Rowe, & Lester, 2017; Wang, Rowe, Min, Mott, & Lester, 2018), select pedagogical micro-tactics in intelligent tutoring systems for undergraduate logic (Shen, Mostafavi, Barnes, & Chi, 2018), and sequence concepts in educational games for elementary mathematics education (Mandel, Liu, Levine, Brunskill, & Popovic, 2014).

Recent advances in reinforcement learning hold significant potential for data-driven tutorial planning, and extending these methods to enable adaptive feedback and coaching at the team level is a natural next step. A common challenge of RL-based tutorial planning is gathering the necessary data to train and evaluate candidate models for delivering coaching and feedback in AISs. Over the past few years, significant progress has been made in devising off-policy policy evaluation techniques, which provide an offline method for investigating the effectiveness of RL-induced control policies that have been trained using historical data from student interactions with an AIS (Mandel et al., 2014). This is an important development for RL-based tutorial planning, because it provides a more accurate mechanism for evaluating candidate tutorial planning models prior to fielding them with live students in classrooms. Preliminary work investigating off-policy policy evaluation techniques with educational games (Mandel et al., 2014) and intelligent tutoring systems (Doroudi, Aleven, & Brunskill, 2017) have shown significant promise. Another key advance includes recent work on human-shaped RL (Taylor, 2018), which points toward the potential of combining human and machine intelligence by leveraging subjective knowledge from instructors and students about the training effectiveness of RL-based tutorial planners.

HUMAN-IN-THE-LOOP ADAPTIVE INSTRUCTION WITH GIFT GAME MASTER

Over the last decade, GIFT has emerged as an important initiative in addressing the authoring challenges posed by AISs. GIFT is an open source service-oriented framework of software tools, methods, and de-facto best practices for designing, developing, and evaluating adaptive training systems. GIFT provides instructors with a suite of web-based tools for rapidly creating intelligent tutors, and it is linked to several ongoing research efforts to devise methods for automating key elements of the adaptive training authoring process. New requirements set forth by the Synthetic Training Environment (STE) have led to recent enhancements in GIFT to better support collective training events. These enhancements include an instructor dashboard, referred to as the GIFT Game Master Interface that facilitates a “human in the loop” AIS interaction model for assessing performance and injecting scenario adaptations during collective simulation-based training events (Figure 1). The new interface enables observer controllers to visualize unit progress through a live map view; monitor completed tasks, active tasks, and upcoming tasks; automatically assess individual and team performance, as well as provide manual performance assessments; and inject scenario adaptations to shape unfolding simulation-based training scenarios at run-time.

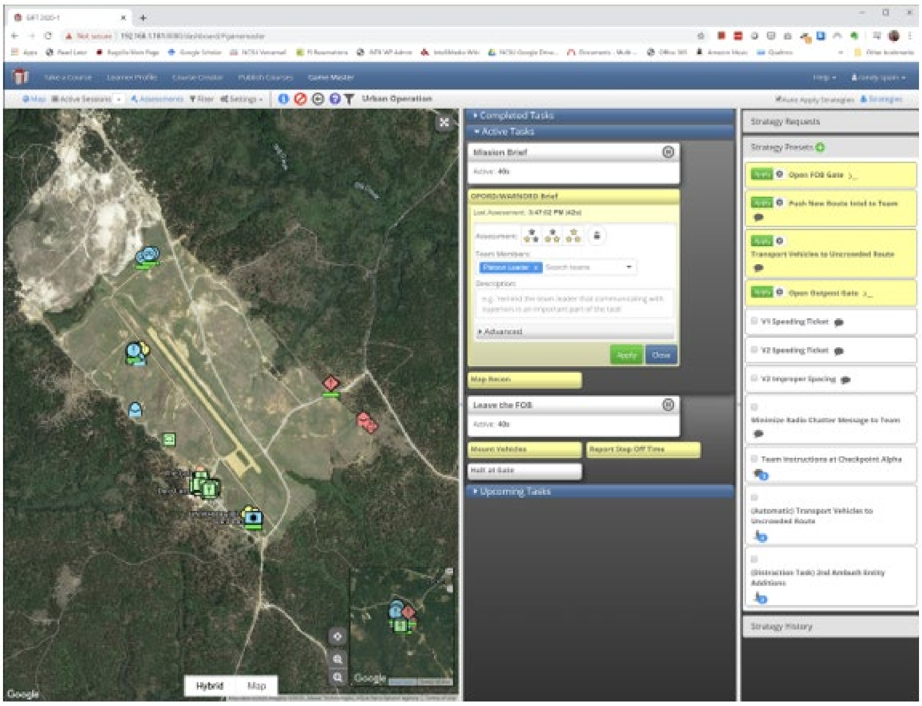


Figure 1. GIFT Game Master Interface

An important feature of GIFT Game Master is the ability to automatically deliver feedback and coaching during simulation-based training exercises. GIFT Game Master enables observer controllers to select pre-configured feedback messages, tailor messages as conditions require, and select to whom feedback and coaching should be delivered (e.g., individuals, entire team). The Game Master Interface provides a flexible, information-rich dashboard for instructors to monitor and control training exercises at run-time along with automated assessment and coaching capabilities. The logic driving assessment, scenario adaptation, and feedback functionalities is specified using domain knowledge files (DKF), which are manually authored using GUI-based tools in GIFT's web interface. In our project, we seek to enhance this adaptive team coaching model by providing methods and tools to induce data-driven models of team coaching and feedback that augment human-in-the-loop adaptive instruction enabled by GIFT GameMaster. This requires integration of data-driven tutorial planning capabilities within GIFT that are specifically targeted at addressing the requirements of collective training tasks.

INTEGRATING DATA-DRIVEN MODELS OF TEAM COACHING WITH GIFT

Integrating data-driven models of team feedback and coaching with GIFT shows significant promise for enhancing the effectiveness of adaptive training tools for a range of military tasks. There are several key requirements for the successful creation of data-driven models of team coaching. In this section, we describe these requirements and considerations in addressing them.

Data-Driven Tutorial Planning within the Learning Effect Model

A critical feature of the GIFT architecture is its modular design. GIFT's architecture is based upon the Learning Effect Model, which characterizes relationships between ITS components (e.g., learner model, pedagogical model, domain model) to enable enhanced ITS generalizability and reuse (Sottilare, Ragusa, Hoffman, & Goldberg, 2013; Sottilare et al., 2018). A vital feature of the Learning Effect Model is its distinction between *instructional strategies* and *instructional tactics*. This distinction informs how GIFT routes information to drive pedagogical decisions during adaptive instruction, and it corresponds to GIFT's separation of the Pedagogical Module and Domain Module within its software architecture. The Pedagogical Module is responsible for the selection of instructional strategies, which are domain independent and informed by the learner model (via real-time learner data, learner states, and learner attributes). The Domain Module is responsible for the selection of instructional tactics, which translate instructional strategies into pedagogical interventions that are specific to the domain and environmental conditions. Decisions about instructional tactics leverage a range of domain-specific information to guide selection of feedback and coaching. The Learning Effect Model shapes how data flows and pedagogical decisions are organized for both single learner and team-based adaptive pedagogy in GIFT.

To date, much of the work on data-driven tutorial planning has been agnostic to distinctions between instructional strategies and instructional tactics. For example, RL-based tutorial planners typically draw upon a broad range of information sources to devise state representations, action sets, and reward models which are used to automatically induce policies for adaptive pedagogy (Azizsoltani et al., 2019; Doroudi, et al., 2019; Wang et al., 2018; Williams et al., 2016). Previous work has explored several approaches for devising state representations, including manual and automated feature selection techniques (Mitchell, Boyer, & Lester, 2013; Shen & Chi, 2016). Reward models have focused on student learning outcomes (Wang et al., 2018), task efficiency (Ausin et al., 2019), and engagement-related constructs (Sawyer et al., 2017). Models of pedagogical actions have spanned instructional micro-tactics (Ausin et al., 2019) to concept sequencing strategies (Mandel et al., 2014) and remediation decisions (Spain et al., 2019).

To integrate data-driven models of team coaching with GIFT, it will be necessary to increase the congruence between the Learning Effect Model and methodologies for data-driven tutorial planning. Enhanced guidance is needed about which types of pedagogical decisions are the responsibility of the Pedagogical Module, which are the responsibility of the Domain Module, and which types of information are available (and not available) to drive state representations and rewards for inducing instructional policies. There are many unexplored questions at this intersection. For example, if an adaptive remediation policy benefits from access to features derived from domain-specific information, does that imply that remediation policies should be treated as tactical decisions within the Domain Module? Or should the remediation policy's dependence on domain-specific features be eliminated, thereby treating remediation as an instructional strategy decision within the Pedagogical module? Or is a mixed, hierarchical approach most appropriate?

Similarly, it will be essential to establish linkages between RL-based coaching and macro-adaptive and micro-adaptive pedagogical decision-making features in GIFT. Currently, GIFT supports a small number of pedagogical models, including a macro-adaptive course flow model based upon Merrill's Component Display Theory (1983) and a micro-adaptive remediation model based upon Chi's ICAP framework (2009). Devising enhanced tools and practices for linking externally developed RL-based coaching models to both macro-adaptive and micro-adaptive pedagogical decision-making features in GIFT is a key opportunity.

It will also be necessary to devise specifications for encoding pedagogical policies—this includes policies for selecting instructional strategies as well as instructional tactics—that can be read, parsed, and executed by GIFT. In the short-term, data-driven policies for team coaching and feedback will need to be induced using external machine learning tools. In the longer-term, machine learning workflows will need to be integrated with GIFT to realize the vision of self-improving instructional systems (Sinatra et al., 2019). A related requirement is the need to create a configurable data pipeline for processing run-time information to compute state representations and rewards for driving models of team coaching and feedback. This is analogous to the data pipeline that is used to process sensor data in GIFT. The pipeline accumulates raw sensor data in a buffer over a fixed time window, provides the data to a user-defined script for transformation into a feature vector representation, and uses the resulting vector(s) as input to external machine learning-based models that drive adaptive behavior within GIFT.

In summary, devising a combination of best practices and software tools to support the integration of data-driven instructional policies with GIFT will be critical for enabling data-driven team coaching capabilities that generalize across collective training tasks and simulation environments.

Exploration-Exploitation Tradeoff in Data-Driven Models of Team Feedback

A critical issue in reinforcement learning is the tradeoff between exploration and exploitation (Sutton & Barto, 2018). Online reinforcement learning involves a process of trial-and-error that balances between selecting actions based upon currently known information (i.e., exploitation) versus selecting actions to find new information and lead to better decisions in the future (i.e., exploration). This tradeoff implies that the learning agent may occasionally select actions that are sub-optimal; the agent engages in experimentation in the hope of discovering a better policy to guide its future behavior. Without experimentation, the learning agent would fail to improve its policy or increase its reward. In the context of team training, it is imperative for data-driven tutorial planners to make decisions that balance following the best policy known thus far and selecting actions that could yield new, better policies for the future.

Given that data-driven tutorial planners must, by nature, embrace experimentation, it is critical that they also be designed to minimize potential detrimental effects on learners. This is important in traditional academic subjects (e.g., math, science) where RL-based tutorial planning has been examined for years, but it has particular salience in military domains where learners are trained to acquire skills with potential life-and-death consequences. Exploratory RL-based tutorial policies are often designed to be “random yet reasonable,” implying that a random pedagogical policy would still produce acceptable training outcomes by restricting the space of possible pedagogical actions available to the planner (Ausin et al., 2019). This requires curation by the system designer to ensure that the chosen instructional strategies (and/or tactics) are broadly positioned to benefit learners across a range of circumstances, and at worst, will do no harm.

In data-driven tutorial planning, the exploration vs. exploitation tradeoff also intersects with questions about how team feedback and coaching are delivered to learners during collective training exercises. Two approaches for delivering team coaching are often distinguished: (1) automated feedback provided directly to Soldiers through a simulation- or technology-based interface, and (2) feedback suggestions provided to a human instructor who will provide instruction and assessment him/herself. In the latter case, a human instructor maintains control over what pedagogical strategies and tactics are delivered; he/she can “overrule” suggestions provided by the tutorial planner based upon their own knowledge, experience, and preferences. This raises questions about how to sample the space of possible pedagogical policies when the exploration-exploitation tradeoff is mediated by expert instructors’ pedagogical decisions; exploration could be biased by the instructional preferences of human trainers. Devising methods to collect sufficient data to induce effective pedagogical models, while also fitting within the instructional workflows of collective training in the Army, is a promising area for investigation.

Enhancing GIFT’s Event Reporting Tool for Data-Driven Team Feedback Workflows

GIFT’s Event Reporting Tool (ERT) enables researchers, instructors, and developers to extract data from learner interactions with GIFT collected in research studies and course deployments. The ERT provides tools for filtering and formatting GIFT log and survey data for offline analysis and processing. The ERT is critical to GIFT’s utility as a research and experimentation platform (Sinatra, 2016). Notably, there are promising opportunities to enhance the ERT to streamline workflows for the creation of data-driven tutorial planning models, which would substantially accelerate progress on automated team coaching and feedback.

GIFT generates a wealth of fine-grained log data on learner interactions with adaptive courses created with the GIFT Course Creator, but only a small subset of the data is relevant to the creation of data-driven models of adaptive coaching and feedback. For example, learner survey responses, tutor replies, and practice outcomes are relevant, but logs of internal message passing between GIFT modules are not. Providing streamlined options for filtering relevant data and message types within the ERT would significantly improve workflows for creating data-driven tutorial policies in GIFT. In addition, developing capabilities to automatically extract common domain-specific and domain-independent features (e.g., pre-test score, proportion of course completed, count of past tutor replies) from interaction logs would help to jump start the creation of data-driven feedback models with data collected using GIFT. Currently, custom data analysis scripts must be created to process and transform ERT-generated data files into data representations that are conducive to analysis with machine learning tools. In addition to extracting common features, providing embedded tools to facilitate automated extraction of custom features, including features specific to team-based training tasks, would yield significant benefits. A corollary to this is the importance of distinguishing features available to the Pedagogical Module (based on information from the learner model) and features that can only be computed within the Domain Module (based on information from the instructional strategies and environmental conditions); highlighting these distinctions is likely to further help facilitate the integration of data-driven team feedback policies into GIFT for run-time use. Finally, extending the ERT to generate formatted reports that can be automatically read and parsed by open-source machine learning tools (e.g., scikit-learn, Keras, OpenAI Gym) is likely to further accelerate the time required to induce data-driven models of team feedback and coaching.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

Delivering tailored coaching and feedback to team members in synthetic training environments is a critical challenge. We have introduced a research collaboration between North Carolina State University, Intelligent Automation, Inc., and the U.S. Army Combat Capabilities Development Command Soldier Center to investigate the design, development, and evaluation of a generalized data-driven framework for team-centered tutorial planning with GIFT. We seek to leverage recent advances in machine learning, and reinforcement learning in particular, to devise pedagogical planning models that automatically deliver run-time feedback during team training tasks in simulated training environments. Integrating data-driven models of team coaching and feedback with GIFT introduces several requirements. These include aligning data-driven tutorial planning methodologies with the Learning Effect Model, accounting for the exploration-exploitation tradeoff inherent in data-driven tutorial planning frameworks, and extending GIFT features, including the Event Reporting Tool, to streamline development of data-driven team coaching functionalities using reinforcement learning techniques. These requirements also point toward future research directions centered on the creation of reinforcement learning-based tutoring architectures that generalize across individual and team tasks, military training domains, and simulation-based training environments. Developing capacity to devise data-driven team feedback policies contributes toward addressing the Army's modernization priorities associated with the Synthetic Training Environment (STE) and also serves to extend GIFT by adding support for modeling tutorial planning at the team level.

ACKNOWLEDGEMENTS

The research described herein has been sponsored by the U.S. Army Research Laboratory under cooperative agreement W912CG-19-2-0001. The statements and opinions expressed in this article do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

REFERENCES

Ausin, M. S., Azizsoltani, H., Barnes, T., & Chi, M. (2019). Leveraging deep reinforcement learning for pedagogical policy induction in an intelligent tutoring system. In *Proceedings of the 12th International Conference on Educational Data Mining,* pp. 168–177.

Azizsoltani, H., Kim, Y. J., Ausin, M. S., Barnes, T., & Chi, M. (2019). Unobserved is not equal to non-existent: Using Gaussian processes to infer immediate rewards across contexts. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 1974–1980.

Billings, D. R. (2012). Efficacy of adaptive feedback strategies in simulation-based training. *Military Psychology, 24*, 114–133.

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

Chi, M., VanLehn, K., Litman, D., & Jordan, P. (2011). 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.

Doroudi, S., Aleven, V., & Brunskill, E. (2017). Robust evaluation matrix: Towards a more principled offline exploration of instructional policies. In *Proceedings of the 4th ACM Conference on Learning@Scale*, pp. 3–12.

Doroudi, S., Aleven, V., & Brunskill, E. (2019). Where's the Reward? *International Journal of Artificial Intelligence in Education, 29*(4), 568–620.

Goldberg, B., Nye, B., Lane, H. C., & Guadagnoli, M. (2018). Team assessment and pedagogy as informed by sports coaching and assessment. In R. A. Sottilare, A. C. Graesser, X. hu, & A. M. Sinatra (Eds.), *Design Recommendations for Intelligent Tutoring Systems, Volume 6: Team Tutoring*. (pp 105–199). U.S. Army Research Laboratory. Orlando: FL.

Johnson, C. I., & Priest, H. A. (2014). The feedback principle in multimedia learning. *The Cambridge Handbook of Multimedia Learning*, 449.

Kozlowski, S. W., Watola, D. J., Jensen, J. M., Kim, B. H., & Botero, I. C. (2009). Developing adaptive teams: A theory of dynamic team leadership. In E. Salas, G. F. Goodwin & C. S. Burke, (Eds.), *Team Effectiveness in Complex*

Organizations: Cross-Disciplinary Perspectives and Approaches (pp 113–155). Psychological Press: New York, NY.

Kulhavy, R. W., & Stock, W. A. (1989). Feedback in written instruction: The place of response certitude. *Educational Psychology Review*, *1*(4), 279-308.

Mandel, T., Liu, Y. E., Levine, S., Brunskill, E., & Popovic, Z. (2014). Offline policy evaluation across representations with applications to educational games. In *Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 1077-1084.

Merrill, M.D. (1983). Component display theory. In C. Reigeluth (ed.), *Instructional Design Theories and Models*. Hillsdale, NJ: Erlbaum Associates.

Mitchell, C., Boyer, K., & Lester, J. (2013). Evaluating state representations for reinforcement learning of turn-taking policies in tutorial dialogue. In *Proceedings of the 14th Annual SIGDIAL Meeting on Discourse and Dialogue*, pp. 339-343.

Rowe, J., & Lester, J. (2015). Improving student problem solving in narrative-centered learning environments: A modular reinforcement learning framework. In *Proceedings of the 17th International Conference on Artificial Intelligence in Education*, pp. 419-428.

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

Serge, S. R., Priest, H. A., Durlach, P. J., & Johnson, C. I. (2013). The effects of static and adaptive performance feedback in game-based training. *Computers in Human Behavior, 29*, 1150–1158.

Shen, S., & Chi, M. (2016). Aim low: Correlation-Based feature selection for model-based reinforcement learning. In *Proceedings of the 9th International Conference on Educational Data Mining*, pp. 507–512.

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.

Sinatra, A. M. (2016). The updated research psychologist's guide to GIFT. *Proceedings of the 4th Annual GIFT User Symposium* (GIFTSym4), pp. 135–144.

Sinatra, A.M., Graesser, A.C., Hu, X., Brawner, K., & Rus, V. (Eds.). (2019). *Design Recommendations for Intelligent Tutoring Systems: Volume 7 - Self-Improving Systems*. Orlando, FL: US Army CCDC. Available at: https://gifttutoring.org/documents

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*, 225–264.

Sottilare, R., Graesser, A., Hu, X. , & Sinatra A. (Eds.). (2018). *Design Recommendations for Intelligent Tutoring Systems: Volume 6 - Team Tutoring*. Orlando, FL: U.S. Army Research Laboratory. ISBN 978-0-9977257-4-2. Available at: https://gifttutoring.org/documents

Sottilare R., Ragusa C., Hoffman, M. & Goldberg B. (2013). Characterizing an adaptive tutoring learning effect chain for individual and team tutoring. In *Proceedings of the Interservice/Industry Training Simulation and Education Conference*; Orlando, FL.

Spain, R., Rowe, J., Goldberg, B., Pokorny, R., Mott, B., & Lester, J. (2019). Towards data-driven tutorial planning for counterinsurgency training in GIFT: Preliminary findings and lessons learned. In *Proceedings of the 7th Annual GIFT Users Symposium*, pp. 111–120.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press. Cambridge: MA.

Swart, E. K., Nielen, T. M., & de Jong, M. T. S. (2019). Supporting learning from text: A meta-analysis on the timing and content of effective feedback. *Educational Research Review,* 100296.

Taylor, M. E. (2018). Improving reinforcement learning with human input. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 5724-5728.

TRADOC (2017). *The U.S. Army Learning Concept for Training and Education: 2020-2040.* Retrieved from: https://adminpubs.tradoc.army.mil/pamphlets/TP525-8-2.pdf

Wang, P., Rowe, J. P., Min, W., Mott, B. W., & Lester, J. C. (2018). High-fidelity simulated players for interactive narrative planning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence,* pp. 3884-3890.

Williams, J. J., Kim, J., Rafferty, A., Maldonado, S., Gajos, K. Z., Lasecki, W. S., & Heffernan, N. (2016). Axis: Generating explanations at scale with learner sourcing and machine learning. In *Proceedings of the 3rd ACM Conference on Learning@ Scale* (pp. 379-388).

Wisniewski, B., Zierer, K., & Hattie, J. (2020). The power of feedback revisited: A meta-analysis of educational feedback research. *Frontiers in Psychology, 10*, 3087.

Wouters, P., & van Oostendorp, H. (2013). A meta-analytic review of the role of instructional support in game-based learning. *Computers & Education, 60*, 412–425.

ABOUT THE AUTHORS

***Dr. Jonathan Rowe** is a Research Scientist in the Center for Educational Informatics at North Carolina State University, as well as an Adjunct Assistant Professor in the Department of Computer Science. He received the PhD and MS degrees in Computer Science from North Carolina State University, and a BS 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, multimodal learning analytics, learner modeling, and computational models of interactive narrative generation. Dr. Rowe also serves on the editorial boards of the International Journal of Artificial Intelligence in Education and IEEE Transactions on Learning Technologies.*

***Dr. Randall Spain** is a Research Psychologist in the Center for Educational Informatics at North Carolina State University where he uses principles, theories, and methods of applied psychology to design and evaluate the impact of advanced training technologies on learning and performance. Dr. Spain holds a PhD in Human Factors Psychology and serves on the editorial board for Military Psychology.*

***Dr. Benjamin Goldberg** is a senior researcher in the Learning in Intelligent Tutoring Environments (LITE) Lab at the Combat Capabilities Development Command (CCDC) Soldier Center, Simulation and Training Technology Center (STTC) in Orlando, FL. He has been conducting research in modeling and simulation for the past eight years with a focus on adaptive learning and how to leverage artificial intelligence tools and methods for adaptive computer-based instruction. Currently, he is the LITE Lab's lead scientist on instructional strategy research within adaptive training environments. Dr. Goldberg holds a PhD from the University of Central Florida in Modeling & Simulation.*

***Dr. Robert Pokorny** is Principal of the Education and Training Technologies Division at Intelligent Automation, Inc. He earned his PhD in Experimental Psychology at the University of Oregon in 1985, and completed a postdoctoral appointment at the University of Texas at Austin in Artificial Intelligence. His first position after completing graduate school was at the Air Force Research Laboratory, where he developed methodologies to efficiently create intelligent tutoring systems for a wide variety of Air Force jobs. At Intelligent Automation, Bob has led many cognitive science projects, including adaptive visualization training for equipment maintainers, and an expert system approach for scoring trainee performance in complex simulations.*

***Dr. Bradford Mott** is a Senior Research Scientist in the Center for Educational Informatics at North Carolina State University. Prior to joining North Carolina State University, he served as Technical Director at Emergent Game Technologies where he created cross-platform middleware solutions for video game development, including solutions for the PlayStation 3, Wii, and Xbox 360. Dr. Mott received his Ph.D. in Computer Science from North Carolina State University in 2006, where his research focused on intelligent game-based learning environments. His current research interests include computer games, computational models of interactive narrative, and intelligent game-based learning environments.*

***Dr. James Lester** is Distinguished University Professor of Computer Science at North Carolina State University, where he is Director of the Center for Educational Informatics. His research centers on transforming education with technology-rich learning environments. With a focus on adaptive learning technologies, his research spans intelligent tutoring systems, game-based learning environments, affective computing, and tutorial dialogue. The adaptive learning environments he and his colleagues develop have been used by thousands of students in K-12 classrooms. He received his PhD in Computer Science from the University of Texas at Austin in 1994. He is a Fellow of the Association for the Advancement of Artificial Intelligence (AAAI).*

Team Communication Analytics Using Automated Speech Recognition

Randall Spain[1], Wookhee Min[1], Jason Saville[1], Bradford Mott[1], Keith Brawner[2], Joan Johnston[2], Gregory Goodwin[2], and James Lester[1]
North Carolina State University[1], U.S. Army Combat Capabilities Development Command Soldier Center, STTC[2]

INTRODUCTION

Automated analysis of team discourse is critical for meeting the Army's vision of providing adaptive coaching and feedback to squads and collective units in synthetic training environments (STE). The communication between team members can provide insightful information about team coordination, cooperation, and shared cognitive states. Automatically analyzing team communication can therefore provide deep insight into team processes and teamwork behaviors that impact team effectiveness. To meet this need, the U.S. Army Combat Capabilities Development Command Soldier Center, Simulation and Training Technology Center, and North Carolina State University are developing a deep learning-driven natural language processing (NLP) framework that aims to automatically analyze team communication data, parse it into classifications schemes, and provide summary statistics of critical team communication features that can be used to analyze and identify antecedents of team performance. By analyzing team discourse during training episodes, the framework will enable the assessment of team communication content, quality, and information exchange features, and provide insights into team processes that could be used to inform team assessment and feedback policies in adaptive instructional systems.

Automatic speech recognition (ASR) serves as a key component in the NLP-driven analytics pipeline for assessing team communication. ASR provides textual representations of spoken utterances between team members. An accurate transcription of team members' spoken communication is paramount for automating team communication analytics. Prior work on ASR is grounded in hidden Markov models and Gaussian mixture models designed to model the temporal dynamics of speech and predict textual representations by determining fitness between the hidden states and the acoustic input (e.g., Hinton et al., 2012; Povey et al., 2011). More recently, researchers in both industry and academia have investigated deep learning, a family of machine learning techniques based on deep neural networks, for acoustic modeling in ASR (Chiu et al., 2018; Yu & Deng, 2016). Common tools that support deep neural network-based ASR include Google Cloud Speech-to-Text, Microsoft Azure Speech to Text, IBM Watson Speech, Apple Speech framework, and Amazon Transcribe, as well as Kaldi, an open-source speech recognition toolkit (Cheng, Povey, Huang, Xu, Khudanpur, & Yan, 2018; Ravanelli, Parcollet, & Bengio, 2019). These tools provide real-time speech recognition capabilities that automatically translate spoken language into text, which can then be further investigated for a variety of natural language processing tasks including syntactic, semantic, and dialogue analyses.

In this paper, we investigate speech recognition performance of several publicly available ASR engines using two team communication datasets. The first dataset includes dialogue acts and information exchange sequences captured during a series of key leader engagements from the Squad Overmatch (SOvM) research program (Johnston, 2018). The second dataset includes dialogue captured between squad members and their platoon leader from an after-action review session. Challenges presented by both datasets include performing accurate ASR, identifying speakers in a multi-party dialogue situation, and capturing domain-specific keywords. Among a range of deep neural network-based ASR engines available for this purpose, we investigate Google Cloud's and Microsoft Azure's Speech-to-Text services as well as the Kaldi open-source toolkit. Google's Cloud Speech-to-Text engine allows speech recognition both offline (i.e., pre-recorded audio data) and online (i.e., live audio captured using a microphone) based on a pre-trained transcription model of choice (e.g., video, phone call, voice command). Its analysis takes into account contextual phrases and keywords to improve speech recognition, reports confidence scores of predictions at the sentence level, and identifies speakers in a multi-speaker conversations. Similar to the Google Cloud ASR service, Microsoft Azure's Speech-to-Text service offers a suite of functionalities such as generating timestamped transcripts and multi-speaker diarization along with software development kit (SDK) support for various programming languages (e.g., C#, C++, Java, Python). Although both Google Cloud's and Microsoft Azure's ASR services demonstrate considerable speech-to-text transcription performance as well as provide a wide range of ASR functionalities relative to other off-the-shelf software, they often have significant limitations such as data privacy (i.e., audio data passed to the cloud servers), paid service, and network dependency (i.e., online service) that should be considered when selecting an ASR engine. Kaldi, an open-source speech recognition and acoustic feature extraction toolkit (Povey et al., 2011), effectively addresses the challenges found in various cloud-based services, but has shortcomings with respect to impoverished performance in handling various types of noise in audio data (e.g., poor audio recording quality, background noise) and slow speech recognition response time (Kimura, Nose, Hirooka, Chiba, & Ito, 2018). To better understand the advantages and disadvantages of these systems, we evaluate the word error rates (WER) and keyword spotting rates of each ASR engine. In addition, the paper presents preliminary results of transcript omission rates generated by each software system. The paper also contains a discussion of the quality and structure of the spoken team communication data and identifies challenges (e.g., quality issues due to environmental noise, speaker diarization challenges, ASR performance issues for a domain-specific dataset) and directions for future research. It concludes with a discussion of how automated verbal team communication analysis could benefit GIFT for supporting team training.

BACKGROUND

Investigating individuals' communication during team training exercises can provide insight into the rich processes underlying team performance and effectiveness (Marlow, Lacerenza, Paoletti, Burke, & Salas, 2018; Smith-Jentsch, Johnston, & Payne, 1998). For instance, dialogue among team members can be used to investigate how well they collaborate, coordinate, and engage in supportive behaviors. Despite the affordances that team dialogue offers for understanding team processes, analyzing team communication has historically been extraordinarily resource-intensive for the team training research community. This is because research team members typically must manually transcribe recordings of team dialogue. Recent advancements in artificial intelligence show significant promise for addressing these challenges by utilizing ASR models that can automatically recognize human speech and create transcripts of team dialogue, thereby reducing the level of human intervention needed to transcribe team communication recordings. Many companies, including Google, IBM, Apple, and Microsoft, offer speech recognition services that can provide real-time transcriptions of audio input using online and offline processes. These systems are built on universal libraries of speech data and use deep learning techniques to accurately predict textual representations and word sequences from audio data.

Despite advancements in ASR capabilities, there are a number of factors that can impact speech recognition accuracy. ASR accuracy can be significantly degraded in noisy and in multi-party, multi-dialogue settings such as classrooms and training environments where a speaker's voice may share similar frequency and temporal characteristics with other speakers, or where loud noises can mask what a person is saying (Chao, Chan, & Lane, 2019). Speaker related acoustic variability is a major source of errors in ASR accuracy (Serizel & Giuliani, 2017). Further, the data ASR engines are trained on and their model architectures can impact speech recognition performance (Aleksic et al., 2015; Ravanelli et al., 2019). This paper aims to identify which contemporary ASR systems are best suited for analyzing team member communication from collective training events that are characterized by multiple speakers and varying degrees of environmental noise.

Previous research examining the performance of ASR systems utilizing data from naturalistic settings shows that the accuracy of speech recognition software can vary. For instance, Blanchard et al. (2015) evaluated five ASR engines to see how well they could transcribe questions posed by teachers in a noisy classroom environment. The speech data included 530 questions captured from three middle grade teachers during classroom interactions. Transcripts were produced by Google Speech, Bing Speech, AT&T Watson, Microsoft Speech SDK 5.1, and two variants of Sphinx 4 and compared to human generated transcriptions. Results showed Google Speech and Bing Speech performed the best among the five providers, reaching word accuracy levels of 56% and 52% (word error rates of 44% and 48%), respectively. Further results showed that Bing Speech's word accuracy surpassed Google's after the research team eliminated long teacher pauses from the audio clips. To further validate the reliability of Bing's ASR accuracy, the team conducted a follow-up evaluation using a large number of speakers. Data included 3,057 dialog turns recorded in a laboratory setting by 28 participants. Not surprisingly, the average word accuracy rates were considerably higher for the lab-based speech recordings (60%) compared to the recording from the noisy classroom environment (52%). The authors concluded that while the technologies were far from perfect, they provided value for automatically transcribing speech from classrooms and other noisy environments.

Kim et al. (2019) conducted a similar evaluation of contemporary ASR engines, evaluating the performance of Google Cloud, IBM Watson, Microsoft Azure, Trint, and YouTube using video recordings of patient interviews between 12 medical students and two simulated patients. A total of 24 interviews and 28,840 words were analyzed and compared to manual transcriptions to identify the best performing ASR service. Results suggested that YouTube offered the most accurate speech recognition relative to the other providers with word error rates (WER) of approximately 28%, followed by Google Cloud (WER: 35%) and Microsoft Azure (WER: 40%). IBM Watson performed the poorest with WER of 50%. One notable difference between this study and the Blanchard et al., (2015) evaluation is that the recordings were captured from an online video conferencing tool which contained no background noise and likely contributed to higher speech recognition accuracy compared to analyzing recordings from noisy classroom settings.

More recently, Georgila, Keuski, Yanov, and Traum (2020) evaluated the accuracy of speech recognition engines from Amazon, Apple, Google, IBM, Microsoft, and Kaldi using data of human interactions with virtual characters in different settings. The virtual character technologies were characterized by question-answer systems designed to address questions about a specific domain or topic as well as virtual characters designed to train tactical questioning and cross-cultural negotiation skills. The dataset also included human-to-human utterances captured from a virtual reality-based application designed to support military call-for-fire and call-for-air-support training missions. The accuracy of each ASR engine was compared to human transcripts using online and offline processes. Results showed that Google Cloud's online speech recognition system performed the best across datasets. The WER for human interactions with the virtual character technologies ranged from 8% to 18%. In contrast, accuracy for the virtual reality training data, which included speech between humans, was poorer with WER reaching levels of 35% for the best performing software and 80% for the worst performing software. The Kaldi LibriSpeech model performed the poorest across domain and these results were attributed to the fact the model was trained on data using audio books rather than conversation speech like the systems from Google, Apple, Amazon, and Microsoft. The authors concluded that despite much progress in ASR technology, current state-of-the-art speech recognition engines still perform suboptimally in noisy and specialized domains where speakers use domain-specific vocabulary and language.

The present study extends this line of research and compares the accuracy of several state-of-the-art ASR services, including commercial and research software, using data from a live training event and an after-action review session in order to evaluate their accuracy for supporting team communication analytics.

METHODOLOGY

Team Speech Recordings

Two datasets were used to evaluate ASR system accuracy. The first dataset included audio recordings of a live training exercise from the SOvM project (Johnston, 2018, Johnston et al., 2019). Specifically, we analyzed audio data captured from the squad leader in the final mission (M3) as he interacted with team members, role players, and virtual characters that were part of the capstone training event. The training event incorporated key events that required squad members to demonstrate advanced situational awareness, resilience, and teamwork skills. Squad tasks for the mission included: conducting a key leader engagement, interacting with hostile actors as they moved through the village, responding to a simulated IED explosion, responding to sniper fire on civilians and participants, and conducting tactical combat casualty care. A total of 204 statements were analyzed, which included utterances from the squad leader and role players from the training exercise that were captured on the squad leader's microphone. These statements were captured from the first 20 minutes of the training event in which the squad leader moved with his unit though the village, conducted a key leader engagement, and interacted with other key role players in the scenario.

The second dataset included team communication from an after-action review (AAR) session. These data were also part of the SOvM project. Interactions were captured between the platoon leader, squad leader, team leaders, and additional team members. Spoken data from subject matter experts who participated in the AAR session were also captured and analyzed. The AAR session took place in a small lecture hall and the session was recorded with a single video recorder with a built-in microphone that was positioned in the front of the meeting room. A total of 320 statements were analyzed from 35 minutes of audio data.

Manual Transcripts

Manual transcripts of the AAR recordings were created and independently verified by two members of the research team. Manual transcripts of the live training event were obtained from Senior Researchers at the U.S. Army Combat Capabilities Development Command Soldier Center, Simulation and Training Technology and were verified independently by members of the research team.

ASR Systems

In this work, we investigated three ASR systems: Google Cloud Speech-to-Text, Microsoft Azure Speech to Text, and Kaldi. The selection of the two commercial ASR engines was informed by previous research showing more accurate speech recognition compared to other competitive ASR toolkits as well as distinctive benefits offered by toolkits for analyzing speech data. Google Cloud Speech-to-Text is a cloud-based ASR service that uses deep neural network-based speech recognition models. It provides a software development kit (SDK) along with programming language-specific client libraries including C#, Java, and Python to use the Speech-to-Text API for analyzing either real-time streaming or prerecorded audio. A distinctive feature offered by the Google ASR engine is the ability to select pre-trained transcription models optimized for different audio sources, including a *default* model trained for long-form audio recordings that feature a single speaker, a *video* model that is trained for transcribing audio from a video with multiple speakers, and a *phone* model that is best for audio data originated from phone calls. The Google ASR service offers additional functionalities such as speaker diarization (i.e., automatic predictions about which of the speakers in a conversation spoke each utterance), keyword-enhanced transcriptions, and a time-stamped transcription of each utterance. Our preliminary analysis on one dataset suggested that the *video* speech recognition model outperforms the *default* model due to the nature of the dataset (e.g., audio extracted from video, multi-speaker). Therefore, the video model was used for all analyses in this paper. We also compared the performance of the model with and without a set of domain-specific keywords (names, acronyms, mission specific terms and phrases) that were developed to improve transcription accuracy. The number of keywords defined for both the live training and AAR sessions examined in this work was 84.

Microsoft Azure's Speech-to-Text, which is a sub-package of the Microsoft Azure Cognitive Services, was also examined in this work. Similar to Google Cloud Speech-to-Text, it provides a programming language-specific Speech SDK that enables a variety of speech recognition functionalities such as transcription of multi-user conversation both online and offline, speaker diarization, and time-stamped transcriptions. In addition, Microsoft Azure's Speech-to-Text allows users to perform speech recognition with speech service containers without sending private data to the cloud. It also allows users to train customized transcription models utilizing a pair of audio and ground-truth transcripts or related text only. This model customization feature holds significant potential since domain-specific jargon and phrases may not be sufficiently captured in a transcription model trained with data in other contexts. Here again, we evaluated the performance of the default ASR model with and without a set of domain-specific keywords that were developed to improve model accuracy and keyword spotting rates.

Lastly, we investigated the Kaldi speech recognition toolkit (Povey et al., 2011). Unlike the cloud-based ASR services, Kaldi is an open-source speech recognition toolkit implemented in C++ and thus researchers can perform both online and offline speech recognition as well as acoustic feature extraction on a local machine. In this work, we utilized a Kaldi version incorporating a pre-trained deep neural network-based acoustic model. Specifically, this model was trained with mel-frequency cepstral coefficients (MFCC) acoustic features and the TEDLIUM speech corpus featuring 118 hour-long English-language TED talks (Rousseau, Deléglise, & Esteve, 2012).

RESULTS

ASR performance was evaluated using word error rate (WER), a common metric of the performance of speech recognition systems. WER was calculated by dividing the total number of insertions, deletions, and substitutions used to correct the hypothesis transcript by the total number of words in the reference transcript (Soukoreff & MacKenzie, 2001). In addition, we also report keyword spotting rates (KSR), since accurately detecting important keywords is a key to the success of downstream natural language processing tasks for team communication analysis, especially in case of dealing with noisy speech data. Finally, we report transcript omission rates (TOR) to evaluate how each ASR engine dealt with noisy speech segments between simply giving up on the recognition job or generating outputs using the best guess of the recognition engine. This metric returns the number of missing transcript lines divided by the total number of lines in the human transcript. From a performance perspective, lower WER and TOR, and higher KSR suggest better ASR performance.

Table 1 summarizes the WER results, showing differences in performance between the baseline model for each ASR engine and baseline models enhanced with keywords. Results show that, overall, Google Cloud's speech recognition (28.3%) outperformed Microsoft Azure and Kaldi on the AAR data, while Microsoft Azure Speech to Text (59.8%) outperformed Google Cloud's speech recognition when examining speech data from the live training exercise. The Kaldi speech recognition system yielded a number of omission errors in the transcript for the live training audio data which was significantly noisier than the AAR audio data, therefore only data for the AAR data are reported. Interestingly, speech recognition accuracy did not significantly improve after adding keywords to the model.

Table 1. Word Error Rate (WER) Results (%).

ASR	Model	Live Training	AAR
Google Cloud	video (neural)	70.42	**28.32**
Google Cloud + Keywords	video (neural)	69.68	28.40
Microsoft Azure	default (neural)	**59.84**	35.11
Microsoft Azure + Keywords	default (neural)	64.07	35.90
Kaldi	neural	N/A	74.81

Table 2 reports KSR and TOR results. The KSR results show that Google Cloud's ASR enhanced with a set of keywords achieved the highest keyword spotting rates (53.6%) for the AAR data, followed by Google Cloud ASR without keyword input (47.8%). Table 2 also shows the keyword-enhanced Microsoft Azure speech recognizer attained the highest KSR (38.6%) for the live training speech data followed by the keyword-enhanced Google Cloud ASR and the Microsoft Azure that did not take into account the keyword set (tied at 27.1%). The TOR results reported in Table 2 indicate that Microsoft Azure Speech to Text achieved the lowest omission rates for both the live training and AAR data (35.8% and 12.2%, respectively), which suggests that Microsoft Azure adopted a more liberal policy for generating transcripts, particularly for noisy data, compared to Google Cloud Speech-to-Text and Kaldi. Table 3 shows errors generated by each ASR engine for the live training and AAR data.

Table 2. Keyword Spotting Rate (KSR) and Transcript Omission Rate (TOR) Results (%).

ASR	Live Training KSR	Live Training TOR	AAR KSR	AAR TOR
Google Cloud	20.00	64.68	47.83	13.75

Google Cloud + Keywords	27.14	64.18	**53.62**	13.13
Microsoft Azure	27.14	**35.82**	33.33	**12.19**
Microsoft Azure + Keywords	**38.57**	48.26	40.58	14.06
Kaldi	N/A	N/A	10.14	56.25

It should be noted that although the best WER results were achieved without utilizing the set of keywords in generating the transcripts, the highest keyword spotting rates were achieved when the speech recognition systems were enhanced with the keywords. These results can be partially explained by the inherent tension between a high keyword spotting rate and a high word accuracy rate. Speech recognition systems that more accurately catch domain-specific keywords often make false positive predictions on general words that are not included in the keyword set, thereby yielding lower word accuracy rates overall.

Table 3. Comparison of Example Transcripts Generated by the ASR Toolkits.

Domain	Transcript	Google Cloud	Microsoft Azure	Kaldi
Live Training	yeah let's start pushing it up we need to go over towards alpha one this time by where the dude with the mic is alright	now let's start pushing it up need to go over towards alpha 1 this time I wear the dude from my kids right huh.	Yeah, we'll start pushing up. Need to go over towards A1 this time by where the dude Mike is right.	N/A
Live Training	seems as though the vendors are arguing on route black but dispersing now hard copy over	arguing on black...	Seems other vendors arguing on route black, but this person now copy over.	N/A
AAR	squad leader talk to me what was the mission and your plan scheme and maneuver	squalor talk to me what was the mission and your plan scheme maneuver	squalor talk? Maine was the mission and your plansky maneuver.	as well it taught me was the mission and plants in your gut.

AAR	so at that point we pushed up route black and then started moving down ars golden towards alpha four correct	so that point we pushed up route black and then start moving down it's our golden towards out for correct	So at that point we pushed up route black and then start going down ASR Golden towards out. Four correct.	so that point we have pushed up route black man started down a sort of towards out for brecon.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

ASR is a critical first step towards the automatic analysis of team communication data in an NLP pipeline. Our goal was to evaluate how accurately Google Cloud's and Microsoft Azure's Speech-to-Text services, as well as Kaldi's open-source toolkit, could transcribe audio recordings of collective training events that included multiple speakers and varying ranges of environmental noise. Results showed that the Google Cloud speech recognition system outperformed Microsoft and Kaldi when transcribing speech from the AAR session, with respect to WER, and that the Microsoft Azure speech recognition system outperformed Google when transcribing speech from the live training event, while Kaldi failed to generate reliable transcriptions for the live training session.

Several conclusions can be drawn from these preliminary results. First, despite many advancements in speech recognition capabilities (Georgila et al., 2020) the transcripts produced by the ASR systems contained a number of errors, particularly for the live training session. These errors could be attributed to background noise, varying acoustic characteristics from the speakers, or a poor match between the software's language model and the target domain. WER for the AAR transcripts demonstrated substantially more reliable results, since the AAR session contained far less background noise compared to the live training event. These results highlight the challenge of obtaining accurate speech recognition for conversational tasks and the need for further research on ASR performance in settings that contain multi-party dialogue and background noise. Second, adding domain specific keywords to the ASR toolkits may have improved keyword spotting performance (Table 2), but it did not increase transcription accuracy (Table 1). This highlights the need to further investigate methods to improve transcription accuracy as well as keyword spotting performance for the target domain.

Ultimately, for natural language analytics to be integrated into AIS platforms such as GIFT to assess and diagnose team performance in real-time, ASR engines need to provide more accurate speech recognition results. A critical question is what level of accuracy should a speech recognition system achieve in order to be considered robust enough, bearing in mind that the goal of speech recognition in an NLP pipeline is not to obtain perfect transcription results, but to obtain transcripts that contain a reasonable representation of the spoken team communication language that can serve as input for natural language processing tasks further down the team communication analytics pipeline. There are several promising research avenues to explore that could ultimately be integrated into GIFT to provide real-time analytics of team communication. One direction is to extend the current work and examine ASR performance on a broader set of team communication data and investigate how techniques for providing contextual information can be used to improve transcription accuracy. Of particular interest is identifying the optimal set of weights that should be assigned to keywords to simultaneously enhance keyword spotting and induce lower word error rates. Another direction is to investigate the extent to which customized transcription models can improve ASR accuracy. Microsoft Azure and Kaldi offer methods to train customized models which could significantly improve their transcription accuracy. Kaldi also holds significant potential for creating robust personalized models, since it enables researchers to devise flexible neural acoustic model architectures utilizing the PyTorch deep-learning library (Ravanelli, et al., 2019). Future research should examine how applying background noise reduction and voice activity detection techniques (e.g., Singh, Venter, Muthu, & Brown, 2019) can improve ASR accuracy for noisy data. This will be especially important for analyzing team communication analytics in real-time during collecting training exercises that are characterized by multiple speakers, background noise, and varying levels of input quality. These future research directions will together point towards obtaining more accurate transcripts that can better facilitate downstream natural language processing tasks to support team training analytics.

ACKNOWLEDGEMENTS

The research described herein has been sponsored by the U.S. Army Research Laboratory under cooperative agreement W912CG-19-2-0001. The statements and opinions expressed in this article do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

REFERENCES

Aleksic, P., Ghodsi, M., Michaely, A., Allauzen, C., Hall, K., Roark, B., ... & Moreno, P. (2015). Bringing contextual information to google speech recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)* (pp. 468-472).

Blanchard, N., Brady, M., Olney, A.M., Glaus, M., Sun, X., Nystrand, M., Samei, B., Kelly, S. & D'Mello, S. (2015). A study of automatic speech recognition in noisy classroom environments for automated dialog analysis. *Artificial Intelligence in Education*, 23-33.

Chao, G. L., Chan, W., & Lane, I. (2019). Speaker-targeted audio-visual models for speech recognition in cocktail-party environments. arXiv preprint arXiv:1906.05962.

Cheng, G., Povey, D., Huang, L., Xu, J., Khudanpur, S., & Yan, Y. (2018). Output-Gate Projected Gated Recurrent Unit for Speech Recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association* (pp. 1793-1797).

Chiu, C. C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., ... & Jaitly, N. (2018, April). State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4774-4778). IEEE.

Georgila, K., Leuski, A., Yanov, V. & Traum, D. (2020). Evaluation of off-the-shelf speech recognizers across diverse dialogue domains. To Appear in *the Proceedings of the Language Resources and Evaluation Conference*.

Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine, 29*(6), 82-97.

Johnston, J. H. (2018). Team performance and assessment in GIFT: Research recommendations based on lessons learned from the Squad Overmatch Research Program. In *Proceedings of the Sixth Annual GIFT Users Symposium* (Vol. 6, p. 175-187). US Army Research Laboratory, Orlando: FL.

Johnston, J. H., Phillips, H. L., Milham, L. M., Riddle, D. L., Townsend, L. N., DeCostanza, A. H., ... & Fitzhugh, S. M. (2019). A team training field research study: Extending a theory of team development. *Frontiers in psychology*, *10*, 1480.

Kim, J. Y., Liu, C., Calvo, R. A., McCabe, K., Taylor, S. C., Schuller, B. W., & Wu, K. (2019). A comparison of online automatic speech recognition systems and the nonverbal responses to unintelligible speech. *arXiv preprint arXiv:1904.12403*.

Kimura, T., Nose, T., Hirooka, S., Chiba, Y., & Ito, A. (2018, November). Comparison of speech recognition performance between Kaldi and Google Cloud Speech API. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (pp. 109-115). Springer, Cham.

Marlow, S. L., Lacerenza, C. N., Paoletti, J., Burke, C. S., & Salas, E. (2018). Does team communication represent a one-size-fits-all approach?: A meta-analysis of team communication and performance. *Organizational Behavior and Human Decision Processes, 144*, 145-170.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P.,Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K. (2011). The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding (No. EPFL- CONF-192584)*. IEEE Signal Processing Society.

Ravanelli, M., Parcollet, T., & Bengio, Y. (2019, May). The pytorch-kaldi speech recognition toolkit. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6465-6469). IEEE.

Rousseau, A., Deléglise, P., & Esteve, Y. (2012, May). TED-LIUM: an Automatic Speech Recognition dedicated corpus. In *LREC* (pp. 125-129).

Singh, C., Venter, M., Muthu, R. K., & Brown, D. (2019). A real-time DSP-based system for voice activity detection and background noise reduction. In *Intelligent Speech Signal Processing* (pp. 39-54). Academic Press.

Serizel, R., & Giuliani, D. (2017). Deep-neural network approaches for speech recognition with heterogeneous groups of speakers including children. *Natural Language Engineering*, *23*(3), 325-350.

Smith-Jentsch, K. A., Johnston, J. H., & Payne, S. C. (1998). Measuring team-related expertise in complex environments. In J. A. Cannon-Bowers & E. Salas (Eds.), *Making decisions under stress: Implications for individual and team training* (p. 61-87). American Psychological Association.

Soukoreff, R. W., & MacKenzie, I. S. (2001, March). Measuring errors in text entry tasks: An application of the Levenshtein string distance statistic. In *CHI'01 extended abstracts on Human factors in computing systems* (pp. 319-320).

Yu, D., & Deng, L. (2016). *Automatic Speech Recognition*. Springer London Limited.

ABOUT THE AUTHORS

Dr. Randall Spain *is a Research Psychologist in the Center for Educational Informatics at North Carolina State University where he uses principles, theories, and methods of applied psychology to design and evaluate the impact of advanced training technologies on learning and performance. Dr. Spain holds a PhD in Human Factors Psychology and serves on the editorial board for Military Psychology.*

Dr. Wookhee Min *is a Research Scientist in the Center for Educational Informatics at North Carolina State University. He received his BS degree in Computer Science from Yonsei University, and MS and PhD in Computer Science from North Carolina State University. He has published 25 papers and issued five patents. His research focuses on artificial intelligence centering on user modeling, educational data mining, multimodal learning analytics, and natural language processing, with an emphasis on game-based learning environments.*

Mr. Jason Saville *is a third-year doctoral student of industrial and organizational psychology at North Carolina State University and Graduate Research Assistant in the Center for Educational Informatics. He received his BA degree in Psychology from North Carolina State in 2018 and also serves as a member of the 4D Lab research team, where his research focuses on the intersections of work, psychology, technology, and global development.*

Dr. Bradford Mott *is a Senior Research Scientist in the Center for Educational Informatics at North Carolina State University. Prior to joining North Carolina State University, he worked in the video game industry developing cross-platform middleware solutions used extensively in commercial games and training applications. His research focuses on game-based learning environments, intelligent tutoring systems, and computational models of interactive narrative. He received his PhD in Computer Science from North Carolina State University in 2006.*

Dr. Keith Brawner *is a Senior Engineer 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.*

Dr. Joan Johnston *is a Senior Scientist with the U. S. Army Combat Capability Development Command Soldier Center at the Simulation and Training Technology Center (CCDC-SC-STTC), where she leads research on training effectiveness for simulations and team training. Prior to ARL, she was a senior research psychologist and NAVAIR Fellow at the Naval Air Warfare Center Training Systems Division. Dr. Johnston received her PhD in Industrial and Organizational Psychology from the University of South Florida.*

Dr. Gregory Goodwin *is a senior research scientist U. S. Army Combat Capability Development Command Soldier Center at the Simulation and Training Technology Center (CCDC-SC-STTC), in Orlando, Florida. For the last decade, he has worked for the Army researching ways to improve training methods and technologies. He holds a PhD in Psychology from Binghamton University and an M.A. in Psychology from Wake Forest University.*

Dr. James Lester *is Distinguished University Professor of Computer Science at North Carolina State University, where he is Director of the Center for Educational Informatics. His research centers on transforming education with technology-rich learning environments. With a focus on adaptive learning technologies, his research spans intelligent tutoring systems, game-based learning environments, affective computing, and tutorial dialogue. The adaptive learning environments he and his colleagues develop have been used by thousands of students in K-12 classrooms. He received his PhD in Computer Science* *from the University of Texas at Austin in 1994. He is a Fellow of the Association for the Advancement of Artificial Intelligence (AAAI).*

Teamwork Training in GIFT: Updates on Measurement and Audio Analysis

Robert McCormack[1], Alexander Case[1], Daniel Howard[1], Jianna Logue[1], Kristy Kay[1], Anne M. Sinatra[2]
Aptima, Inc.[1], U.S. Army Combat Capabilities Development Command (CCDC) – Soldier Center – Simulation & Training Technology Center (STTC)[2]

INTRODUCTION

High performing teams are those that have the expertise to accomplish the task at hand, but also the skills to work together reliably and efficiently. Breakdowns in teamwork are often cited as a cause for poor, and sometimes deadly, outcomes (e.g., Wilson, Salas, Priest & Andrews, 2007). The shoot down of Iran Air flight 655 by the USS Vincennes is a prime example of how failures in teamwork can lead from individual operator error to complete team failure (Bell & Kozlowski, 2011). Conceptually, teamwork can be decomposed into team processes and team states. Team processes include the actions and behaviors required for teams to function, such as coordination and communication. Team states are the emergent properties of a team that impact performance, such as cohesion and shared situational awareness. The relevance of these teamwork skills for team performance has been evidenced in the academic literature across a variety of domains, including the medical field and the military (e.g., Sottilare, Burke, Salas, Sinatra, Johnston & Gilbert, 2018; Wilson et al., 2007).

Traditionally, teamwork skills have been treated as a byproduct of training or operations, thought to emerge (or not) over time as teams work together. Increasingly, though, teamwork has become a focus of training itself – providing teams with the knowledge of, and opportunity to practice essential team skills. Nevertheless, there remain challenges to training teamwork skills efficiently, and consequently there has been a push toward using intelligent tutoring systems (ITSs) and frameworks such as the Generalized Intelligent Framework for Tutoring (GIFT; Sottilare, Brawner, Goldberg & Holden, 2012; Sottilare, Brawner, Sinatra & Johnston, 2017) for this purpose. GIFT provides a basis for conveying knowledge about team states and processes, enabling opportunities to practice these skills in different environments, and delivering feedback about the team's ability to utilize these skills. To this end, the authors have been developing a system utilizing GIFT and VBS3 (Virtual Battle Space 3.0) to deliver scalable teamwork training. This paper will provide an update on the development and implementation of this system (previously presented at GIFTSym 6 and 7; McCormack et al., 2018; McCormack et al., 2019), with particular focus on the measurement strategy and a system for capturing, transcribing, and assessing verbal communications.

TEAMWORK TUTOR

In previous GIFTSym papers, the authors described a teamwork tutor system that enables scalable team training options in GIFT. A brief overview is provided here along with updates to the teamwork skill construct measurements. Previous efforts (Bonner et. al., 2017) at developing teamwork skill training have found success through iterative addition of team members from a single player up to two to three individuals in a small team. While this approach does enable measurement at individual, subteam, and team levels it requires a great deal of effort to carefully consider all possible combinations of individuals when developing measures. Furthermore, scaling to medium to large teams quickly becomes difficult to manage. Our system, on the other hand, utilizes a top-down approach to developing measures and content by starting at the team level and breaking the group down to sub-teams or individuals when necessary. While it may make it more difficult to control measurement at every level, this enables training developers to more easily and quickly develop teamwork measurement constructs focused on overall team skills. In general, both approaches (bottom-up and top-down) can be useful depending on the aims of training, however since our focus is on team-level skill development we chose the latter.

Our system utilizes GIFT and VBS3 to deliver a realistic scenario that provides opportunities to practice and measure teamwork skills. The scenario for this effort is adapted from an Army Basic Leader Course (BLC) Combat Search and Rescue (CSAR) training scenario. Originally developed for live training, it adapts well to training a 9-person squad in the virtual environment. It takes place along a linear path through a heavily wooded area. In the scenario, an F-16 pilot has ejected and landed in the area with their medical condition is unknown. The primary objective of the scenario is for the team to locate and rescue the pilot and then continue on the path. The team receives intelligence that enemy militia are in the area and have been testing novel IED (improvised explosive device) techniques. The scenario is broken down into five smaller vignettes:

1. Setting off from the forward operating base and encountering a potential IED;
2. Locating and rescuing the pilot;
3. Receiving fire from hostile militia;
4. Finding a second potential IED;
5. Encountering an unknown individual with unknown intent.

Previous efforts (McCormack et. al., 2018; McCormack et al., 2019) described these vignettes in detail, so we now turn to our approach to measuring teamwork skills.

MEASURING COHESION

One of the main objectives of this effort is to identify teamwork skills that can be trained and assessed within the aforementioned training system. To this end, the authors utilize a process for identifying, operationalizing, and implementing teamwork constructs as unobtrusive and automated measurements. This process, RADSM (Rational Approach to Developing Systems-based Measures; Orvis et al., 2013), consists of six steps intended to ensure that measures are conceptually sound and contextually relevant. The steps are:

1. Identify environmental context and construct of interest;
2. Determine the attributes and behaviors indicative of the construct;
3. Identify system-based information and data related to the construct;
4. Develop measurement indicators based on the attributes and behaviors that are identifiable in the data;
5. Implement the measurement indicators, and;
6. Validate the indicators through experimentation (Orvis, 2013).

The key insight in this process is at the intersection of top-down, theoretically-driven behaviors and attributes and bottom-up, data driven information from the training environment (in this case VBS3). This provides a conceptually-grounded and realistically-implementable approach to measurement. The end result of this process is a set of behaviorally-anchored measurement indicators that can be assessed automatically and unobtrusively (that is, not requiring human coding or input) given the data available in the system. This real-time assessment of measures is very important for ITSs as it helps to facilitate real-time feedback.

Our focus of this process had been on developing measures of two teamwork constructs: coordination and cohesion. Previously (McCormack et al., 2019) we described a number of measures of coordination that were developed and implemented. We have since turned our attention to cohesion. Performance in a team has been shown to be positively correlated with cohesion (for example, Mullen & Cooper, 1994 showed in their meta-analysis a "small but significant effect" between cohesion and performance). Cohesion is typically decomposed into two categories: task cohesion and social cohesion. Here, we define task cohesion as the ability of team members to work together to achieve a common goal. Social cohesion refers to how well the members of a team like each other and interact (often in the absence of a goal-oriented environment). Through the RADSM process we identified a number of behaviors and attributes of both task and social cohesion in teams. Examples of these are shown in Table 1.

Table 1. Behaviors and Attributes related to Cohesion

Facet of Cohesion	Task / Social	Description
Proportion of communication	Task	All team members contribute somewhat evenly to communications
Team-focused language	Social	Communications with more team-focused (we, us, our) than individual-focused (I, me, my) language are used
Acknowledgement	Task	Communications with acknowledgement are frequently used
Listening and empathy	Social	Active listening, such as engagement in conversation, asking and answering questions, and timely responses are used

Utilizing these and other behaviors and attributes, we next developed relevant measures of cohesion for our training vignette. These are shown in Table 2. Each of these measures have been implemented and tested within the training system.

Table 2. Example Measures of Cohesion

Measure Name	Description
No Person Left Behind	The team members should be maintaining awareness of each other to ensure that no person is falling behind. We measure this by determining, for each person, how long it has been since they have entered any other team member's field of view. This measure starts when the team starts moving down the road in formation and stops when the halt is called. For each person, we measure the amount of time that has elapsed since they have entered any other person's field of view.

Mirroring of Stance/Movement	Team members should have the same stance (standing/kneeling/prone) or movement (walking/running) as each other. We measure the amount of time that more than one individual is out of sync with the rest of the team (e.g., 2 people are running while the rest of the team is walking).
Use of Inclusive Language	Team members should use more inclusive (we/us/our) language than exclusive (I/me/mine/my) language. We measure the ratio of the use of inclusive (We/us/our) to exclusive (I/me/mine) language.
Use of Indicators for Acknowledgment	Team members should use more acknowledgement (roger, copy, okay, got it) messages than negative/uncertain (unable, unknown, don't know, believe, seems, probably, assuming) messages. We measure the ratio of the use of acknowledgement (roger, copy, okay, got it) to negative/uncertain (unable, unknown, don't know, believe, seems, probably, assuming) language.

SPEECH ANALYTICS

Communication is an essential aspect of teamwork, not only for sharing information and coordinating activities, but as a way to develop emergent team states, such as cohesion and trust. Communication can take many forms, such as face-to-face conversations, hand signals, email, and social media. In developing the teamwork tutoring system for GIFT, the authors original approach focused on measuring the actions and behaviors in VBS3 that were indicative of teamwork. However, it became clear that communication is such an essential aspect of teamwork that the effort would be remiss if it did not include the ability to capture and analyze team communications. As demonstrated in the cohesion measures presented in the previous sections, many of the observable behaviors are centered around the way teams speak and interact. To this end, the authors have designed and implemented a system for capturing verbal communications, automatically transcribing them to text, and analyzing them for indicators of coordination and cohesion within GIFT. The system utilizes a set of microservices to capture verbal communications from a computer headset and save them as audio files. These audio files are then sent to a speech-to-text service that transcribes the spoken words to text. The text is analyzed within GIFT to identify indicators of teamwork.

We developed a lightweight microservice that is capable of capturing audio, automatically translating that audio to text, and making the results available to third party applications or services. The architecture for this service is shown in Figure 1. The first component of the speech microservice is an audio recorder. Users wear headsets attached to their computer with the recorder continuously running. Instead of recording everything, the audio recorder can detect when a user is speaking and when they are silent. When a moment of silence is detected after speech, the recorder will save the speech as an audio file and continue listening for the next speech utterance. The audio file is saved automatically to a specified directory. The next component, the speech detection service, monitors that directory for new audio files. When one is detected it sends the audio to a speech-to-text engine. The specific engine used is customizable, but by default, it uses Windows built-in speech-to-text engine. This can be swapped for other another engine such as Amazon's Transcribe service. After the text is produced by the speech engine it is timestamped and saved to a text file. This text file can then be sent to an external application, such as GIFT. Alternatively, an application programming interface API is also available to directly access the transcribed results.

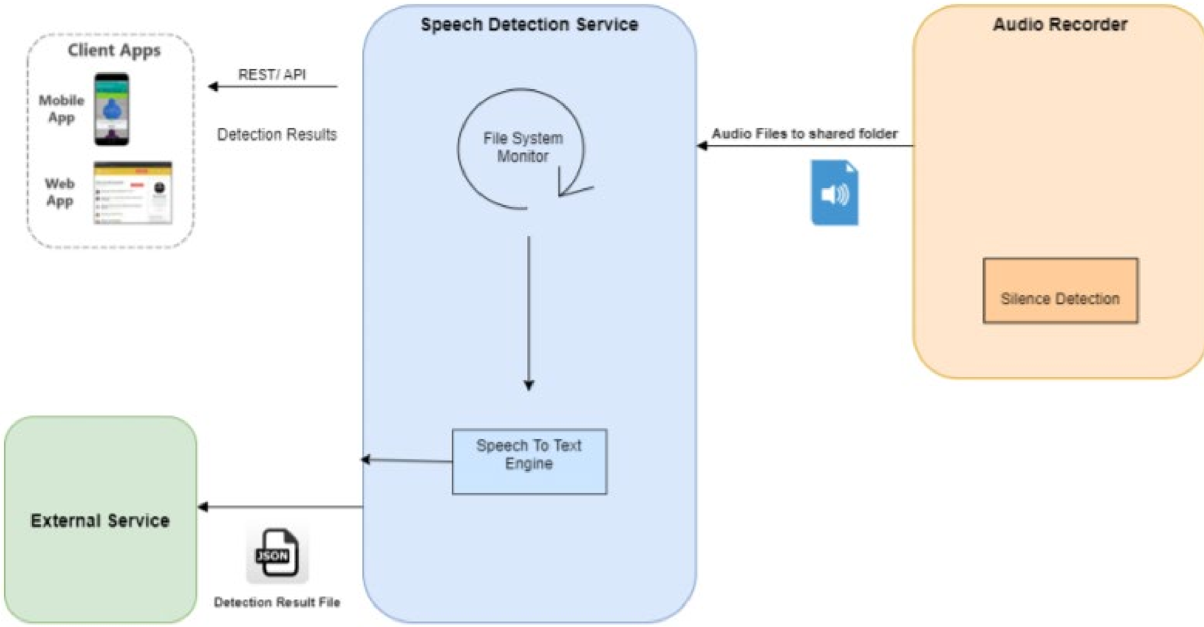


Figure 1. Speech Detection Microservice

As part of the teamwork tutor system, the speech detection microservice enables a wider range of measurement possibilities. Figure 2 shows how the microservice is integrated with GIFT in our system. Each trainee on the team has their own audio recorder and speech detection service running on their computer. A speech recognition interop plugin within GIFT's gateway module receives the transcribed text from the speech detection microservice from each trainee. As such, the system can identify what was said (transcribed text), when (timestamp), and by who (trainee id). Within GIFT's domain module any number of condition classes can access the transcribed speech from the gateway module to enable speech-aware measurements.

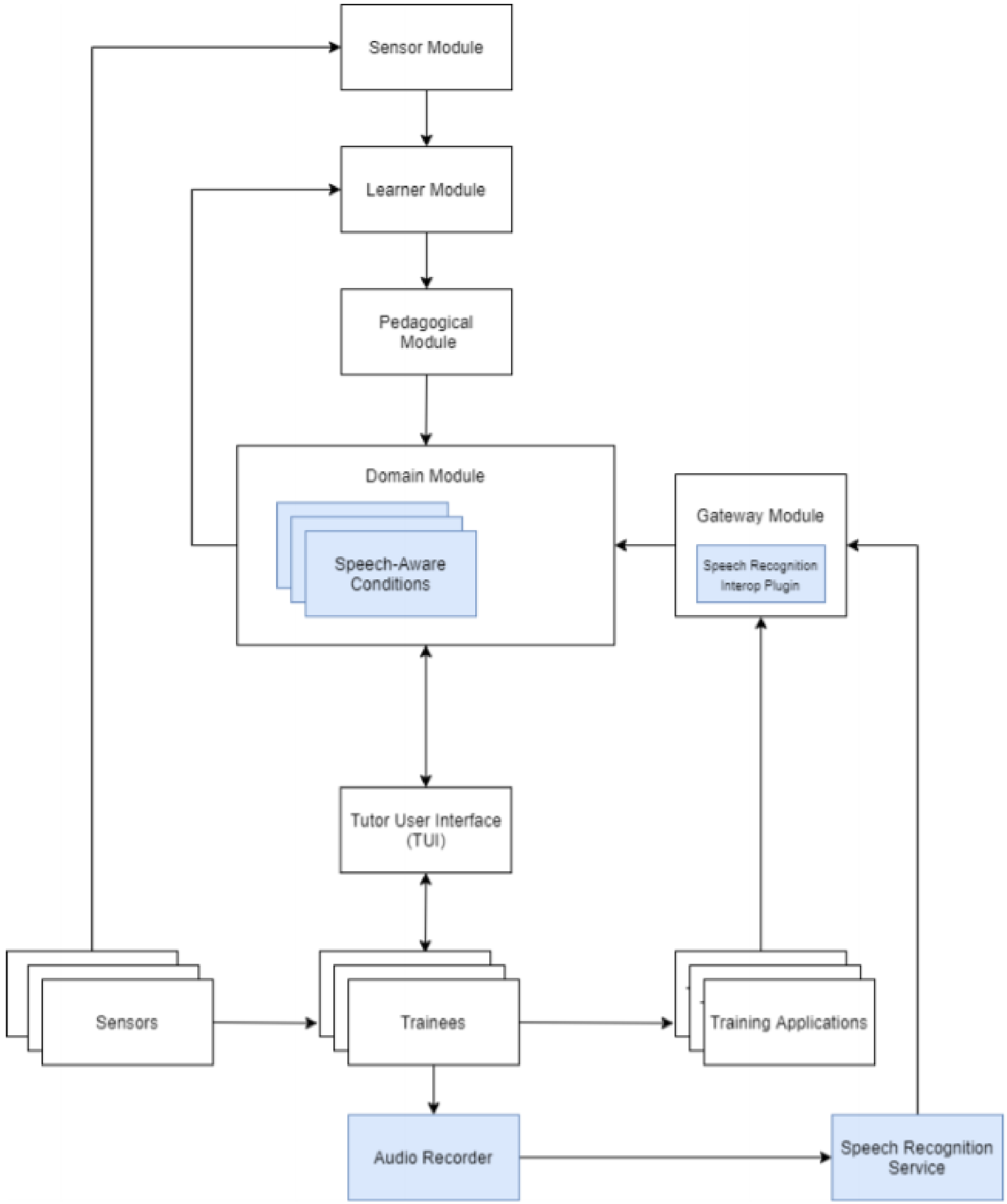


Figure 2. Use of the Speech Detection Microservice within GIFT

Within the teamwork tutor system, we have successfully implemented speech-aware condition classes for a number of coordination and cohesion measures. For the coordination construct we have implemented a condition that listens for sequences of words spoken in a particular order by different individuals. This enables measurement of coordinated sequences of speech, such as the "5 Cs" of IED mitigation (see the authors' previous paper presented at GIFTSym 2019 for a description of the this measure). For cohesion, several classes have been developed and tested that detect the use of certain types of language, such as inclusive speech and acknowledgements. These classes are widely applicable to many different types of measurement in different contexts.

CONCLUSION AND RECOMMENDATION FOR FUTURE RESEARCH

The teamwork training architecture, training scenario, and novel measurement approach for assessing teamwork skills described in the paper offers a promising solution to rapidly training teams in virtual environments using GIFT. Furthermore, the speech-to-text service enables unobtrusive and more naturalistic assessment of team communications and interactions. The next step in this work will be to both validate the effectiveness of the training solution as well as assess the validity of the coordination and cohesion measures. To this end, the authors plan to conduct an experiment utilizing the teamwork tutor system. This experiment aims to extend previous research evaluating GIFT as a viable option for team training by accomplishing two goals: (1) examining the effect of real-time versus post-vignette feedback on team processes and performance, and (2) validating the measures of cohesion and coordination designed and implemented within GIFT. The main manipulation in this study will be presence of real-time feedback as teams progress through the vignettes versus only feedback given intermittently at the end of each vignette. The overall hypothesis is "the functional performance difference score (i.e., performance enhancement) will be greater for teams that receive real-time feedback throughout the vignettes compared to post-vignette feedback." Results of this study are intended to be reported at the next GIFTSym.

ACKNOWLEDGEMENTS

The research reported in this document was performed in connection with contract number W911NF-17-C-0061 with the U.S. Army Contracting Command - Aberdeen Proving Ground (ACC-APG). The views and conclusions contained in this document/presentation are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of ACC-APG, U.S. Army Combat Capabilities Development Command (CCDC) – Soldier Center, U.S. Army Research Laboratory or the U.S. Government unless so designated by other authorized documents. Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

Bell, B. S., & Kozlowski, S. W. J. (2011). Collective failure: The emergence, consequences, and management of errors in teams. In D. A. Hoffman & M. Frese (Eds.), Errors in Organizations (pp. 113-141). New York, NY: Routledge.

McCormack, R., Kilcullen, T., Sinatra, A., Brown, T., & Beaubien, J. (2018). Scenarios for Training Teamwork Skills in Virtual Environments with GIFT. Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym6), Orlando, FL.

McCormack, R., Kilcullen, T., Sinatra, A., Case, A., & Howard, D. (2019). Teamwork Training Architecture, Scenarios, and Measures in GIFT. Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym7), Orlando, FL.

Mullen, B., & Copper, C. (1994). The relation between group cohesiveness and performance: An integration. Psychological Bulletin, 155, 210-227.

Orvis, K. L., Duchon, A., & DeCostanza, A. (2013, January). Developing Systems-based Performance Measures: A Rational Approach. In The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC) (Vol. 2013, No. 1). National Training Systems Association.

Sottilare, R., Brawner, K., Goldberg, B. & Holden, H. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). US Army Research Laboratory.

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.

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.

Wilson, K. A., Salas, E., Priest, H. A., & Andrews, D. (2007). Errors in the heat of battle: Taking a closer look at shared cognition breakdowns through teamwork. Human Factors, 49, 243–256.

ABOUT THE AUTHORS

Dr. Robert K. McCormack *is a Principal Mathematician and Deputy Director of the Performance Assessment Technologies Division at Aptima. He has expertise in the areas of unobtrusive measurement, computational linguistics (NLP), machine learning, epidemiological modeling, and human sociocultural modeling and analysis. Dr. McCormack received a Ph.D. and M.S. in Mathematics from Texas Tech University, and a B.A. in Mathematics and Computer Science from Austin College*

Mr. Alexander Case *is a Software Engineer at Aptima, Inc. His technical interests focus on front- and back-end applications, data collection and analysis, and video game and simulation development. Other areas of interest include application telemetry, containerization, continuous integration, and configuration management. While at Aptima, he has worked on Angular web applications and developed server-side applications in Python and C#. Mr. Case holds a Bachelor's degree in Computer Science from Bridgewater State University.*

Mr. Daniel Howard *is a Principal Software Engineer at Aptima, Inc. Mr. Howard works on developing a variety of applications focusing on complex visualization and the analysis of large volumes of data, and is responsible for the efficient storage and retrieval of this measurement data through Aptima's ASA™ framework. Mr. Howard holds a M.S. and B.S. in Computer Engineering from Rochester Institute of Technology.*

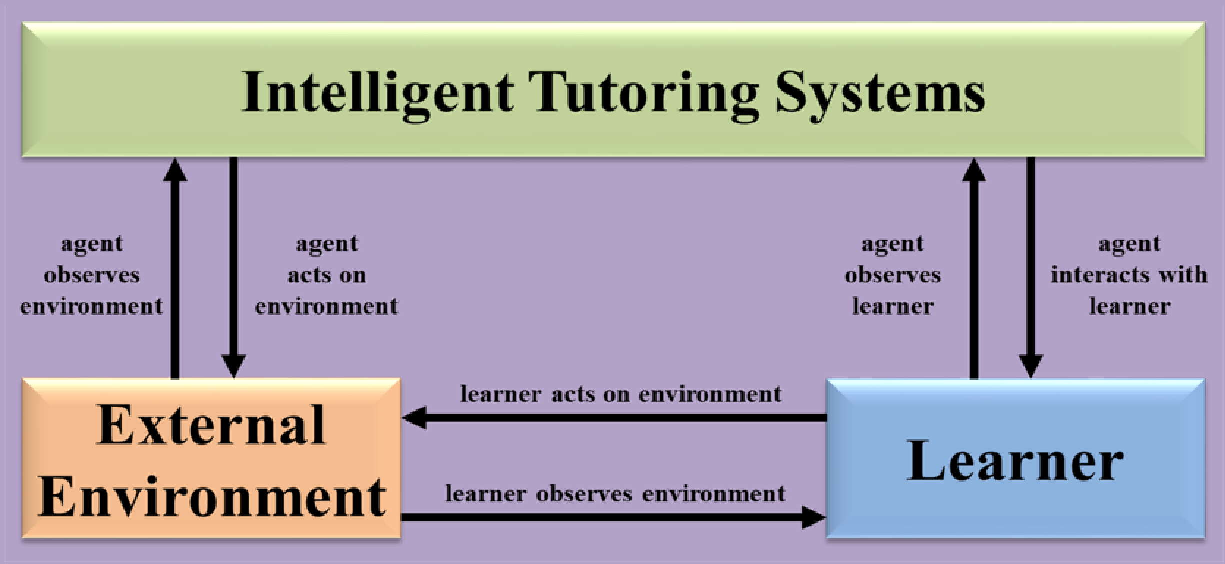
Ms. Jianna Logue *is an Associate Software Engineer at Aptima, Inc. with experience programming in C#, C, C++, Java, and Angular, and other Web technologies. Ms. Logue has a B.A. in Russian Language and Literature from Georgetown University and an A.S. in Computer Programming and Analysis from Seminole State College. She is currently working toward a second B.S. in Computer Science at University of Central Florida.*

Ms. Kristy Kay *is Scientist at Aptima, Inc. with research experience in performance measurement, team dynamics and training, training design and evaluation, and data analysis. Ms. Kay received a M.S. in Industrial/Organizational Psychology from San Diego State University, and a B.A. in Psychology from California State University San Marcos. She holds an Adjunct Faculty position in the Industrial/Organizational Psychology program at San Diego State University.*

Dr. Anne M. Sinatra *is a Research Psychologist, and part of the adaptive training research team within CCDC Soldier Center – STTC. She works on the Generalized Intelligent Framework for Tutoring (GIFT) project. Her background is in Human Factors and Cognitive Psychology.*

Proceedings of the Eighth Annual GIFT Users Symposium

GIFT, the Generalized Intelligent Framework for Tutoring, is a modular, service-oriented architecture developed to lower the skills and time needed to author effective adaptive instruction. Design goals for GIFT also include capturing best instructional practices, promoting standardization and reuse for adaptive instructional content and methods, and technologies for evaluating the effectiveness of tutoring applications. Truly adaptive systems make intelligent (optimal) decisions about tailoring instruction in real-time and make these decisions based on information about the learner and conditions in the instructional environment.



The GIFT Users Symposia began in 2013 to capture successful implementations of GIFT from the user community and to share recommendations leading to more useful capabilities for GIFT authors, researchers, and learners.

About the Editor:

- ***Dr. Benjamin S. Goldberg*** *leads adaptive training research at the U.S. Army Combat Capability Development Command – Solider Center and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT).*

Part of the Adaptive Tutoring Series

9 780997 725803