Proceedings of the Tenth Annual GIFT Users Symposium

May 2022
Orlando, Florida
(Virtual)

Celebrating 10 Years of GIFTSym

Edited by:
Anne M. Sinatra

Part of the Adaptive Tutoring Series

Proceedings of the 10th Annual GIFT Users Symposium (GIFTSym10)

Proceedings of the 10th Annual

Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym10)

Edited by:
Anne M. Sinatra

The views expressed herein are those of the authors and do not necessarily reflect the views of the U.S. Army Combat Capabilities Development Command – Soldier Center

Use of trade names or names of commercial sources is for information only and does not imply endorsement by the U.S. Army Combat Capabilities Development Command – Soldier Center.

This publication is intended to provide accurate information regarding the subject matter addressed herein. The information in this publication is subject to change at any time without notice. The U.S. Army Combat Capabilities Development Command – Soldier Center, nor the authors of the publication, makes any guarantees or warranties concerning the information contained herein.

Printed in the United States of America
First Printing, May 2022

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

International Standard Book Number: 978-0-9977258-2-7

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

CONTENTS

FROM THE EDITOR

Welcome to the Proceedings of the 10th Annual GIFT User Symposium! This year we are celebrating 10 years of GIFT Symposiums and have accepted 14 papers for publication. These papers all showcase Generalized Intelligent Framework for Tutoring (GIFT) and the work that is being done with GIFT.

GIFT is a flexible open-source intelligent tutoring system (ITS) architecture. GIFT is free, and the architecture is domain independent, which allows for reuse. GIFT is being developed with multiple goals in mind including the efficient creation of Adaptive Instructional Systems (AISs), as well as to support research into tutoring best practices. The authoring tools have been created and improved through the years with both user experience and flexibility in mind.

Our fantastic team, and program committee did a great job assisting with the development of GIFTSym10, and reviewing this year. We want to recognize them for their efforts:

- **Keith Brawner**
- **Elyse Burmester**
- **Benjamin Goldberg**
- **Gregory Goodwin**
- **Michael Hoffman**

As this is the 10th GIFTSym we are adding this years' 14 papers to an already large documented series of research. In those 10 years GIFTSym has taken place in 3 different locations (Memphis, TN; Pittsburgh, PA; Orlando, FL) in addition to virtually!

We encourage you to visit the documents tab on giftuttoring.org (https://www.gifttutoring.org/projects/gift/documents) and view the proceedings of previous GIFTSym years. We are proud to have 10 volumes which include a total of **178** papers. The proceedings are a great resource to see how GIFT has changed through the years.

These current (and previous) proceedings document the work that has been done with GIFT, and create a resource to see GIFT's development over time. Specifically, the themes for this year's GIFTSym include:

- New GIFT Features and Guides
- Authoring Tools
- Adaptive Instructional System (AIS) Architecture and Ontology
- Measurement and Assessment
- Collective/Team Based Methods

The editor and program committee would like to thank all authors and contributors on the papers in this volume – as well as previous GIFTSym volumes. The GIFT community's contributions, lessons learned, suggestions, and research have helped shape GIFT to become what it is today.

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 for Intelligent Tutoring Systems book series, technical reports, journal articles, and conference papers. GIFTtutoring.org also includes a users' forum to feedback on GIFT and influence its future development.

Thank you for 10 great years of GIFTSym!

Anne M. Sinatra, Ph.D.
GIFTSym10 Chair and Proceedings Editor

Theme I: New GIFT Features and Guides

The GIFT Architecture and Features Update: 2022 Edition

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

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 and 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 user requirements suggested in the previous GIFTSym9 proceedings. The development team takes comments within the symposium seriously, and this paper serves to address requirements from prior years.

As a follow up to the previous GIFT Symposium architecture updates (Brawner & Ososky, 2015; Ososky & Brawner, 2016; Brawner et al., 2017; Brawner & Hoffman, 2018; Brawner et al., 2019; Goldberg et al., 2020, Hoffman et al., 2021) this version highlights new tools and 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 the new members of the GIFT community and new GIFT users – 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 et al., 2012). First, we would encourage you to simply get started, as the tools and example courses have been designed to assist users in exploring GIFT's tools and methods for the purpose of creating Adaptive Instructional Systems.

If you struggle with any individual aspect of the system, the team has produced short "how to" videos to help around the sticking points. There are now many videos available on 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 1600 postings and responses.

Documentation has been made freely available online at https://gifttutoring.org/projects/gift/wiki/Documentation, with interface control documentation available at https://gifttutoring.org/projects/gift/wiki/Interface_Control_Document_2021-2, and a developer guide available at https://gifttutoring.org/projects/gift/wiki/Developer_Guide_2021-2. These documents are updated each software release. In this release, we would also like to highlight the available instructions for hosting your own

Amazon Web Services (AWS) instance (https://gifttutoring.org/projects/gift/wiki/Amazon_Web_Service_Install_Instructions).

GIFT Development and Release Strategy

There are two GIFT instances available to everyday users, GIFT Cloud and GIFT Desktop. GIFT Cloud follows an every-Friday system update schedule when relevant updates are ready from the engineering team. For the desktop version, we have maintained a 12 month or less release cycle with a recent regression tested release in November 2021. To support experimentation, intermittent extensions of the core GIFT baseline are performed to facilitate data and interaction requirements based on specific research questions of interest. These are performed on an "as needed" basis, and often serve as the feature extensions included in the next public-release. 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 six years over AWS. 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 supports hundreds of simultaneous users for experiments. We are generally confident in the systems' ability to stay up and cope with demand. The current limitations are that team training in a virtual environment and sensor-based interactions are not supported on the cloud instance, but that requirement will be addressed.

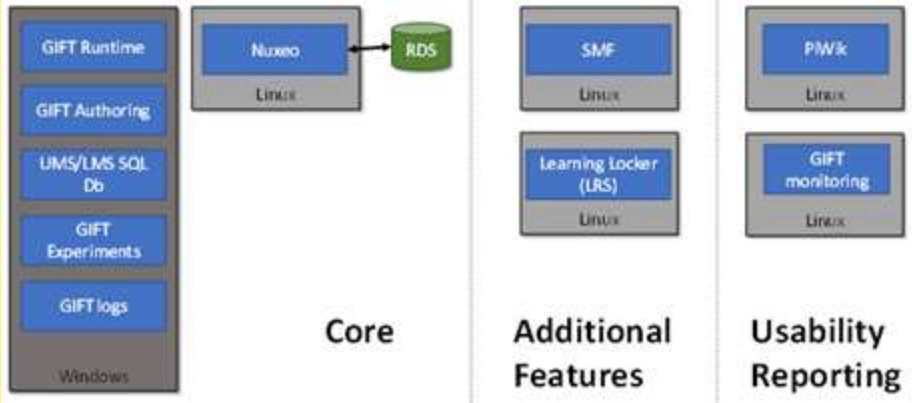


Figure 1. Simple Diagram Overview of GIFT Cloud Items

Behind the scenes, however, the re-tooling to move to a deployment version of development in a desktop instance to a cloud environment in production has been working well. For the remainder of the paper, we will cover the latest improvements added over the last development cycle.

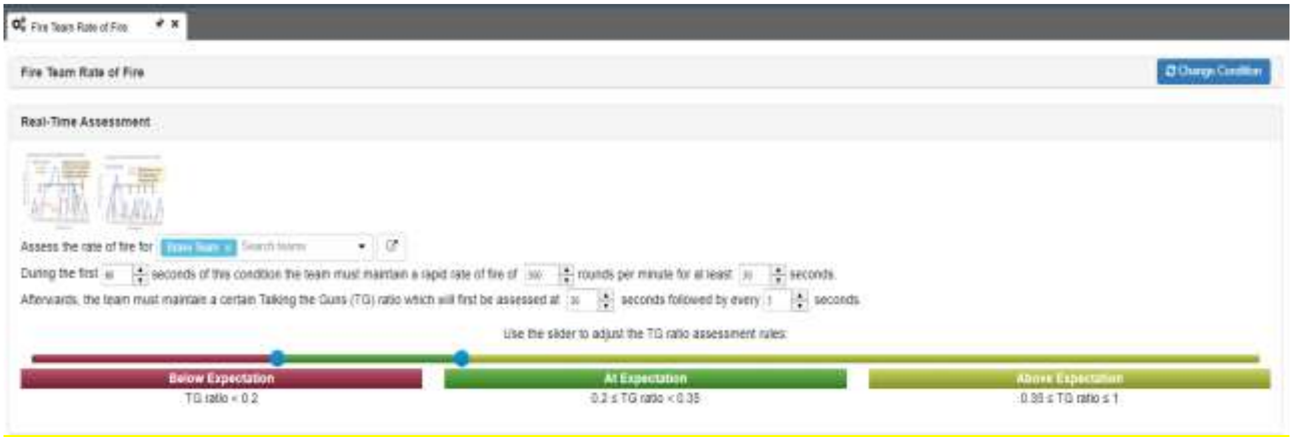
NEW GIFT FEATURES AND UPDATES

Since the last feature update from GIFTSym9 (Hoffman et al., 2021), 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 open source version of GIFT or on GIFT Cloud. Each new feature will be presented with information on the functions it supports and the system and data level dependencies to implement.

Changes to Conditions for Automated Assessments and Data Capture

One of the most powerful features of GIFT is the ability to automate real time assessments across an array of supported training and simulation-based environments. Currently the majority of these assessments take place in condition classes written in Java. One of these condition classes, the Fire Team Rate of Fire, was improved recently.

Fire Team Rate of Fire



Assess the rate of fire for Fire Team × Search teams

During the first 60 seconds of this condition the team must maintain a rapid rate of fire of 300 rounds per minute for at least 30 seconds.

Afterwards, the team must maintain a certain Talking the Guns (TG) ratio which will first be assessed at 30 seconds followed by every 1 seconds.

Use the slider to adjust the TG ratio assessment rules:

Figure 2. Screenshot of the Fire Team Rate of Fire condition authoring user interface in the GIFT course creator.

The Fire Team Rate of Fire condition (see Figure 2) is used to assess whether a team is maintaining an appropriate level of suppressive weapon fire over a specified period of time. This condition can be used in situations where a distributed ratio of weapon fire should happen to prevent the enemy from maneuvering. During this engagement, there might be instances where a weapon system is inoperable due to a jam or reloading. In order for the appropriate rate of fire to be maintained, the other team members must increase their rate of fire. This is normally a coordinated effort involving leadership and communication. The condition provides a mechanism to assess the underlying behaviors and diagnostics of cooperation and performance across the team. While not a new condition, it has received some minor changes. In the previous version of this condition, the author could decide the rate at which an assessment would happen from a choice of one, two, three, four or five minutes. Now the condition provides the first assessment after an initial window of time, followed by reoccurring assessments at another defined time interval. This change accounts for the team transitioning into a suppressive rate of fire and then maintaining it over time. Additional logic was added to account for a rapid rate of fire that normally happens immediately after reacting to contact and before suppressive fire. Here the author can specify how long the team should maintain a rapid rate of fire and what are the minimum rounds per minute that needs to be fired by the team. While this additional condition logic improves the fidelity of the automated assessment, there are still additional features that could be added in the future such as analyzing team communication, and tracking which weapon system or team members are not following doctrine for improved after action review. This condition class, as with many others, is used to provide automated assessment while in a training application. In the most recent GIFT release, GIFT was integrated with another training application called the SE Sandbox.

SE Sandbox Training Application

The SE Sandbox application joins a growing list of training environments integrated into GIFT. While SE Sandbox is built to be game engine agnostic, the version available on the GIFT portal for the community to use with the GIFT 2021-2 release, leverages Unity and contains a scenario called "React to Fire (Forest)". To add this application to a GIFT Course, a new course object was added to the course creator (seen in Figure 3). For runtime communication between GIFT and SE Sandbox, a new Gateway module interop plugin class was created. These two pieces are part of the standard development approach for integrating any new training applications into GIFT; please refer to the GIFT Software Developer Guide wiki for more information (https://gifttutoring.org/projects/gift/wiki/Developer_Guide_2021-2). The communication between GIFT and SE Sandbox is comprised of Google Remote Procedure Calls (gRPC) and basic Socket messaging using Google Protocol Buffer message syntax definition.

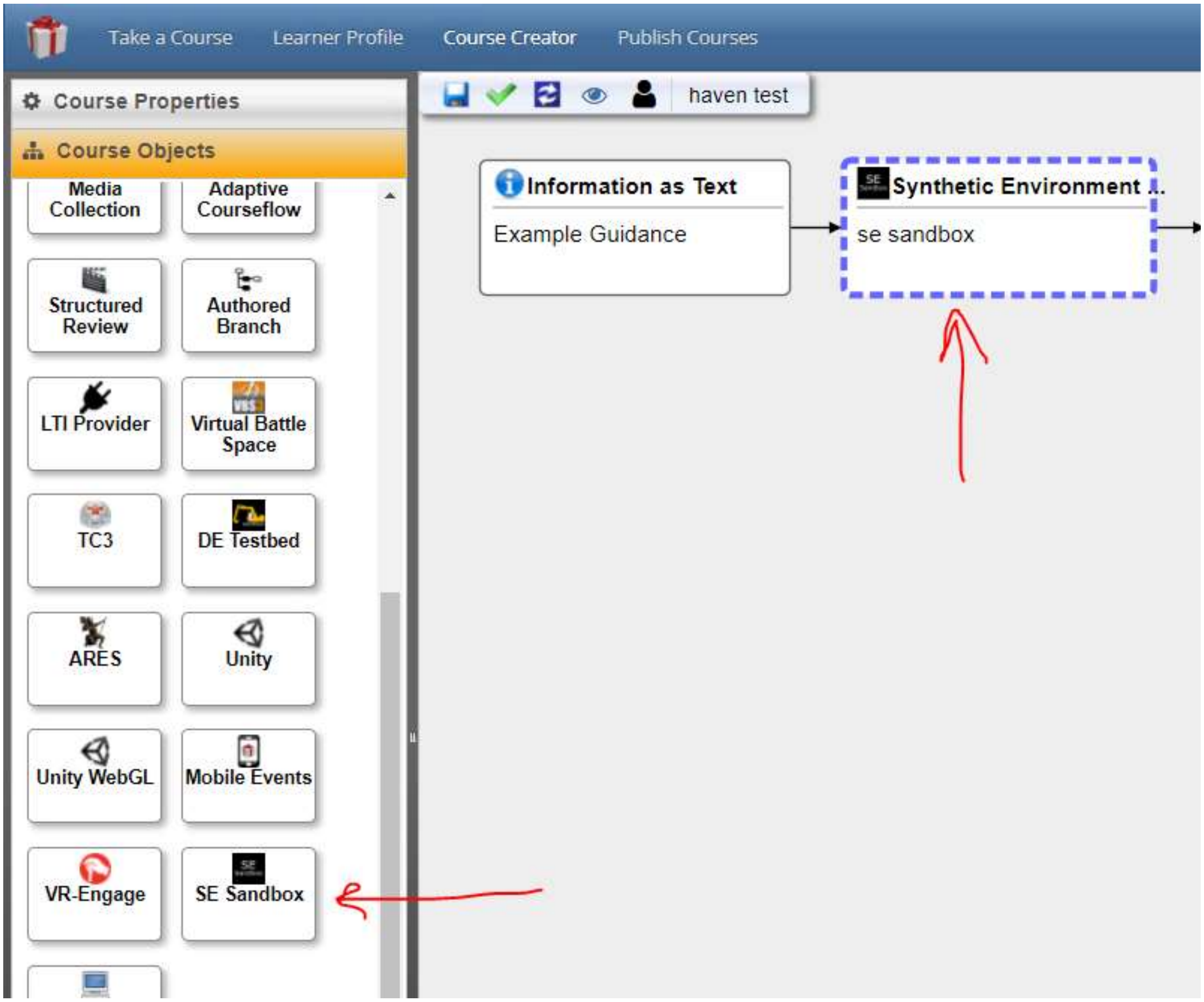


Figure 3. Screenshot of the SE Sandbox Course object in the Course Creator

The SE Sandbox scenario is paired with three GIFT Courses in the GIFT release. The first course titled "Set Up Your Machine for SE Sandbox" provides step by step guide to help the user ready their computer for running the GIFT – SE Sandbox integration. From there users are encouraged to take the second GIFT course called "STE SE Sandbox Individual Exemplar" which involves a single player following instructions delivered in GIFT's Tutor User Interface (TUI) while walking around in the virtual world as seen in Figure 4.

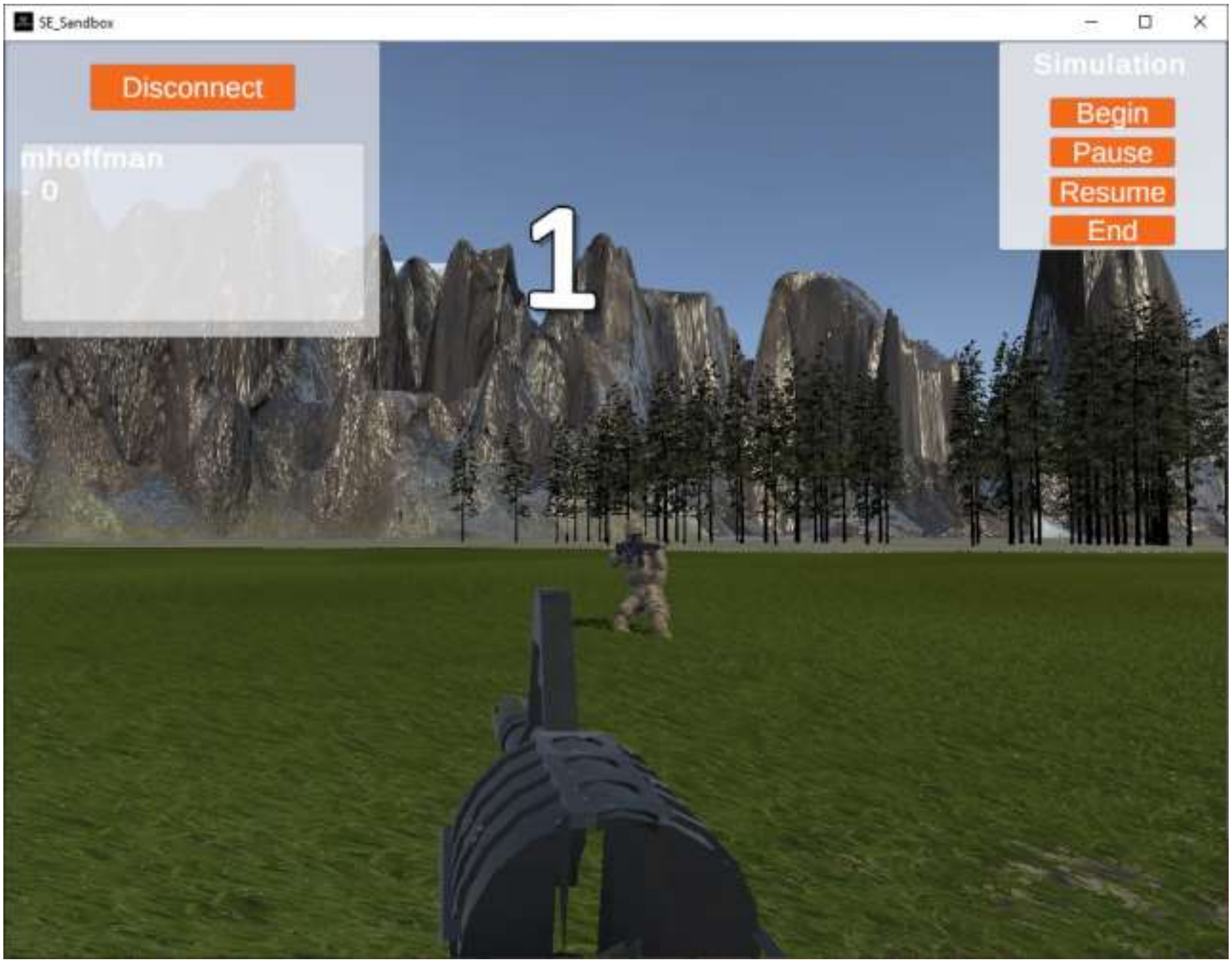
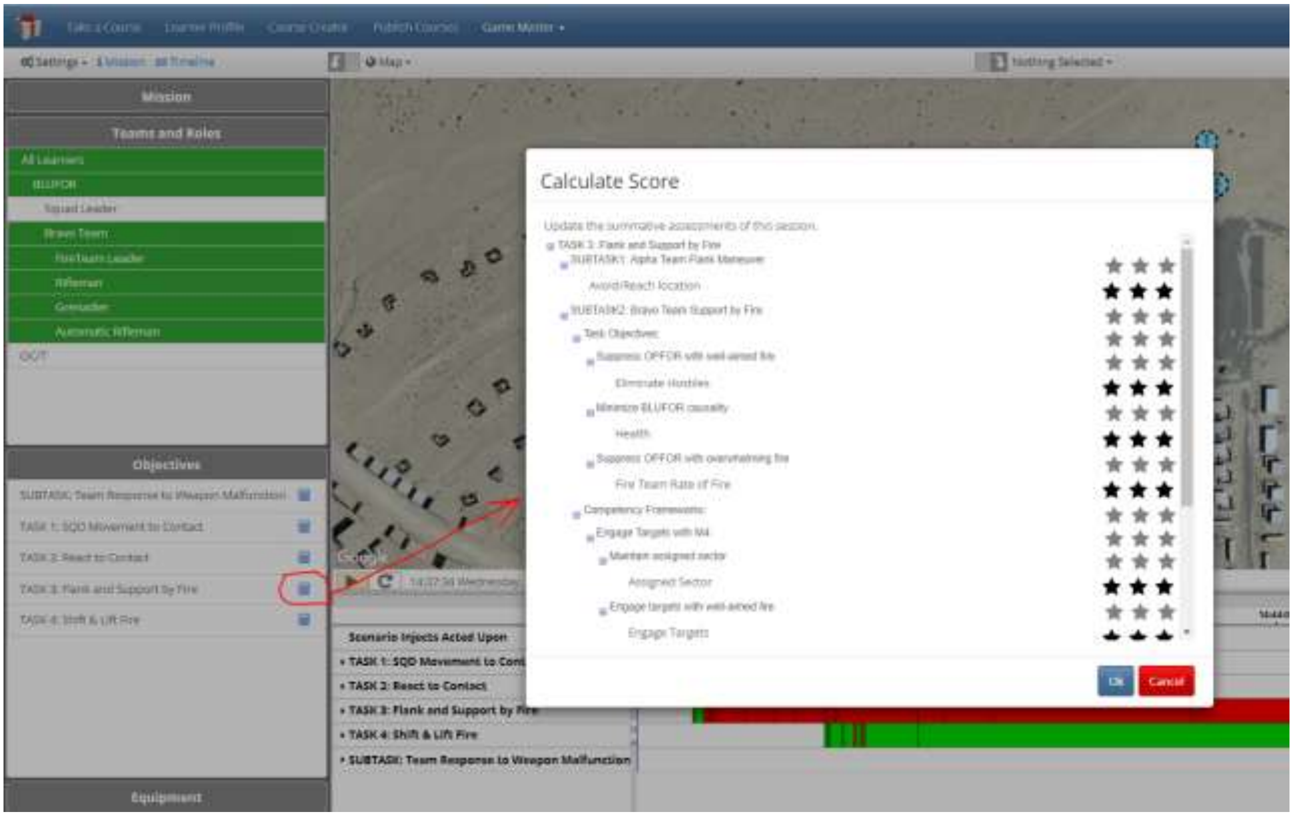


Figure 4. Screenshot of the React to Contact (Forest) SE Sandbox scenario

While the scenario unfolds, GIFT is executing a Domain Knowledge File (DKF) that monitors attributes such as learner location, orientation, weapon fire, detonation and timing. These attributes are used to calculate automated assessments of performance which then trigger scenario adaptations to be applied such as feedback and further instructions. The third and final GIFT Course, "STE SE Sandbox Team Exemplar", allows up to four players to train in the virtual world while GIFT assesses their actions. While a learner executes the training in SE Sandbox or any other training application, we often mention that an Observer Coach/Trainer (OC/T) should be monitoring the session using the GIFT Game Master. The Game Master provides a dashboard to manage the various data streams available to GIFT. Recently the Game Master received a few improvements.

Game Master Improvements

During the summer months of 2019, the GIFT Game Master was created. The first public version made its way into the GIFT 2020-1 official release in April of 2020. Since that time, the Game Master has routinely been shown in GIFT demonstrations and the GIFT team is continuously looking for ways to improve the user experience. One recent improvement involved showing weapon fire and detonation events on the map panel, represented as colored circles. In addition, a fire line is drawn between the firing entity and the location of the detonation to allow for easier identification of firing direction. This simple improvement really highlights engagements to the observer for easier situational awareness and attention. Another noticeable change to the Game Master Past Session user interface is a new focus on summative assessments. In the past the Game Master was entirely centered around formative assessments, i.e. learner state changes regarding changes in real-time performance. Summative assessments provide an overall score for each level of a real-time assessment (i.e. DKF). Now the timeline panel will default to showing the summative scores instead of the formative assessments. The instructor can also edit these scores as needed in the new summative score dialog seen in Figure 5. Any changes made cause new data entries to be made while also preserving the original data. At a minimum, these entries are saved in GIFT but when a Learner Record Store (LRS) is enabled, Experience Application Programming Interface (xAPI) statements are generated as well which is explained in the next section.



Calculate Score

Update the summative assessments of this session.

- TASK 3: Flank and Support by Fire
  - SUBTASK1: Alpha Team Flank Maneuver ★★★
    - Avoid/Reach location ★★★
  - SUBTASK2: Bravo Team Support by Fire ★★★
    - Task Objectives: ★★★
      - Suppress OPFOR with well-aimed fire ★★★
        - Eliminate Hostiles ★★★
      - Minimize BLUFOR casualty ★★★
        - Health ★★★
      - Suppress OPFOR with overwhelming fire ★★★
        - Fire Team Rate of Fire ★★★
    - Competency Frameworks: ★★★
      - Engage Targets with M4 ★★★
        - Maintain assigned sector ★★★
          - Assigned Sector ★★★
        - Engage targets with well-aimed fire ★★★
          - Engage Targets ★★★

Figure 5. Screenshot of the new Summative Assessment scoring dialog in the Game Master user interface

xAPI Improvements

There is still an ongoing focus to establish a persistent modeling capability in GIFT to support competency-based training methods and data interoperability requirements for utilization within a learning ecosystem. We continue our effort to map assessments and outcomes captured in GIFT with competency frameworks that track evidence and performance over long periods of time. To accomplish this, we are integrating and extending core components and data specifications associated with the Advanced Distributed Learning (ADL) Initiative's Total Learning Architecture (TLA; Walcutt & Schatz, 2019) to enable tracking of experiential learning events that are delivered across simulation and synthetic resources. This extends the current utility of the TLA beyond a traditional distributed learning model with an emphasis on tracking human performance related experiences across multiple engagements.

To enable this vision, we are currently integrating the TLA's Competency and Skill System (CASS) with GIFT through a custom xAPI profile. The xAPI community is embracing the technology through the implementation of xAPI Profiles (Bowe & Silvers, 2018) within the IEEE Learning Technology Standards Committee (LTSC) (Robson & Barr, 2018). As GIFT's ability to produce and consume xAPI statements matures, the logic is rolled into GIFT Cloud and GIFT Releases for the community. Since the last GIFT Symposium a year ago, there have been several improvements made in GIFT's Learning Management System (LMS) Module, which is responsible for dealing with xAPI statements. For more information including examples of xAPI statements GIFT produces, visit https://gifttutoring.org/projects/gift/wiki/XAPI_Statements_2022-1. We also welcome participation, and more information on the exact developments can be found at:

- CASS - https://www.cassproject.org/
- xAPI Profiles - http://sites.ieee.org/sagroups-9274-1-1/
- LTSC - http://sites.ieee.org/sagroups-ltsc/home/

Chain of Custody

While GIFT xAPI statements currently provide a great deal of information to any other system that may be reading from the LRS, they do not contain the level of detail stored within the output of a GIFT domain session. GIFT writes a log file of all of the inter-module communications including all training application state messages. It also keeps a copy of the DKF that was used during assessment for future reference. The domain session output can also include other artifacts such as captured audio and video files. This collection of files is useful for recreating the training experience. To connect the dots from the LRS back to the GIFT instance that created the statements, and then to the domain session output for that lesson, attributes (e.g. IP address, GIFT install location, session output folder location) were added to all GIFT created xAPI statements, and are collectively referred to as the chain of custody.

Environment Adaptation

A new statement was created to include the details of an environment adaptation strategy applied by GIFT during a DKF part of a GIFT course. For example, GIFT could change the time of day to midnight in SE Sandbox to increase the difficulty of performing a task. In the future, other strategy types besides Environment Adaptation, such as Feedback could be expressed in a GIFT xAPI statement.

Bookmark/Note

A bookmark is used as a marker of an important event or observation made by the OC/T during an active session. It can be created in the Game Master by the OC/T monitoring a GIFT session. The xAPI statement includes information about who created the bookmark, when it was created and the content of the bookmark (e.g. empty message, text, reference to audio file).

Overriding Summative Assessments

With the recent added ability to edit the summative assessments of a GIFT session in the Game Master Past Session user interface, a new process was introduced to manage the appropriate xAPI statements. When an existing summative assessment is changed, the corresponding xAPI statement is voided to indicate that the statement does not contain the latest information. A new statement is created that contains the newly provided assessment value. The new statement also contains a reference to the voided statement to provide a historical link to what happened during action and how it was changed after action.

Stress and Difficulty

In support of ongoing efforts to recommend, plan for, execute on, and assess learner experiences under different conditions, GIFT now has the ability to define and track stress and difficulty values at both the Task and Strategy components of DKFs. These new values have made their way into the appropriate, existing xAPI statements as additional attributes when available in GIFT.

GIFT xAPI Profile

The GIFT xAPI Profile is now available as a JSON-LD file in the folder GIFT\config\lms\profiles\ of a GIFT instance. It describes the syntax of the xAPI statements GIFT can produce. The profile is not complete and continues to be updated based on the latest developments as they mature.

Experience Training Support Package (xTSP) Import

To help automate the creation of a GIFT DKF in support of a larger Plan, Prepare, Execute, Assess phased training approach, GIFT now allows for importing an Experience Training Support Package (xTSP). The xTSP is a JSON file that contains information used to populate a training application environment and defines the structure of the tasks, standards and conditions under which training will take place. As seen in Figure 6, the GIFT Course Creator allows the author to import an xTSP file to help automate the creation, or partial creation, of a DKF. This lowers the learning curve and time required to create a DKF from scratch, by far the most difficult part of authoring in GIFT to date. In the future, we hope to expose user friendly tools by which an xTSP can be created.

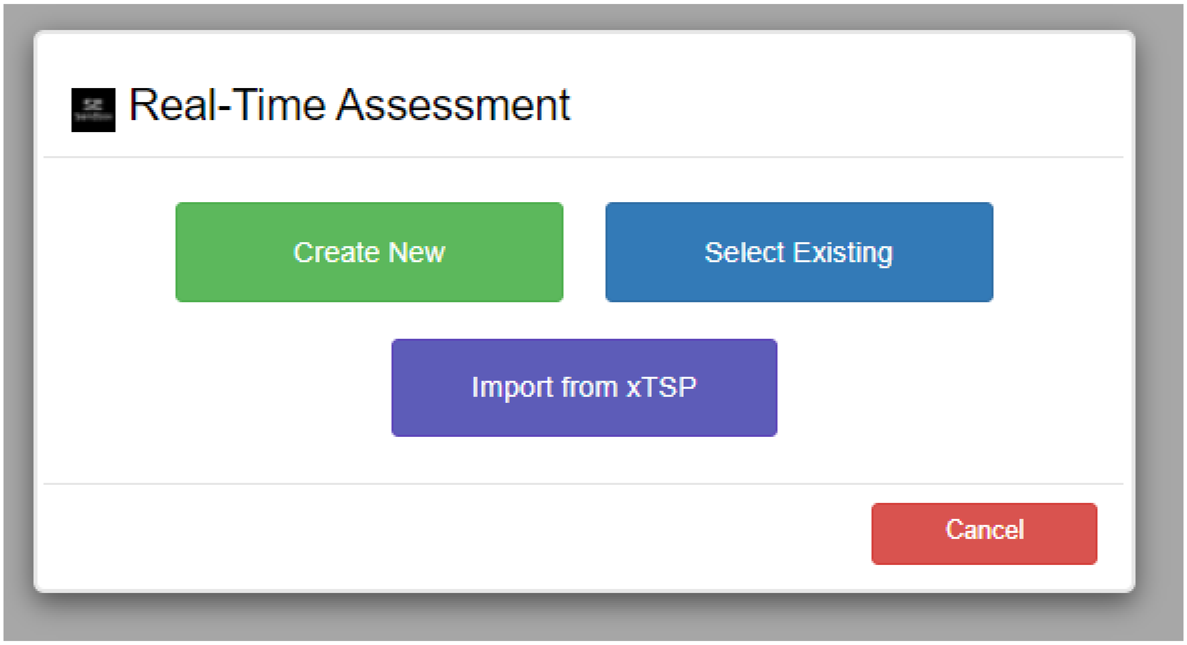


Figure 6. Screenshot of the dialog used to import an xTSP to automatically create a GIFT DKF.

Miscellaneous Improvements

It is worth noting that several other major changes were applied to GIFT. The list below includes a few highlighted changes. Refer to the GIFT Release notes wiki page for more information.

- GIFT messages were converted from JSON to Google Protocol Buffers for better compression and schema defined message syntax. The GIFT ICD wiki page has been updated to reflect this change.

- Upgraded Java from JDK 8 to 11 (required several other third party library updates, replaced Java Web Start delivery of Remote Gateway module with a zip file).

- Media support in GIFT Surveys was greatly improved to include an audio player and a video player.

- Courses on the 'Take a Course' Dashboard page are now automatically Published Courses allowing the user to manage data collected and run reports on the data, something that you could only do with GIFT Experiment or LTI Published Courses before.

REQUESTED FEATURES FROM GIFTSYM9

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, there were relatively few papers which actively requested or demanded features for development. This is good and shows a robust platform – the majority of papers presented describe an activity which is ongoing with GIFT, rather than addressing some weakness or shortfall.

GIFT AND IEEE STANDARDS ON ADAPTIVE INSTRUCTIONAL SYSTEMS

The discussion continues on adaptive instructional systems through the IEEE Learning Technologies Standards Committee (LTSC). LTSC coordinates with other organizations that produce specifications and standards for learning technologies. The GIFT community invites the reader to join the conversation on what data exchange standards for learning technologies might look like in the future. GIFT is scheduled to be included in the Adaptive Instructional Systems Consortium Resource Repository later this year as a tutoring architecture. 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 are able to make!

REFERENCES

Bowe, M., & Silvers, A. (2018). US DoD xAPI Profile Server Recommendations. Data Interoperability Standards Consortium. https://adlnet.gov/publications/2018/08/us-dod-xapi-profile-server-recommendations/

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., & 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., Hoffman, M., Nye, B., & Meyer, C. (2019). Architecture and Ontology in the Generalized Intelligent Framework for Tutoring: 2019 Update. Paper presented at the Proceedings of the 7th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym8).

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.

Goldberg, B., Brawner, K., & Hoffman, M. (2020). *The GIFT Architecture and Features Update: 2020 Edition.* Paper presented at the Proceedings of the 8th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym8).

Hoffman, M., Goldberg, B., & Brawner, K. (2021). *The GIFT Architecture and Features Update: 2021 Edition.* Paper presented at the Proceedings of the 9th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym9).

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

Robson, R., & Barr, A. (2018). Learning Technology Standards - the New Awakening. In R. Sottilare, K. Brawner, A. Sinatra, & B. Goldberg (Ed.). Proceedings of the Sixth Annual GIFT Users Symposium: US Army Research Laboratory. https://www.gifttutoring.org

Sottilare, R., Brawner, K. W., Goldberg, B. S., & Holden, H. A. (2012). The Generalized Intelligent Framework for Tutoring (GIFT).

Walcutt, J.J., & Schatz, S. (2019). Modernizing Learning: Building the Future Learning Ecosystem. Washington, DC: Government Publishing Office.

ABOUT THE AUTHORS

Michael Hoffman is a senior software engineer at Dignitas Technologies and the technical lead for the GIFT project. For over a decade he has been responsible for leading the engineering of GIFT, collaborating with the intelligent tutoring system (ITS) community, and supporting ITS related research. 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 Soldier Center and their contractors. He is also the Project Manager on the Flexible and Live Adaptive Training Tools (FLATT) project which is providing a new and intuitive way to leveraging GIFT and the technical lead on helping to integrate GIFT into TSS/TMT.

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.

The 2022 Instructor's Guide to the Generalized Intelligent Framework for Tutoring (GIFT)

Anne M. Sinatra
US Army Combat Capabilities Development Command (DEVCOM) – Soldier Center

INTRODUCTION

Welcome to the 2022 Instructor's Guide to the Generalized Intelligent Framework for Tutoring (GIFT). This is part of a series of guides that have been published since 2014. This series includes The Research Psychologist's Guide to GIFT (Sinatra, 2014; Sinatra, 2016; Sinatra, 2018; Sinatra, 2020), The Authoring Guide for GIFT (Sinatra, 2021), and the Instructor's Guide to GIFT (Sinatra, 2015; Sinatra, 2019). Each of these guides is written from a different perspective and serves the purpose of explaining how a person in that specific role could utilize GIFT, lessons learned, and suggestions for making improvements to GIFT that are specific to that area.

There have been a number of updates and approaches implemented in GIFT that serve as improvements to the instructor's experience when implementing GIFT as part of an existing in-person class. Additionally, the previous instructor's guides considered adaptive courseflows out of the scope, whereas this guide will discuss the adaptive courseflow, and how it can be utilized by an instructor.

Many of the lessons learned come from a practical application as part of a recent pilot study which involved creating adaptive lessons in GIFT. In this pilot, GIFT was utilized as an online component of an in-person class prior to the beginning of class attendance. While this guide will not delve into the details of the pilot study itself, there are a number of approaches covered in this guide that were utilized and were built out for our use case. The approaches that are relevant to instructors who wish to use GIFT in their classes will be covered in this guide.

WHAT IS GIFT, AND WHY WOULD I USE IT IN A CLASS?

GIFT is a domain-independent intelligent tutoring system (ITS) framework (Sottilare et al., 2017). GIFT has a number of different uses, including creating ITSs, and as a testbed for ITS research. While it includes adaptive tutoring functionality, GIFT can also be used to create courses that are linear for more traditional online learning, or non-ITS related research questions.

GIFT promotes reuse with the inclusion of authoring tools. The authoring tools have been designed to include drag-and-drop functionality, and a "courseflow" that shows the author the order that the elements they have selected will be experienced in by the learner.

An instructor can use GIFT for any topic area that they would like to create instruction for or include existing materials from. There have been GIFT tutors created in a variety of different topic areas including logic puzzles, military tactics, and marksmanship (Sinatra et al., 2016; Boyce, 2016; Goldberg et al., 2018). GIFT has both a desktop version and a Cloud version. The Cloud version allows for easy authoring access, as well as an easy way of distributing materials to learners.

GIFT is free, flexible, and provides a way to create adaptive tutoring and GIFT courses for your students to use. These GIFT courses can either be adaptive or linear, and since GIFT is highly flexible, you can decide how to design your courses, and how to implement it in your class.

USING GIFT IN A CLASS

Previous Instructor's Guides have highlighted using GIFT as not only a means for presenting material, but also for teaching about creating materials. However, due to the amount of recent updates that align with an instructor creating GIFT courses for use with a class, the focus of this guide will be on this more traditional implementation. If you would like to learn more about other ways of utilizing GIFT as a class assignment please review the previous two Instructor's Guides (Sinatra, 2015; Sinatra, 2019).

There are a number of different approaches that can be used to implement GIFT in a class, however, some of the more common approaches are:

- As materials provided prior to the course starting to either present an initial topic, or provide review (e.g., a review of general Research Methods I and II before a Research Methods III course).

- As activities that can be completed during a class session, potentially in a computer lab with an instructor present to answer general questions.

- As materials that can be used for review on the student's own time, and cover a topic that has been taught about in class, or that serves as a review before an exam.

- As a primary way of learning information in an online course, or a supporting lesson in an online course.

Each of the above approaches have their pros and cons. From a GIFT course design standpoint, it is important to approach the course creation knowing which way you are planning to implement GIFT. This will allow you to write and/or modify any of the informational supporting materials and instructions to the learners so that they are consistent with how you are implementing it. Instructions for your students can be authored in GIFT in the form of "information as text", or as part of specific GIFT course objects.

While constructing your GIFT course and understanding the choices and functionalities you can use is important, it is also very important to determine how you will share these courses with your class from a practical standpoint. As the majority of these are new functions and processes in GIFT, a large portion of the paper has been devoted to them.

COMMONLY USED GIFT TERMINOLOGY

For the purposes of this guide, the following GIFT terminology is defined below:

GIFT Course: An individual tutor in GIFT that is accessed based on a course tile when you login to your account. While there are current discussions about changing the terminology used, at this time, course refers to essentially an individual lesson.

GIFT Authoring Tool: This is where you will create your GIFT courses. There is drag-and-drop functionality and different course objects that can be used.

Survey Authoring System: You can create your own questions, and question banks using the survey authoring system.

Adaptive Courseflow: If you choose to use adaptive tutoring, you can utilize an adaptive courseflow course object. You will define the materials that will be shown for the primary materials as well as remediation.

Course Concepts: These can be defined by the author and are used for adaptive training, and question banks.

Question Bank: A question bank course object selects a defined number of questions per concept (with the desired level of difficulty). Question bank objects provide an assessment, but they do not always present the same questions or the questions in the same order.

Survey/Questionnaire: A series of questions that are always presented, and always presented in the same order. Concepts cannot specifically be associated with surveys/questionnaires at this time.

Learner: When discussing GIFT, we refer to the individual who is taking the course as a learner (instead of using the term student). The learner is engaging with the course, and the GIFT system is learning more about them which can be used for adaptation.

Extract Data: If you would like to extract the individual GIFT logs you can do so using this function. However, you will need to have a version of GIFT installed on your computer and place the logs in the appropriate folder for this to be done. For the purposes of this paper it is recommended that the online Create a Report functionality is used.

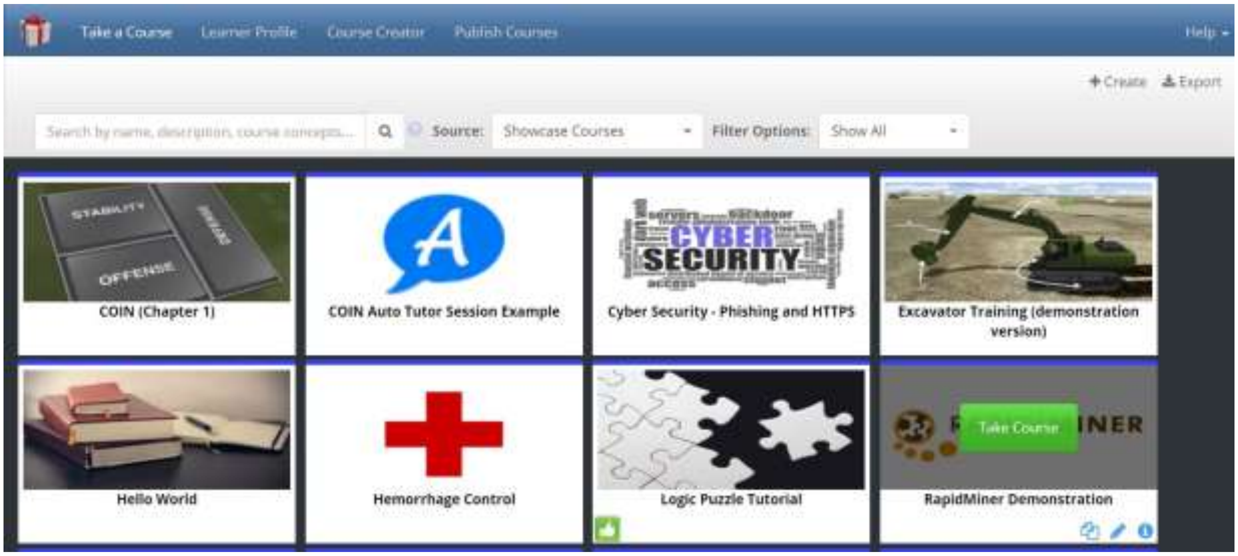
Create a Report: While there is currently no traditional gradebook in GIFT, the function that is closest to it is "create a report". By creating a report you will extract the data into a .CSV file that can be opened in Excel and saved as an Excel file. Further, if you have selected to do so, the data can be merged by each learner so their data is combined on one row like a gradebook.

CREATING YOUR GIFT COURSE

As last year's Authoring Guide to GIFT (Sinatra, 2021) heavily covered the course authoring process including adaptive courseflows and concepts, the specific discussion of course creation will be minimal in this specific guide. It is recommended that you review last year's authoring guide if you would like to see a worked example, and a step-by-step authoring approach (Sinatra, 2021). In this section, I discuss some basic functionality required for authoring GIFT courses, and some decisions instructors need to make as they develop their GIFT courses.

Logging into GIFT and Course Tiles

When you log into GIFT you will see a course tile page (see Figure 1 for an example). There will be Showcase courses that are available by default, and any additional GIFT courses that you create will be visible on this interface. The Showcase courses are existing courses that you can take as a learner, and that you can view in the Course Authoring Tool. You cannot directly edit these; however, you can copy them and edit those versions. When you hover your mouse over a specific course tile a green "Take Course" button will appear in the center of the tile, and in the bottom right corner you will see icons including a pencil. To open a specific course in the GIFT authoring tool you will click on the pencil. If you want to create a new course you can click on "Course Creator" at the top of the screen in the blue bar. You will be prompted to enter a course title, and then the Authoring Tools will open.



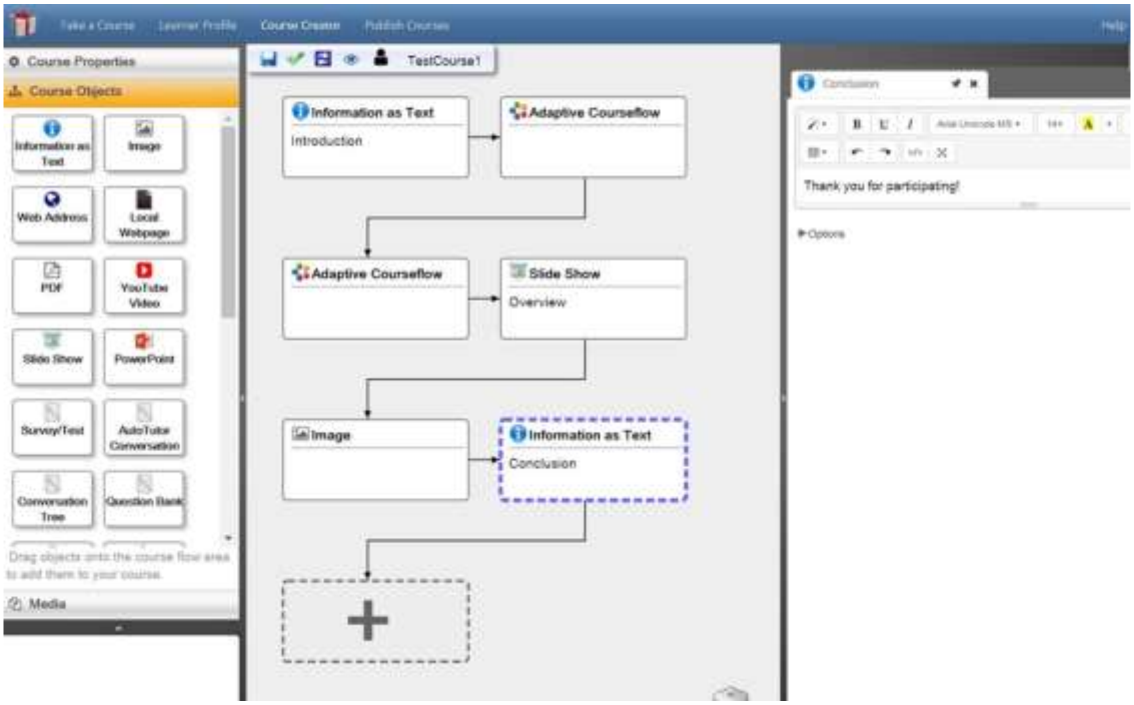
Source: Showcase Courses

Filter Options: Show All

Figure 1. GIFT Course Tiles

GIFT Authoring Tools

The GIFT Authoring Tools have a drag-and-drop interface. See Figure 2 for a screenshot of the Authoring Tools and an example of a GIFT course. On the left side of the screen there are different course objects that you can use in your course. Next to the course objects you will see your courseflow, which shows you the order that the course objects will be experienced in by the learner. If you select a specific course object the right side of the screen will show you the specific authored properties of that object. You can also move these within the courseflow using drag-and-drop functionality.



Drag objects onto the course flow area to add them to your course.

Thank you for participating!

Figure 2. The GIFT Authoring Tools and Courseflow

Adaptive or Linear Courses

One of the first choices you will need to make when deciding to create a GIFT course is if you would like to use adaptive tutoring or not. You can utilize GIFT's tools to create a linear course that is the same every time a learner interacts with it. Or you can choose to adapt such that the system is evaluating learners on their proficiency on certain course topics, and providing remediation if they do not perform as expected. A clear walkthrough of how to create a linear course can be found in the 2019 Instructor's Guide to GIFT (Sinatra, 2019), and a clear walk through of creating an adaptive course in GIFT can be found in the 2021 Authoring Guide for GIFT (Sinatra, 2021).

The simplest approach to creating adaptive tutoring in GIFT is to use the Adaptive Courseflow Object, which can be seen as a course object in Figure 2. The learner will experience the course object, but will not be able to move to the next course object without mastering it (passing the author defined proficiency). In order to use this, you will need to define concepts for your course, which is done in Course Properties (seen on the top left of Figure 2). More details about the authoring process can be found in the 2021 Authoring Guide (Sinatra, 2021). For each adaptive courseflow object you create you will define the concept(s) that will be covered; you will also provide overall content for the system to present that covers all of your concepts, author questions that can be used for remediation that cover your concepts, define how many questions you want to be presented to the learner/how many they need to get correct to move forward, and provide remediation materials for each concept. After receiving initial material, the learner will receive questions that cover the concepts, and it if they do not pass the defined number of questions they will receive remediation on the specific topic that they missed. They will then be asked questions again (these could be the same or different), and will continue receiving remediation/the process until they have passed all the concepts. There is more authoring involved with creating an adaptive course than a linear course, but it does provide remediation and an engaging learning experience.

Authoring Questions

You can provide questions to learners in the form of a survey/test, which will always provide the same questions in the same order. Or you can utilize a question bank which will display an author defined number of questions on different concept areas. In order to utilize the question bank course object you will need to define course concepts, and make sure that you align all of your questions to a concept. You author your questions using the interface in Figure 3. For each question you can choose to add a Tag, if you would like to specifically identify the answers to the question in your output data (this can be seen on the right side of Figure 3). There is both a writing mode and a scoring mode. To switch modes and define the number of points that should be received for each answer/the correct answer (and concepts if you are using the question bank course object) click the blue button on the top of the screen that says "Scoring Mode".

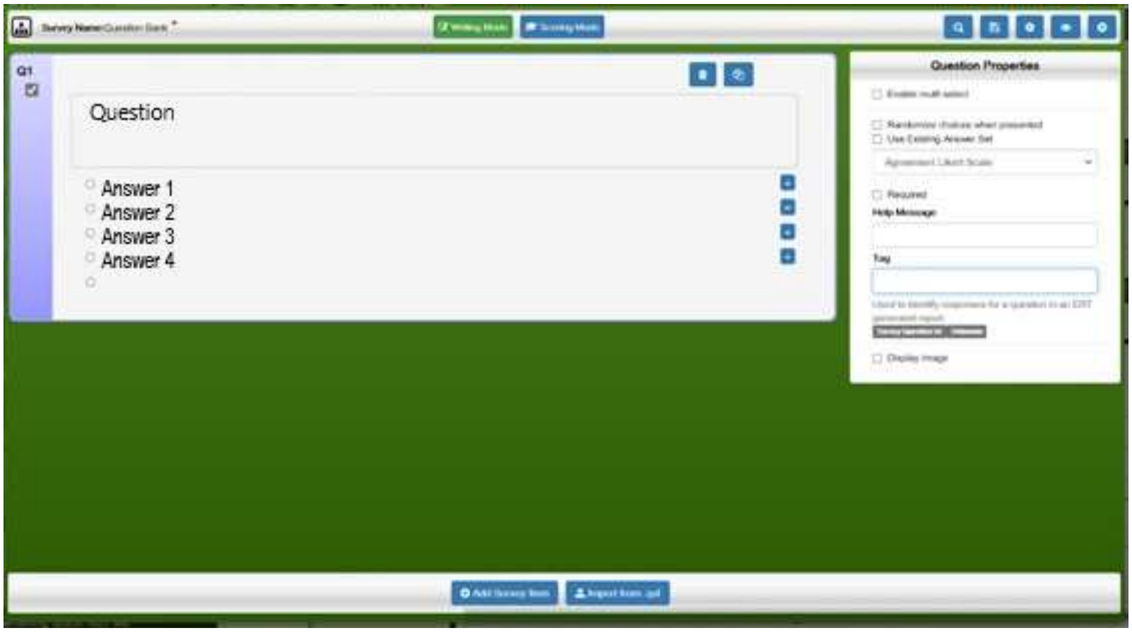


Figure 3. Question Authoring Interface

Published Course or Shared through GIFT

If you want to be able to track who has completed your course, and to limit who is able to complete the course to your specific class then you would share your course through GIFT. The specific instructions and considerations on how to do so are included in the next section. If you do not need to limit who accesses your course, and you either do not need to track who completed it, or want to create a self-report question that asks learners to enter their name, you can use the "Publish Course" functionality. The published course functionality is described in detail in the 2019 Instructor's Guide to GIFT along with how to extract the data from it.

HOW TO SHARE YOUR GIFT COURSES WITH YOUR CLASS

Login Process and Implementation

As identified in previous Instructor's Guides (Sinatra, 2015; Sinatra, 2019), GIFT does not currently have different interfaces for different user roles (i.e. Instructor, Student). While there still are not overall different login interfaces, the below functions have been added to move toward the needed functionality:

- Share a Course
- Permission Levels for Courses
- Create Report from an Authored Course

How to Set up an Instance of a Course for a Class

You will need to ask your students to create a GIFT account, and then to send you their user name. You will need to know a student's user name in order to share the course with them. On the main GIFT login screen there are a number of different course tiles; each of these are individual courses which can be taken. It is expected that there will be multiple GIFT courses that will be shared with your class. For each of the individual GIFT courses you will need to repeat the process that is listed in this section (sharing the course, setting permission levels, and creating data reports).

Share a Course and Permission Levels

While the Share a Course functionality has been previously established, the permission levels are new. Previously if a course was shared the individual could open up the editing interface and see how it was structured and the correct answers. The new permissions have an option that is aligned with the role of a student, which allows them to only take the course. This means that they can take the course, but they cannot view or edit it.

Permission Levels

There are three options of permissions:

- EditCourse: Can Edit, Copy, Export and Take
- ViewCourse: Can View, Copy, Export, and Take
- TakeCourse: Can Take

Roles that might use each of the permission levels:

- EditCourse: Primary course author and any co-authors
- ViewCourse: Course instructor (if this is a different person than the course author), Subject Matter Experts, Teaching Assistants
- TakeCourse: Students

EditCourse allows full editing of the course and if this permission is shared with an individual they could actually edit and change the course itself (it is not a copy). Therefore, Edit permission should be considered carefully before it is granted.

ViewCourse allows for the course to be exported/copied and then that version can be edited. It allows an individual to look at the content of the course, how it is authored, and the answers; however, it is Read-Only and they cannot change or impact the course itself. If you wanted to show the authored course to another individual so that they can check it for errors, review the content, or have an understanding of it you can use this type of sharing. Do be aware that they will have the ability to copy or export the course, so make sure it is someone who should have access to the answers and materials.

TakeCourse allows the individual to only take the course. They will not be able to open the course in the course authoring tool or see how it is structured or the answers. This is the most appropriate approach to use with a student in a class, as they will not be able to see the answers.

Share a Course Functionality

In order to share a course, you will select the icon with the person and plus sign on the course tile, as shown in Figure 4.

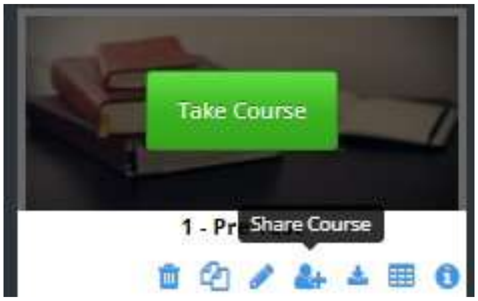
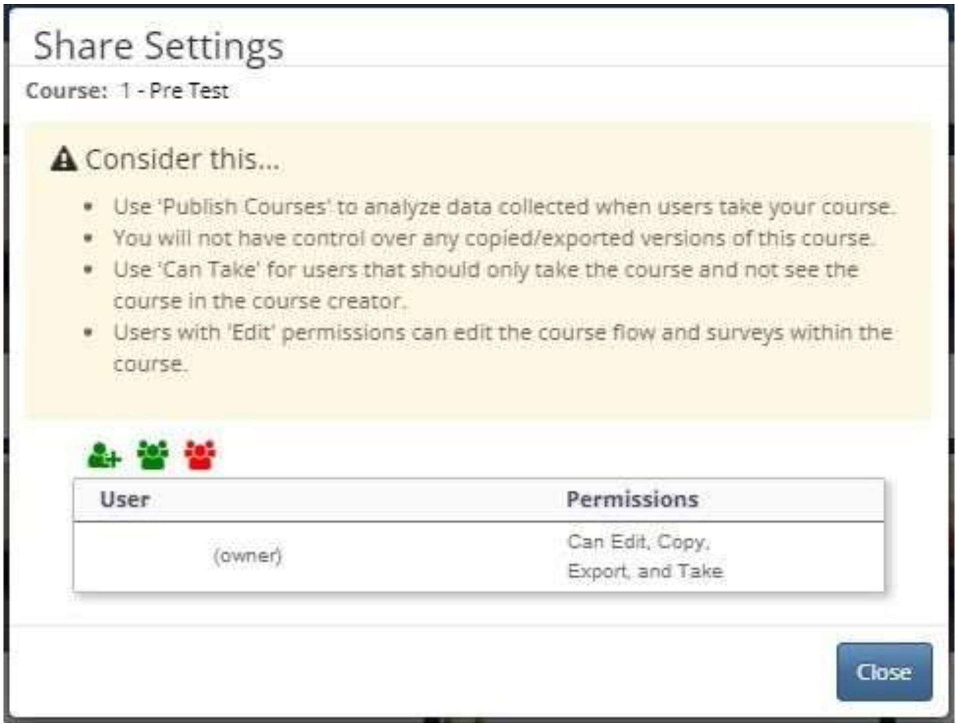


Figure 4. A course tile with the Share a Course Icon highlighted

Once you click on "Share Course", this will bring up a screen called "Course Settings" that can be seen in Figure 5. There is some information that is available to explain the options, and there are three icons: a green one with a person and a plus (add one user), a green one with three people (add multiple users), and a red one with three people (remove all users).



Share Settings

Course: 1 - Pre Test

⚠ Consider this...

- Use 'Publish Courses' to analyze data collected when users take your course.
- You will not have control over any copied/exported versions of this course.
- Use 'Can Take' for users that should only take the course and not see the course in the course creator.
- Users with 'Edit' permissions can edit the course flow and surveys within the course.

User	Permissions
(owner)	Can Edit, Copy, Export, and Take

Figure 5. Share Settings for a Course; The share icons are in green, and the unshare icon is red. There is a list of users with access to the course and their specific permissions.

If you select the add one user button you will see the interface in Figure 6. You will need to know the user's GIFT login name and enter it. You will also need to make your selection of what type of permissions you want them to have (as described in the previous subsection).

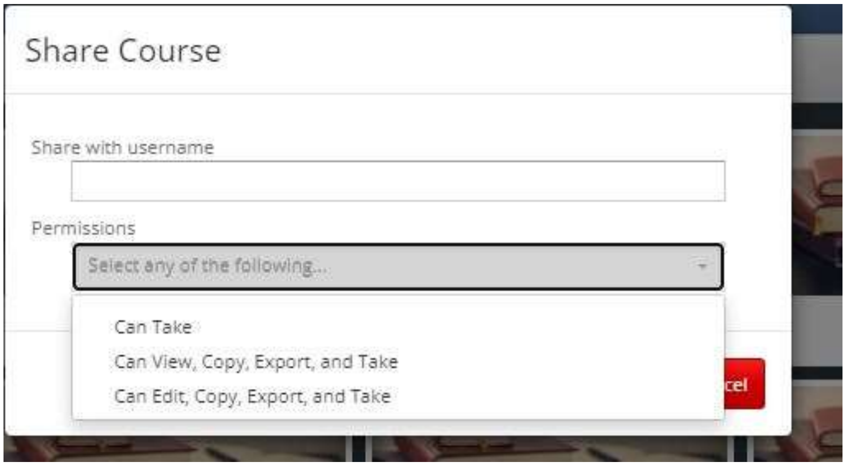
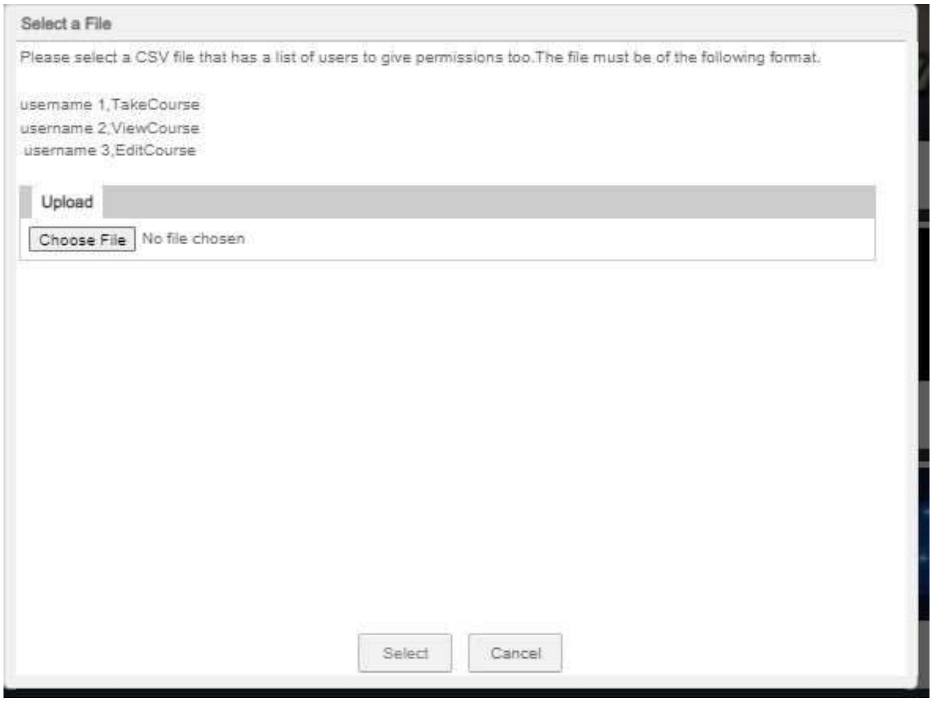

Share Course
Share with username
Permissions
Select any of the following...
Can Take
Can View, Copy, Export, and Take
Can Edit, Copy, Export, and Take
cel

Figure 6. Share a Course screen for a single user.

If you would like to add multiple users at the same time then you will need to create a .CSV file that includes all of the user names and each user's permission level. When you click on the icon to share with multiple users the interface in Figure 7 will be displayed. You will need to choose your .CSV file and upload it. The user roles will then be assigned, and the course shared.


Select a File

Please select a CSV file that has a list of users to give permissions too.The file must be of the following format.

```
username 1,TakeCourse
username 2,ViewCourse
 username 3,EditCourse
```

Figure 7. Share with Multiple Users interface.

You can create your .CSV file using Excel, but when you save it you will need to save it as a .CSV. For each row in Column A put the user name, and in Column B put the selected permission level (TakeCourse, ViewCourse or EditCourse). You will use a new row for each user name. After this is complete you will save it as a .CSV file, and upload it. If you want to add additional users you can create a new .CSV file that only includes those users and upload it (it will not remove the users that were previously uploaded).

HOW TO EXTRACT DATA AFTER YOUR CLASS HAS TAKEN A GIFT COURSE

Create Report from an Authored Course

The ability to export data from an authored course is a very important new feature that was recently implemented in GIFT. Previous to this feature, it was required that after creating a GIFT course it needed to be "Published", which provided a link that could be sent to an individual that you wanted to take the course. However, the "Published Course" process was more in line with what would be needed for an experiment; it did not require a login, therefore, it did not tie the data to any specific student as would likely be needed for it to be implemented in a classroom environment. Afterwards, in a Published Course, any data that was collected could be exported by the original course author, but it would not include any needed identifiers to link it to students in a class. Further, previously, if you were to run a GIFT course from your main course tile screen or share it, and someone ran it from their login, the course author would be unable to extract the data.

In the new implementation, the individual who authored a course and shared it, can now create a data report from the main course tile screen. A new symbol/icon has been added to the course tile that can be clicked to create a report based on the data; data can then be saved as a .CSV file and opened in Excel. See Figure 8 for an example of the course tile with the create report icon.

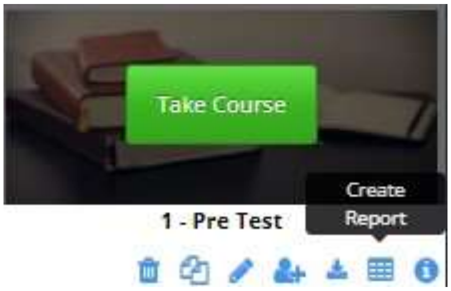
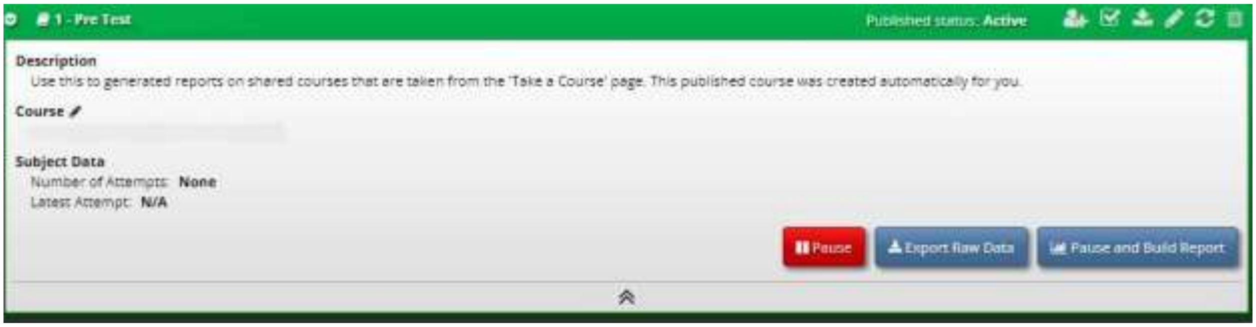


Figure 8. Create Report Icon on Course Tile

As this is a fairly new implementation, behind the scenes the system is treating this somewhat like a published course. When you click on the "Create Report" button it opens the Published Course interface and has created and opened a new version of the course which gives you the option of pausing data collecting, exporting raw data, and pausing and building a report. See Figure 9 for an example of what it looks like when you click the button. Note, if you also have a published version of the course you will need to carefully look at it to make sure you have selected the right one. A published version will have a URL listed, whereas, a Create Report version will not have a URL.



1 - Pre Test

Published status: Active

Description

Use this to generated reports on shared courses that are taken from the 'Take a Course' page. This published course was created automatically for you.

Course

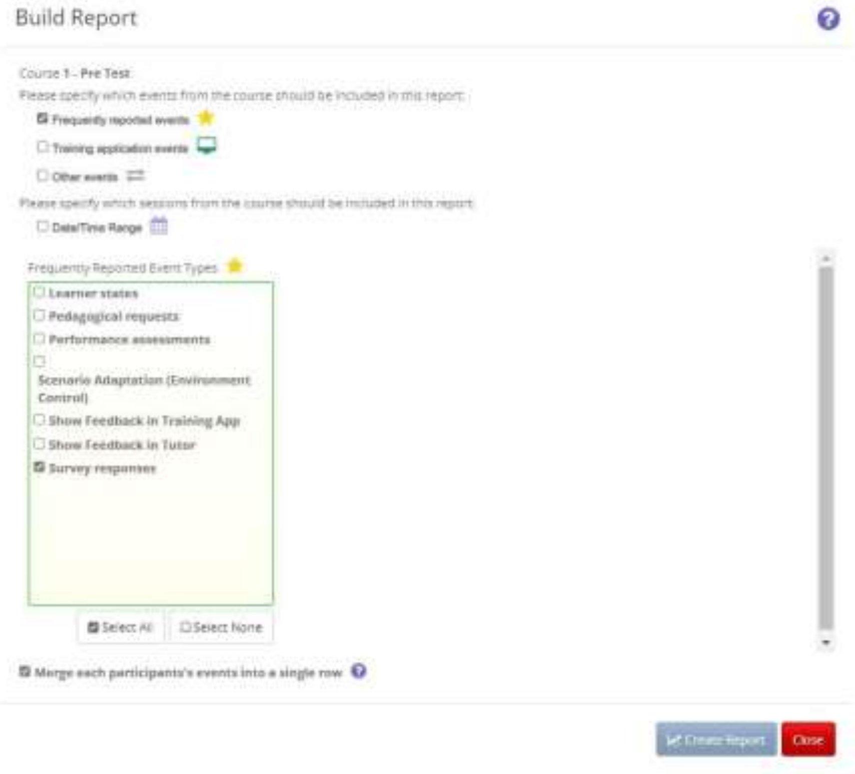
Subject Data

Number of Attempts: None

Latest Attempt: N/A

Figure 9. Second interface that is shown when creating a report.

Once "Pause and Build Report" is clicked another interface will be shown, which can be seen in Figure 10. For the purposes of this paper, and specifically for use as part of classes, the two items that are most likely to be needed to be checked are "Survey responses" and "Merge each participant's events into a single row". This will result in a .CSV file that is similar to a gradebook; each line will represent a different learner, and each column will represent a different score. Depending on how many surveys, question banks, or adaptive courseflows were included in the course, and if adaptivity was used the output sheet will either be fairly straightforward, or more complicated. In many cases there will be raw scores that will need to be added for an overall score. One current gap in GIFT is the lack of a gradebook interface.



Build Report

Course 1 - Pre Test

Please specify which events from the course should be included in this report:

☑ Frequently reported events

☐ Training application events

☐ Other events

Please specify which sessions from the course should be included in this report:

☐ Date/Time Range

Frequently Reported Event Types

☐ Learner states

☐ Pedagogical requests

☐ Performance assessments

☐ Scenario Adaptation (Environment Control)

☐ Show Feedback in Training App

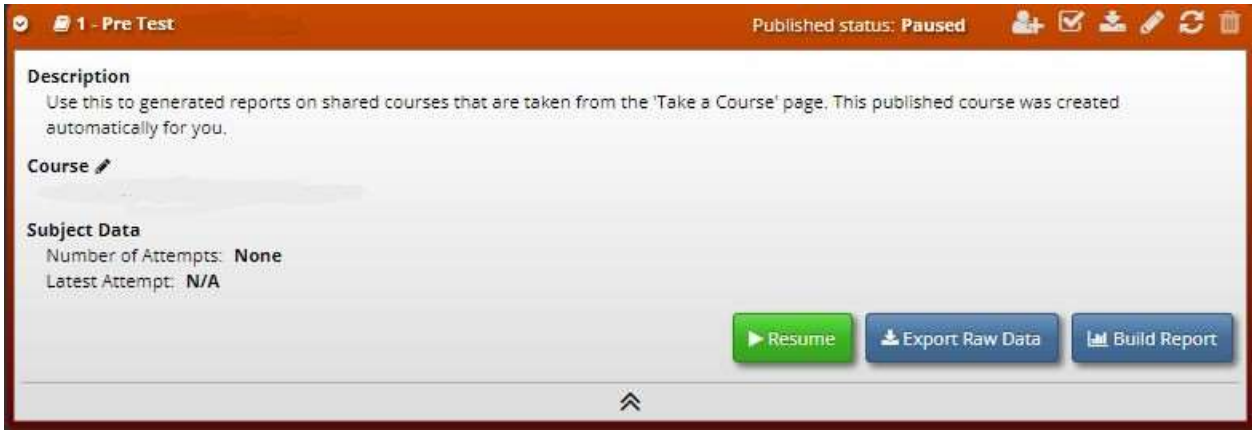
☐ Show Feedback in Tutor

☑ Survey responses

☑ Merge each participants's events into a single row

Figure 10. Screenshot of the Build Report Screen

After you have created your report, make sure to click "Resume" so that learners can take your course again. Otherwise they will receive a notice that they cannot currently access the course when they try to run it. See Figure 11 for a screenshot of the interface that is displayed after a report is generated.



Use this to generated reports on shared courses that are taken from the 'Take a Course' page. This published course was created automatically for you.

Course: [redacted]

Number of Attempts: None

Latest Attempt: N/A

Figure 11. After creating a report, the course is represented in the interface above. It is important to click "Resume" if you want learners to be able to continue taking the course.

If you have selected "Merge each participant's events into a single row" then you will be able to open the .CSV file in Excel and each row will represent a different learner. The columns on the top will represent the questions that they were asked. The headers will include any Tags or question names that you entered into the system. If you utilized surveys/questionnaires this should be relatively straightforward. If you used adaptive courseflows and question banks this will be a little more unique.

Understanding Adaptive Training Performance in the Data Output

GIFT outputs are still a large challenge from an instructor perspective. As noted above, the instructor will need to extract the data themselves and then examine an excel file. One important item to note is that it may be helpful to extract your data in two different ways. Clicking the "Merge each participant's events into a single row" will show you something that is most similar to a traditional gradebook, with one row representing each learner, and each column representing a question or score that was received. However, when utilizing adaptive training, and the adaptive courseflow object, sometimes only seeing the data from this perspective can be confusing or does not necessarily capture the pattern of the adaptive interaction. It may be beneficial to extract the data a second time without checking the merge button. This second approach will show a spreadsheet that has an indicator of what UserID (learner) each row of the spreadsheet belongs to. There will be additional rows based on the number of times that the learner experienced specific questions, which is an indicator that they received remediation. There will be a column for each of your defined concepts that tells you the overall score that the learner received each time through the adaptive courseflow on that concept. You can then determine how many times they missed specific concepts, and how many times it took them to go through the adaptive courseflow before they were successful in passing it. Ideally, a gradebook that calculates and provides this information to the instructor would be beneficial so that it does not need to be extracted and calculated manually.

RECOMMENDATIONS FOR FUTURE GIFT FEATURES

There are a number of recommendations that can help improve GIFT's functionality from an instructor perspective. Each Instructor's Guide has included a section like this one; some of the past recommendations have been implemented in GIFT, while others may be included below with additional new suggestions.

Updating naming within GIFT to be consistent with other systems

As noted in the terminology section of this paper, in GIFT terminology a GIFT course is a single course tile that has an authored sequence of course objects in it, and can cover as many concepts as the author wants it to. However, it generally needs to be completed in one sitting, as progress in the course itself is not saved, and to ensure it can be successfully completed it may be beneficial to cover a small amount of concepts in each course. Traditionally, those who do not work directly with GIFT might refer to this as a GIFT Lesson. The term lesson implying that it is one item that when used together with others represents a larger course. Updating the terminology for individual lessons and courses that have been created in GIFT will help make it more consistent with the mental models that instructors may have when they use the system (e.g., each tile is a lesson, not a course).

Linking GIFT lessons/courses together in an overall course

An additional approach to help with updating naming within GIFT, would be potentially to link together multiple lessons into an overall GIFT course tile. Currently a class might have 15 individual GIFT courses associated with it, and each would be viewed as a separate course tile. It might be beneficial to allow the ability to group GIFT lessons together so that students can understand the sequence items should be taken in, and the system can examine the performance on the specific linked lessons and use that to update the learner model for feedback/assessments in other linked lessons.

Gradebook Functionality

GIFT currently does not include an easy to use gradebook or visualization tool to see how students performed during GIFT courses. If the recommendation provided above of linking multiple GIFT courses into an overall course that is assigned to a specific instructor and is shared with a number of different students was implemented, it may be one step towards a gradebook. In the current form it is possible to get the desired information that is traditionally in a gradebook, but there are multiple steps that are taken, and it is not necessarily a straightforward process, which may limit adoption by instructors.

CONCLUSIONS

GIFT provides many benefits to instructors who want to use it as a part of their classes. The current guide provides some lessons learned and processes for using GIFT in its current state as an element of a class. If an instructor would like to create a GIFT course, I recommend using this guide as an overview for lessons learned in course set up, and combine it with last year's authoring guide (Sinatra, 2021). The GIFT team has been very responsive to suggestions that have been made previously on how to continue to improve an instructor's experience interacting with GIFT. It is my hope that I will be writing an additional instructor's guide in the near future that will further describe how to utilize new features that can help make the instructor's experience even better.

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.

REFERENCES

Boyce, M. W. (2016). From concept to publication-a successful application of using GIFT from the ground up. In Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym4) (p. 125).

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. (2019, May). The 2019 Instructor's Guide to GIFT. In *Proceedings of the 7th Annual GIFT Users Symposium* (p. 19). US Army Combat Capabilities Development Command–Soldier Center.

Sinatra, A. M. (2020, May). The 2020 Research Psychologist's Guide to GIFT. In *Proceedings of the 8th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym8)* (p. 23). US Army Combat Capabilities Development Command–Soldier Center.

Sinatra, A. M. (2021, May). The 2021 Authoring Guide for GIFT. In *Proceedings of the Ninth Annual GIFT Users Symposium (GIFTsym9)* (p. 25). US Army DEVCOM–Soldier Center.

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, 1-19.

ABOUT THE AUTHORS

Dr. Anne M. Sinatra *is a Research Psychologist at US Army Combat Capabilities Development Command (DEVCOM) – Soldier Center. Dr. Sinatra has created a number of different GIFT courses, and worked with GIFT since 2012. Her background is in Human Factors and Cognitive Psychology.*

THEME II:
AUTHORING TOOLS

Facilitating the Integration of Virtual Humans within GIFT

Robert Sottilare[1], Angela Woods[1], Nick Giranda[1], Matthew Bertrand[1], Eric Ortiz[1] and Brad Friedman[2]
[1]Soar Technology, Inc. and [2]US Army Futures Command

INTRODUCTION

As identified in a 2021 Direct to Phase II Small Business Innovation Research (SBIR) topic (A214-036), the US Army seeks an integrated virtual human (VH) - adaptive instructional system (AIS) capability. According to Sottilare and Brawner (2018), AISs are artificially-intelligent, computer-based systems that guide learning experiences by tailoring instruction and recommendations based on the goals, needs, and preferences of each individual learner or team in the context of domain learning objectives. AISs are a group of learning technologies that include intelligent tutoring systems (ITSs), intelligent mentors (recommender engines), and intelligent instructional media.

The anticipated benefits of an integrated VH-AIS includes enhanced trainee engagement and rapport, training effectiveness, and system usability. The design goal of such a VH-AIS system is to:

- train soldiers using realistic and dynamic virtual characters in virtual, augmented, and mixed reality environments
- reduce the time, cost and skill required to create adaptive instruction with integrated virtual characters

While there are several examples of successful commercial products and academic prototypes for both VHs and AISs, there are few proven processes to facilitate the integration of VHs within AIS frameworks. Many VH solutions provide only visualization of characters with scripted behaviors (e.g., gestures and communications) which by themselves are insufficient to fully engage trainees. Most VH solutions do not support mixed initiative dialogue, which is a conversation between two entities (in this case – a VH and a human learner) where control over the conversation is transferred from one entity to another (Novick & Sutton, 1997). Adding to the complexity, VH development often requires high-level skills (e.g., computer programming and instructional system design knowledge) to fully integrate within AIS authoring capabilities (e.g., course creation) and instructional processes.

This paper discusses the challenges, technical approaches and implementation of a US Army-sponsored SBIR project called the Mentoring for Optimized Training with Integrated Virtual Adaptive Tutoring Enhancements (MOTIVATE) project (Contract W900KK-21-C-0023). The goal of MOTIVATE is to benefit the warfighter by delivering a solution for a realistic, integrated VH interaction that is interoperable with ITSs and other AISs. MOTIVATE will provide realistic behaviors and interventions with individual learners to simulate one-to-one human tutoring experiences, and enhance Soldier training and performance during adaptive instruction by enhancing learner rapport. This capability will provide more engaging feedback and interactions to Soldier trainees by using VHs to activate both verbal and non-verbal communication pathways between learners and AISs in much the same way that expert individual human tutors capture a learner’s attention and guide them through the learning process.

The goal of MOTIVATE is to support the tutoring and mentoring of trainees, build learner rapport with VHs, and improve learner motivation and performance. VH development directly addresses a critical Army Synthetic Training Environment (STE) program technology gap to improve artificial intelligence (AI) and adapt training to provide more intuitive learning experiences to users, specifically adaptive training authors and individual learners.

MOTIVATE, GIFT AND VH CAPABILITIES

To develop an integrated VH-AIS under MOTIVATE, the design team identified candidate VH and AIS capabilities, and then built storyboards that described the interaction of components within the proposed integrated system. The AIS capability was selected first and it was critical that this AIS was open-source so the code could be easily modified and integrated with other MOTIVATE system components. It was also essential that this AIS was familiar to the Army and part of a training program baseline to facilitate transition. The AIS software architecture selected was the Generalized Intelligent Framework for Tutoring (GIFT; Sottilare et al., 2012; Sottilare et al., 2017). GIFT is an open-source science and technology prototype developed by the US Army that was recently transitioned and familiar to the Army STE program. Selecting GIFT meant that the MOTIVATE solution under development could be embedded within the GIFT baseline and readily used and evaluated by the STE technical team.

Candidate VH systems were also evaluated and included both prototype academic systems and commercial products. For the limited scope of the MOTIVATE project, our goal was to minimize the number of VH systems integrated with GIFT while also selecting complex and flexible capabilities that would enable us to identify interoperability standards and recommended practices that could facilitate the integration of future VH capabilities within GIFT. Two VH capabilities were considered. One commercial product, the Media Semantics People Builder (MSPB) was considered based on previous compatibility with GIFT. The second, an academic prototype, is the University of Southern California-Institute for Creative Technologies (USC-ICT) Virtual Human Toolkit (VHT; Gratch et al., 2013). VHT was selected based on the Army's familiarity and long investment in its development, its proven track record of application in various training and educational environments, and its ability to support natural language understanding and generation. In the next section, we explore the technical approach for an integrated VH-AIS in the MOTIVATE project.

TECHNICAL APPROACH

The design goal for the MOTIVATE project is to simplify the authoring processes within the GIFT Course Creator, and then to engage learners with VHs in every possible phase of adaptive instruction. The MOTIVATE technical approach also seeks to reduce the time and skill needed to integrate VHs within various phases of instruction (i.e., rules, examples, recall, remediation and practice) for any adaptive course conducted using GIFT.

To simplify the authoring processes and make it easier to integrate VHs, modifications to the controls within the Course Creator were deemed necessary. However, the MOTIVATE technical approach was also designed to keep the authoring processes largely unchanged in that a MOTIVATE-enhanced GIFT baseline still does not require the author to program, create new condition classes or specifically define every VH utterance during course creation. The priority is to begin with simple author-defined utterances and gradually emphasize more complex interactions. For some simple utterances, it was necessary to allow the author to determine the exact utterance for each of the GIFT instructional phases (i.e., rules, examples, recall, remediation, and practice). However, in more complex cases, an automated triggering mechanism for VH interventions will be implemented. These triggers will include the learner's state (e.g., performance, emotional, or workload) and the logic of the VHT used to generate tutor responses when they are understood.

The development work performed to date has concentrated on the integration of VHs as support mechanisms for learners. GIFT interventions are triggered by state assessments (e.g., competency or performance). So, pre-tests or competency statements can be used to initialize competency in GIFT. The competency level can then be used to trigger VH interventions just like any other intervention in GIFT. It is expected that VH interactions will increase with lower competency and performance scores since the VH role is to support learning and lower competency indicates the need for higher levels of scaffolding.

As noted earlier, part of the design process was to identify components, capture their interactions in storyboards of use-cases, and then examine those use-cases to drive changes to the GIFT Course Creator. We began by identifying interactions between components of GIFT, the VHT, and a virtual character embedded within GIFT's tutor-user interface (TUI) as shown in Figure 1 below. GIFT interventions are initiatives by the tutor to communicate with the learner or to modify the difficulty of the instructional content to match the capability of the learner per Vygotsky's (1987) zone of proximal development (ZPD). The ZPD measures the difference between what a learner is capable of doing without support, and what they can do with support from a more knowledgeable person with higher competency or expertise in the topic being instructed. GIFT interventions may take the form of:

- feedback to the learner on their performance or progress toward learning objectives
- direction to the learner about next steps in an adaptive course
- support or encouragement of the learner to persevere (demonstrate grit)
- prompts that request more information from the learner about a recent learning experience
- questions (pre-test, check on learning)
- changes to content difficulty based on recent learner performance or changes in competency

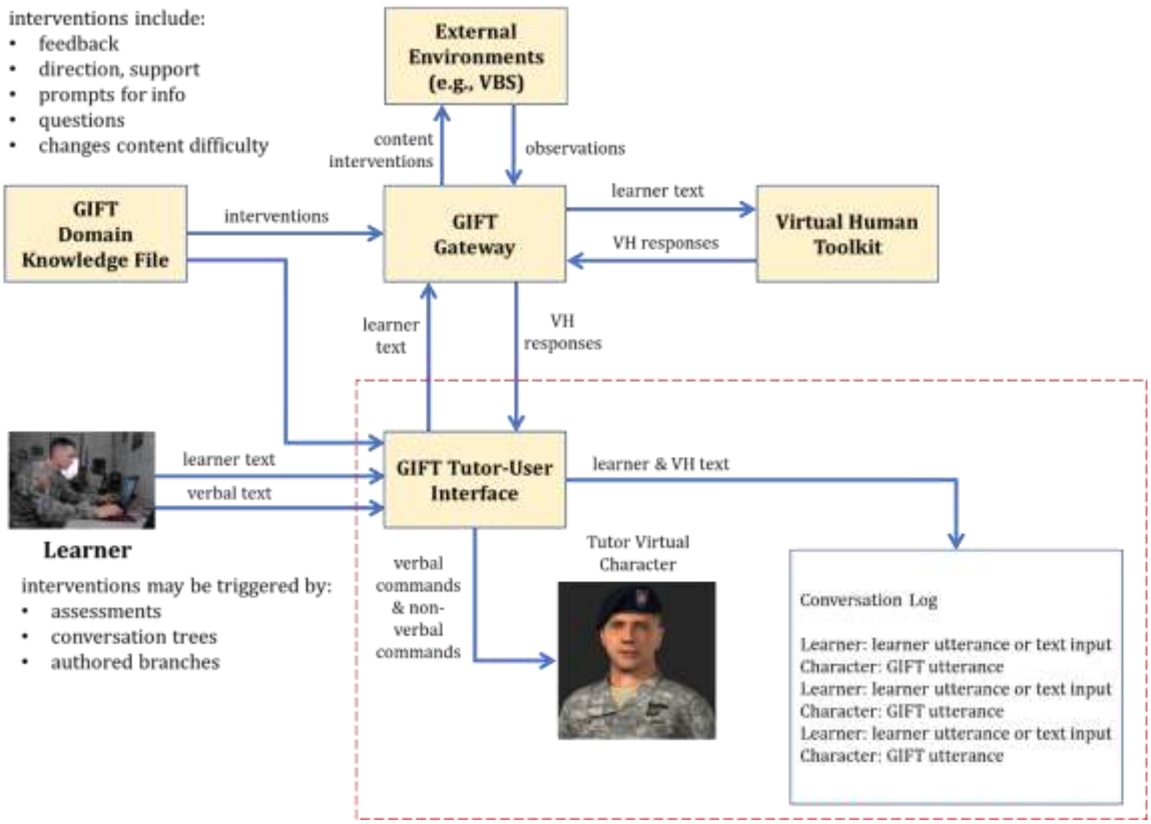

interventions include:
• feedback
• direction, support
• prompts for info
• questions
• changes content difficulty
External Environments (e.g., VBS)
content interventions
observations
GIFT Domain Knowledge File
interventions
GIFT Gateway
learner text
VH responses
Virtual Human Toolkit
learner text
VH responses
learner text
verbal text
GIFT Tutor-User Interface
learner & VH text
Learner
interventions may be triggered by:
• assessments
• conversation trees
• authored branches
verbal commands & non-verbal commands
Tutor Virtual Character
Conversation Log
Learner: learner utterance or text input
Character: GIFT utterance
Learner: learner utterance or text input
Character: GIFT utterance
Learner: learner utterance or text input
Character: GIFT utterance

Figure 1. GIFT-VH Interactions in MOTIVATE

In Figure 1, the TUI is composed of both the tutor portrayed by the virtual character and the conversation log that tracks the verbal interactions between the learner and the tutor. The TUI should be able to host various VH capabilities, but we chose to embed the VHT character initially. The technical team also facilitated message changes within GIFT to relay both verbal and non-verbal commands to the virtual character. To facilitate the design process, the technical team has identified three example use-cases to date that illustrate the interactions between GIFT and the two VH capabilities (VHT and MSPB) in the MOTIVATE architecture. We examine those use-cases in more detail below.

GIFT-Initiated Utterances

The first use-case we explored involves GIFT generating verbal utterances that provide analogous information to what GIFT provides now as textual information (written statements) in course objects such as the "information as text" object. Figure 2 illustrates the MOTIVATE process for use-cases involving GIFT-initiated utterances. The process is driven by configuration changes in the five instructional phases (i.e., rules, examples, recall, remediation, and practice) defined in an adaptive courseflow object. The technical team added a simple checkbox control to each of the phases to enable the inclusion of a VH utterance during that phase. In this use-case the author directly specifies the utterance. For example, if a real-time assessment of the learner's performance triggers a reflective intervention in the remediation phase (see step 1, Figure 2), then the author could specify a relevant utterance. This trigger and control would then direct the GIFT TUI through a GIFT message (see step 2, Figure 2) to verbally and non-verbally communicate with the learner (see step 3, Figure 2). To simplify the design, a default positive state is at the root of non-verbal communications delivered by the virtual character in the TUI. This default state includes nodding, smiling, and direct eye contact with the learner.

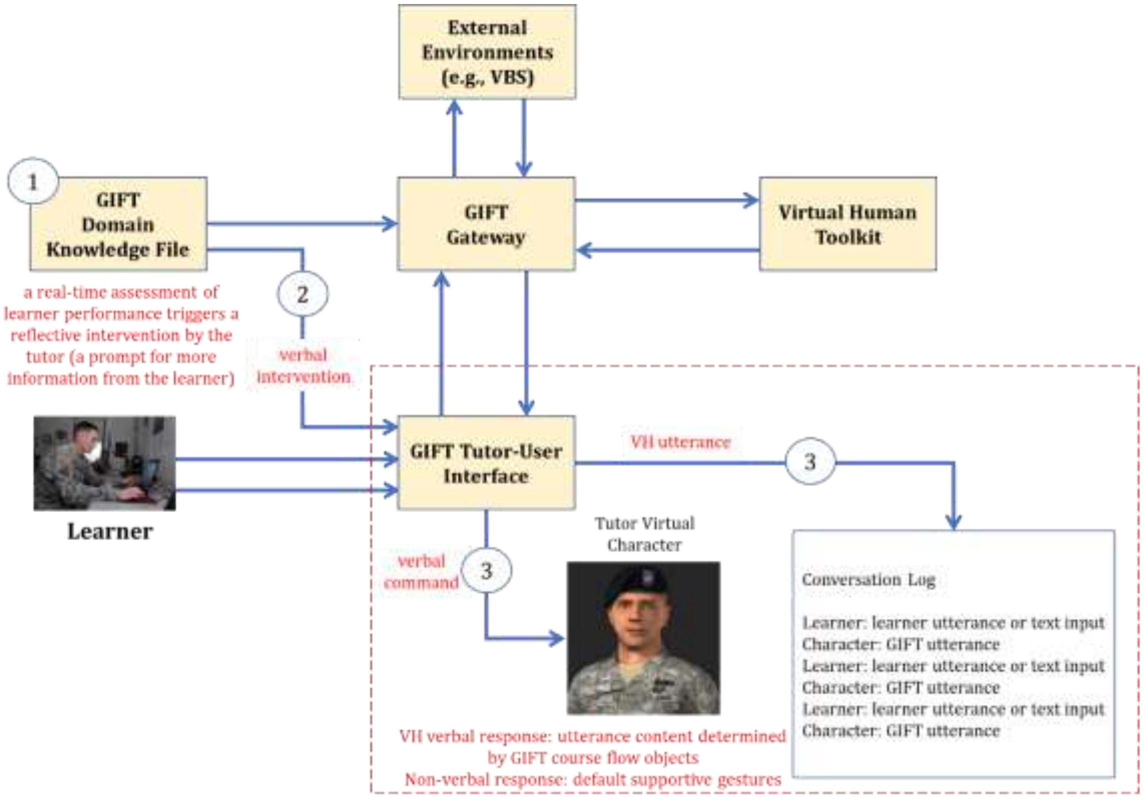
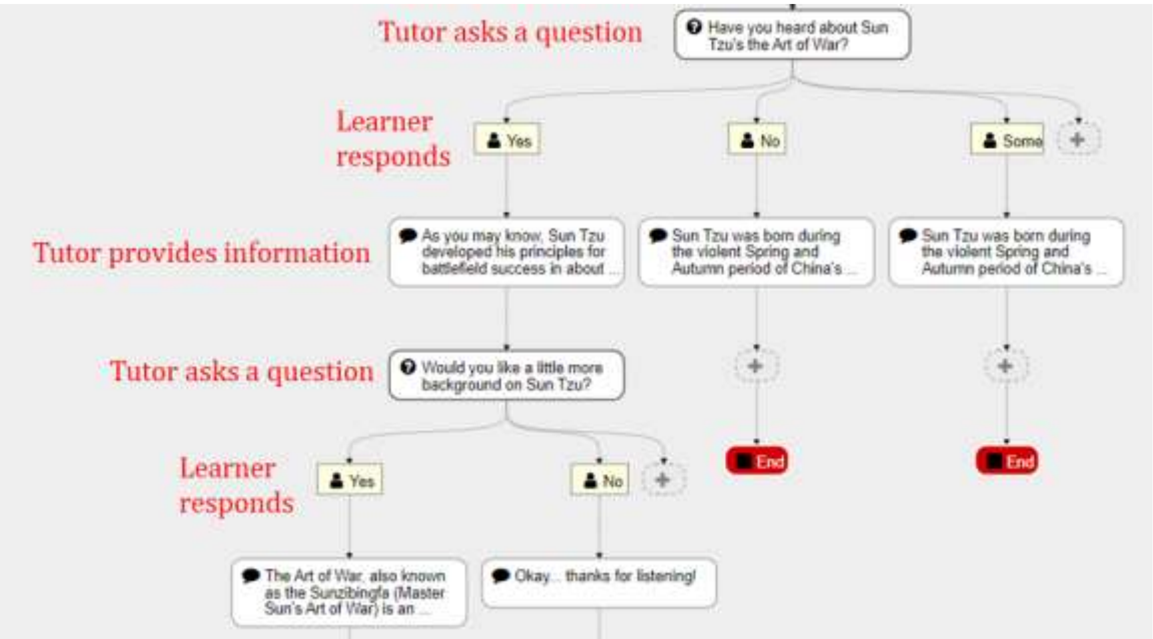

External Environments (e.g., VBS)
1
GIFT Domain Knowledge File
GIFT Gateway
Virtual Human Toolkit
2
a real-time assessment of learner performance triggers a reflective intervention by the tutor (a prompt for more information from the learner)
verbal intervention
Learner
GIFT Tutor-User Interface
VH utterance
3
verbal command
3
Tutor Virtual Character
Conversation Log
Learner: learner utterance or text input
Character: GIFT utterance
Learner: learner utterance or text input
Character: GIFT utterance
Learner: learner utterance or text input
Character: GIFT utterance
VH verbal response: utterance content determined by GIFT course flow objects
Non-verbal response: default supportive gestures

Figure 2. GIFT-VH Interactions during a GIFT-Initiated Utterance

Specifically, the design goal is to provide a context-dependent communication during each adaptive courseflow object phase by presenting content to the learner verbally, non-verbally (gestures), and textually (in writing). The conversation log will capture interventions that can be reviewed by either the learner or the GIFT tutor during the after-action review. Next, we discuss a use-case for interactive conversations with turn-taking between the tutor and the learner.

GIFT-Initiated Conversations



Have you heard about Sun Tzu's the Art of War?

As you may know, Sun Tzu developed his principles for battlefield success in about ...

Sun Tzu was born during the violent Spring and Autumn period of China's ...

Sun Tzu was born during the violent Spring and Autumn period of China's ...

Would you like a little more background on Sun Tzu?

The Art of War, also known as the Sunzibingfa (Master Sun's Art of War) is an ...

Okay... thanks for listening!

Figure 3. Sample Conversation Tree in GIFT

A second use-case involves the mining of conversation trees which are a type of course object within GIFT (Figure 3). This example is more complex than the simple individual GIFT utterances discussed in the last section in that this use-case involves the execution of a conversation tree. The tree drives interventions (e.g., questions, prompts, directions) by the tutor and the learner responds to these interventions as the tutor and the learner take turns communicating. The design of GIFT-initiated conversations will enable the learner to communicate verbally, textually through the conversation log in the TUI, or through drop-down menu selections.

Similar to the previous use-case, a real-time assessment (step 1, Figure 4) triggers the activation of a selected conversation tree which sends messages to stimulate the virtual character in the TUI. However, instead of only a single message, multiple messages are sent to the TUI as the tutor alternately delivers verbal, textual, and non-verbal communications (step 4, Figure 4), and then awaits a response from the learner (step 3, Figure 4).

To implement the GIFT-initiated conversations use-case required the development of new GIFT messages to transmit interventions from the domain knowledge file (DKF) conversation tree course object (e.g., questions, prompts or information) to the TUI, and additional messages to transmit learner selections back to the DKF to determine the pathway selected within the conversation tree and the selection of the next action to be taken by the tutor.

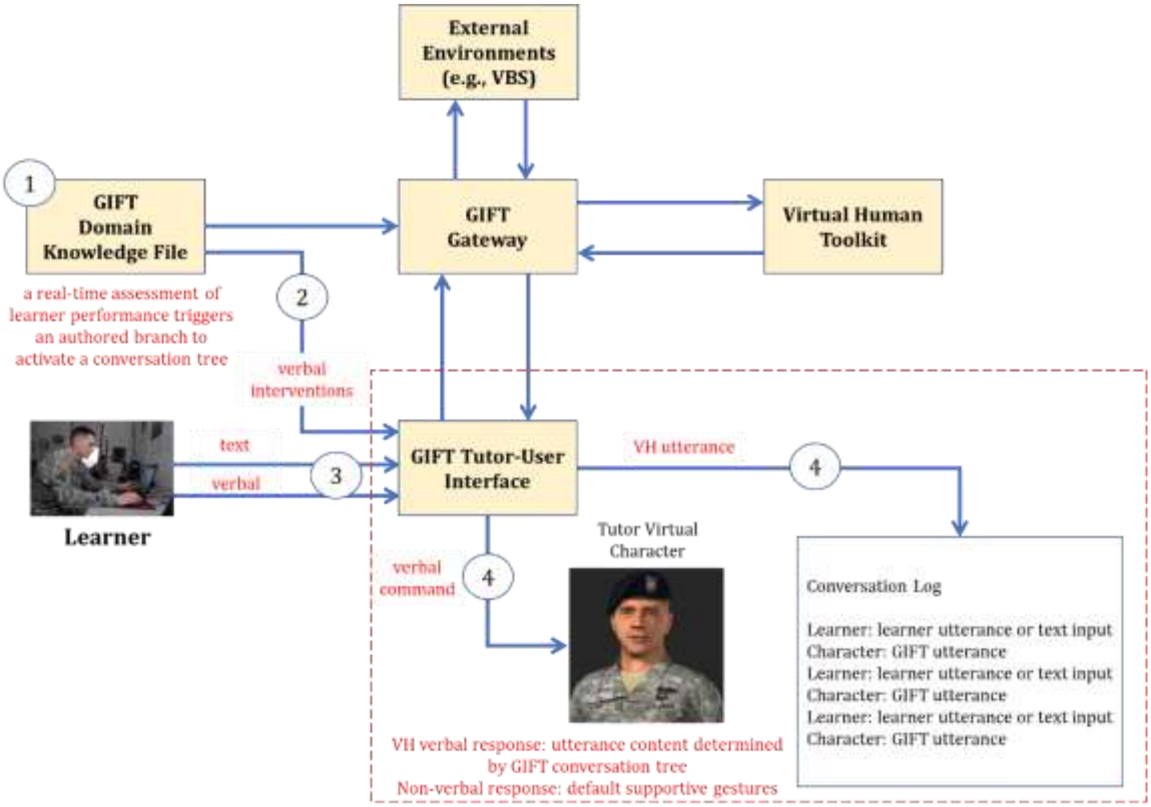

External Environments (e.g., VBS)
1
GIFT Domain Knowledge File
GIFT Gateway
Virtual Human Toolkit
a real-time assessment of learner performance triggers an authored branch to activate a conversation tree
2
verbal interventions
text
verbal
3
Learner
GIFT Tutor-User Interface
VH utterance
4
verbal command
4
Tutor Virtual Character
Conversation Log
Learner: learner utterance or text input
Character: GIFT utterance
Learner: learner utterance or text input
Character: GIFT utterance
Learner: learner utterance or text input
Character: GIFT utterance
VH verbal response: utterance content determined by GIFT conversation tree
Non-verbal response: default supportive gestures

Figure 4. GIFT-VH Interactions during a GIFT-Initiated Conversation

Learner-Initiated Utterances

The third used-case that we examined focused on the support of mixed-initiative dialogue in GIFT. Again, mixed initiative dialogue is about transferring conversational control between the participating entities (Novick & Sutton, 1997). In this specific use-case, we focused on how the learner might initiate or control a conversation with the tutor in GIFT. This use-case comes with some additional complexities in that GIFT must be able to understand any learner utterances and then select or generate an appropriate response. To accomplish this task, GIFT will translate speech-to-text and then analyze the text. The design team investigated various methods to enable natural language processing (NLP) of learner utterances in text form and natural language generation (NLG) to select appropriate tutor responses. Some of the most common methods for NLP are described below along with a short description of how they might be applied in an AIS architecture.

- Sentiment Analysis is the "use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information" (Wikipedia, sentiment analysis, 2022); sometimes referred to as subjectivity analysis, opinion mining, and appraisal extraction; in AISs, sentiment analysis might be useful in assessing learner stress, frustration, boredom, anger or other emotions that negatively influence learning.
- Named Entity Recognition (NER) is "a subtask of information extraction that seeks to locate and classify named entities mentioned in unstructured text into pre-defined categories such as person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc." (Wikipedia, named entity recognition, 2022); sometimes referred to as entity identification, entity chunking, and entity extraction; in AISs, NER can automatically scan learner utterance text, identify keywords within them and categorize utterances using relevant tags.

- Text Summarization is “the process of shortening a set of text data computationally, to create a subset that represents the most important or relevant information within the original content” (Wikipedia, text summarization, 2022); in AISs, text summarization may be used to breakdown scientific, medical, and technical jargon into its most basic terms in order to make it more understandable.

Since NLP and NLG are not embedded capabilities within GIFT, we selected the VHT (Figure 5) to perform these functions. This meant that we also need to design a GIFT gateway module to enable transmission and receipt of messages between VHT and GIFT that can be used to receive text-based utterances and stimulate the virtual character in the GIFT TUI to respond to the learner’s utterance. The VHT uses speech recognition to convert verbal utterances to text and then uses a classifier to determine the meaning of the learner’s utterance. Next an agent determines how to respond to the learner’s utterance.

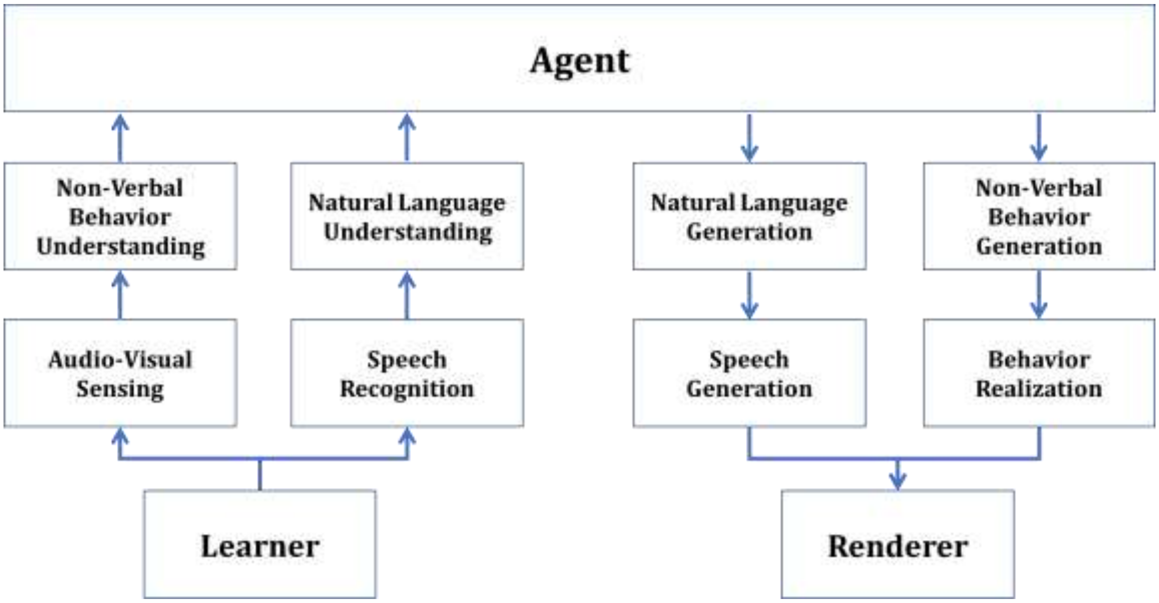

Agent
Non-Verbal Behavior Understanding
Natural Language Understanding
Natural Language Generation
Non-Verbal Behavior Generation
Audio-Visual Sensing
Speech Recognition
Speech Generation
Behavior Realization
Learner
Renderer

Figure 5. USC-ICT Virtual Human Toolkit Software Architecture (Hartholt, 2021)

In previous projects, USC-ICT used a question/answer agent, a rule-based action selector, and a corpus linguistics methodology that involves computer-based empirical analyses (both quantitative and qualitative) of language use by employing large, electronically available collections of naturally occurring spoken and written texts or corpora. To simplify the initial design of MOTIVATE and to reduce the required workload and skills required to author a VH within GIFT, the technical team opted to use a simpler mapping of text utterances to appropriate responses. The text and the mapping can be easily changed by GIFT course authors to enable transition from one domain of instruction to another.

Figure 6 identifies the flow of information within the architecture for the learner-initiated utterance use-case. First, the learner initiates a verbal or textual utterance (step 1, Figure 6) which is captured as a text-based message and relayed to the GIFT gateway (step 2, Figure 6) and then routed to the VHT (step 3, Figure 6). The VHT processes the text, classifies it and determines the appropriate verbal and non-verbal responses (step 3, Figure 6) and routes a text-based message to the GIFT gateway which in turn routes it to the TUI where the virtual character renders the speech and non-verbal behaviors.

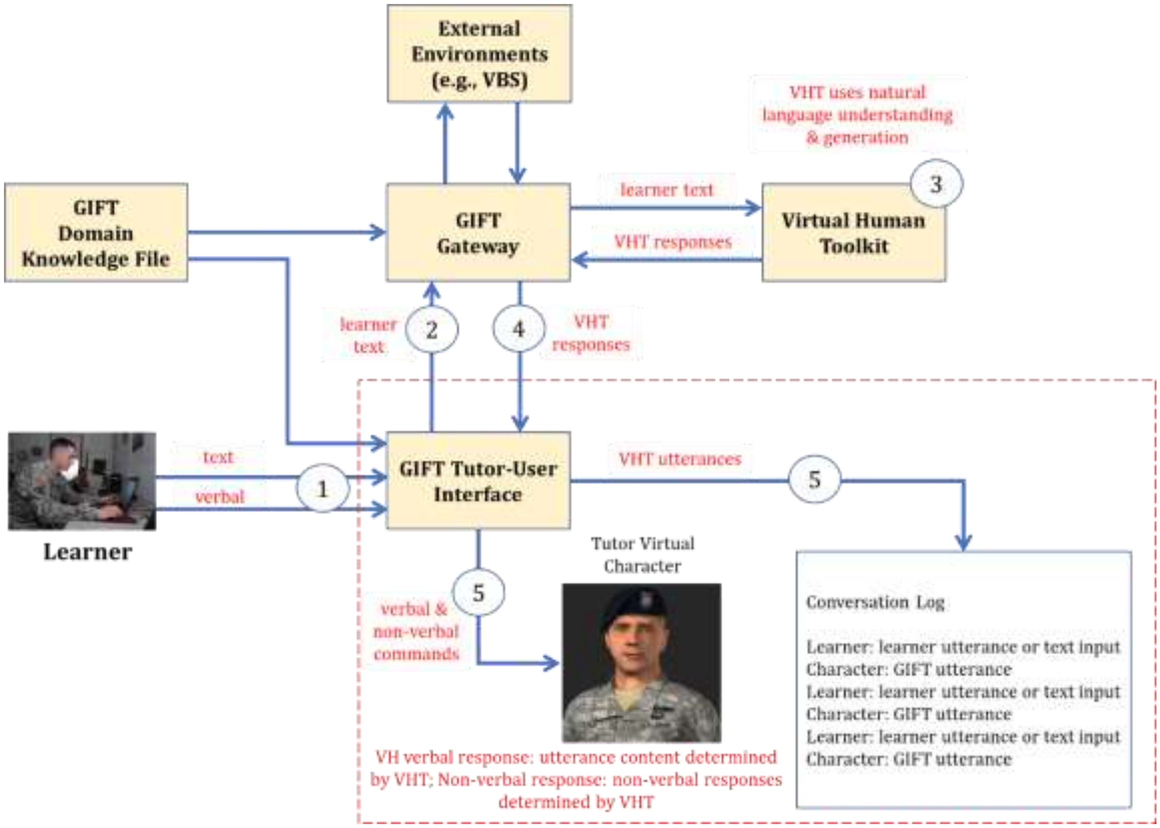

External Environments (e.g., VBS)
VHT uses natural language understanding & generation
GIFT Domain Knowledge File
GIFT Gateway
learner text
VHT responses
3
Virtual Human Toolkit
learner text
2
4
VHT responses
Learner
text
verbal
1
GIFT Tutor-User Interface
VHT utterances
5
Tutor Virtual Character
5
verbal & non-verbal commands
Conversation Log
Learner: learner utterance or text input
Character: GIFT utterance
Learner: learner utterance or text input
Character: GIFT utterance
Learner: learner utterance or text input
Character: GIFT utterance
VH verbal response: utterance content determined by VHT; Non-verbal response: non-verbal responses determined by VHT

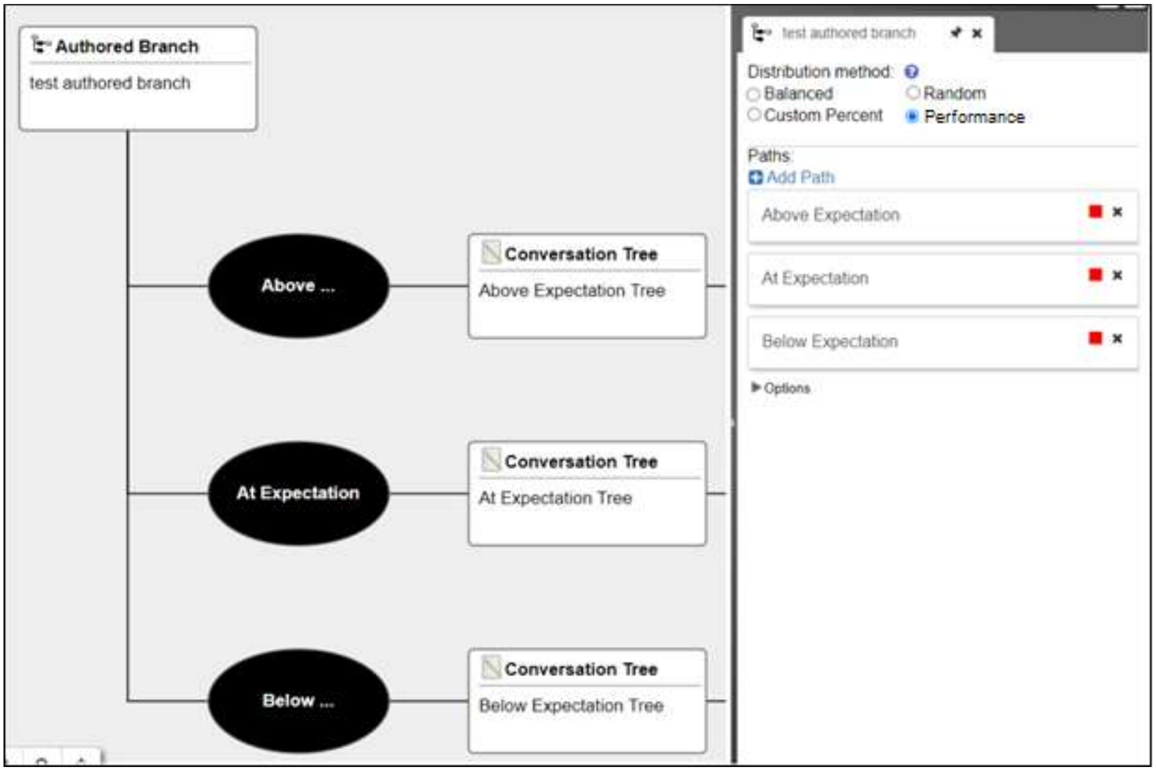
Figure 6. GIFT-VH Interactions during Learner-initiated utterances

Next, we discuss challenges associated with implementing a full VH capability that is compatible with GIFT.

CHALLENGES AND LIMITATIONS

Based on the technical approach described above, MOTIVATE, once fully implemented, will allow adaptive course authors to integrate and control VHs internally within the GIFT TUI and externally in Unity-based environments integrated with GIFT through the GIFT gateway. There are several challenges identified within the MOTIVATE project scope and some have already been addressed. As of the submission date of this paper, the MOTIVATE design team has identified methods for integrations within the adaptive courseflow object and its five phases of instruction (rules, examples, recall, remediation and practice). For example, simple checkboxes and VH media selections are used in a similar fashion as other types of media that are implemented in current GIFT courses. Another authoring simplification is the use of defaults for driving non-verbal behaviors of VHs within the TUI. A VH implementation that uses a commercial virtual character in the TUI uses default gestures like head nods and smiling to provide encouragement to the learner. The team has also begun to identify the content, format, and routing of GIFT messages for the integration of external Unity-based environments and the stimulation of virtual character behaviors in those environments.

The initial focus of MOTIVATE was to provide new controls within the existing GIFT authoring process in Course Creator to make it easy to assign VHs to any of the phases within a GIFT course. A major challenge is to implement VHs within the authoring process while still maintaining the original process as much as possible. The GIFT-initiated use-cases discussed in the technical approach section of this paper are triggered by assessments of the learner's performance. To provide additional flexibility to GIFT course authors, the technical team redesigned the authored branch course object to add a new pathway that is dependent upon the learner's assessed performance (Figure 7). This was necessary since the current authored branch course object provides three methods of distribution (balanced, random, or custom percent) to select one of three pathways, but none of the paths are dependent on the learner's performance. The new learner performance pathway will examine whether the learner's performance is below, at, or above expectations and route the learner to appropriate content aligned with their performance as noted in Vygotsky's ZPD (1987).



Distribution method:
○ Balanced ○ Random
○ Custom Percent ● Performance

Paths:
Add Path
Above Expectation
At Expectation
Below Expectation
▶ Options

Figure 7. Modified GIFT authored branch with added performance distribution

Additional challenges lie ahead for the development of a guided authoring process, mixed initiative dialogue, and the ability to take advantage of all the capabilities within the VHT. A major constraint is to be able to integrate VHs in GIFT courses without expecting adaptive course authors to program. The primary challenge in the MOTIVATE design is to provide a guided authoring process and reduce the manual steps in that process through the selection of default states. The intent is to design an authoring mentor to identify steps in the end-to-end authoring process so that GIFT can support the author by recognizing next steps and ultimately validating the completeness of the course. For example, it will be important to ensure that content has been aligned with each of the GIFT concepts (learning objectives).

Another major challenge is the ability to support mixed initiative dialogue that is more reminiscent of AutoTutor, a conversational tutor (Graesser et al., 2012) while still maintaining learning efficiency reminiscent of military instruction. To implement a VH to participate with the learner in a dialogue governed by a GIFT conversation tree is a challenge, but implementing VHs within GIFT courses using the VHT to understand and generate dialogue is much more complex. Given project resources, we determined our optimum approach should take advantage of the VHT capabilities available. While VHT and AutoTutor both have conversational capabilities with NLP, AutoTutor has none of the sensing capabilities contained within the MultiSense module in the VHT. The VHT uses MultiSense to process a variety of sensor data to interpret the visual and aural scene including learner behaviors, and then uses that information to drive decisions about learner interventions.

To exploit the VHT, the MOTIVATE design will require either the complex development of a corpus to represent terms and phrases within a domain of instruction or the simple mapping of recognized, frequently used phrases to appropriate responses. The first requires significant expertise and effort to develop a corpus that will cover any domain-relevant utterances by the learner. The second limits the utterances of the learners to a select group that is easily mapped to suitable replies. This requires less skill and is likely to be less work, but is more prescriptive and less flexible as the domain expands or authors attempt to migrate one domain corpus to other related, but different domains of instruction (e.g., dismounted soldier tactics and armor tactics). The team will also investigate methods to auto-generate corpora in various domains based on available textual data (e.g., field manuals), but it may not be possible to implement this under the current MOTIVATE project.

Now that we have addressed challenges, we move on to limitations. The primary limitations of the current MOTIVATE approach are driven by the scope of the Phase II SBIR project and its technical objectives. MOTIVATE currently focuses on individual learners and adaptive course authors who are guided through the course creation process. The current MOTIVATE approach could be greatly improved through the use of automation to reduce the tasks and workload of the author, and by expanding VH interventions to be multicast to support collaborative or team learning activities as discussed in next steps.

NEXT STEPS

To date, VHs have been integrated in GIFT as a mechanism to deliver feedback or interact with the learner using dialogue to coach or guide learning. Future use cases will apply VHs as actors (e.g., non-player characters) within a simulation or serious game environment external to GIFT. The major challenge is to integrate the VH as an agent to enable observations within the external environment to drive VH decisions and actions that support GIFT learning objectives.

Additional next steps for the current MOTIVATE project are to 1) investigate additional use-cases to support expanded VH automaticity using the VHT, 2) explore alternatives to the VHT and commercial MSPB, 3) design/develop a VH-led after action review (AAR) process that captures multiple learner assessments during an team-based adaptive course, 4) develop recommendations for new/modified GIFT messages based on MOTIVATE implementation, 5) design/develop/modify GIFT gateway for the VHT and external Unity environments, 6) model VHs as military personnel and make other changes to enhance learner engagement, and 7) research the need to fuse/analyze learner data from multiple sources using Massive Online Analysis (MOA), open source framework for data stream mining, in order to support more accurate learner assessments and VH integrations.

Potential next steps for an enhanced MOTIVATE project in the future could include recommendations for more automated processes and the ability to extend VH interactions to teams where multiple learners are working collaboratively to accomplish a set of goals or assigned tasks. Specific recommendations for an enhanced MOTIVATE capability are:

- Automated inclusion of VHs in GIFT courses – set defaults for including VHs in various phases of instruction
- Authoring templates and guided processes – reduce author workload and track progress of content development associated with VH integration to ensure complete adaptive training courses
- Team Interventions and Conversations – extend VH interventions and conversations from individual learners to collaborative learning groups
- AAR processes – expand structured (report-like) AAR processes based on conversation logs and learner actions, and eventually realize fully automated AARs with VHs
- VHs in augmented and mixed reality environments – extend current VH capabilities in virtual reality to support training in augmented or mixed reality training environments such as the Microsoft Integrated Augmentation Environment as the head-mounted display for the Army's IVAS

- Corpus development and integration processes – expand rudimentary mixed initiative dialogue with limited trainee utterance selections to support natural language understanding and generation, and auto-generation of domain corpora based on available sources; investigate the Natural Language Toolkit (NLTK), and open-source (NLP software tool powered with Python NLP to support domain corpora development from text or audio sources
- GIFT-Unity interoperability standards – extend the processes and establish standards and recommended practices for integrating Unity environments with GIFT to drive communication behaviors of VH external to GIFT; extend ability for artificially intelligent agent capabilities to drive non-communications behaviors in external environments (e.g., maneuvers or weapons engagements)

ACKNOWLEDGEMENTS

The research described herein has been sponsored by the U.S. Army Futures Command (AFC) under contract W900KK-21-C-0023. Statements and opinions expressed in this paper do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

REFERENCES

Graesser, A. C., D'Mello, S., Hu, X., Cai, Z., Olney, A., & Morgan, B. (2012). AutoTutor. In *Applied natural language processing: Identification, investigation and resolution* (pp. 169-187). IGI Global.

Gratch, J., Hartholt, A., Dehghani, M., & Marsella, S. (2013). Virtual humans: a new toolkit for cognitive science research. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 35, No. 35).

Hartholt, A. (2021). Virtual Human Toolkit Tutorial. University of Southern California – Institute for Creative Technologies (USC-ICT).

Novick, D. G., & Sutton, S. (1997, March). What is mixed-initiative interaction. In *Proceedings of the AAAI spring symposium on computational models for mixed initiative interaction* (Vol. 2, p. 12).

Sottilare, R. A., Brawner, K. W., Goldberg, B. S., Holden, H. K. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). Concept paper released as part of GIFT software documentation. *U.S. Army Research Laboratory— Human Research & Engineering Directorate (ARL-HRED)*, Orlando, FL, USA.

Sottilare, R., Brawner, K., Sinatra, A., Johnston, J. (2017). An Updated Concept for a Generalized Intelligent Framework for Tutoring (GIFT). *US Army Research Laboratory*, Orlando, FL, USA.

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

Vygotsky, L. (1987). Zone of proximal development. *Mind in society: The development of higher psychological processes*, 5291, 157.

Wikipedia contributors. (2022, February 21). Named-entity recognition. In Wikipedia, The Free Encyclopedia. Retrieved 18:22, April 12, 2022, from https://en.wikipedia.org/w/index.php?title=Named-entity_recognition&oldid=1073163208

Wikipedia contributors. (2022, February 6). Sentiment analysis. In Wikipedia, The Free Encyclopedia. Retrieved 18:13, April 12, 2022, from https://en.wikipedia.org/w/index.php?title=Sentiment_analysis&oldid=1070318812

Wikipedia contributors. (2022, April 3). Automatic summarization. In Wikipedia, The Free Encyclopedia. Retrieved 18:34, April 12, 2022, from https://en.wikipedia.org/w/index.php?title=Automatic_summarization&oldid=1080772675

ABOUT THE AUTHORS

Dr. Robert Sottilare *is the Director of Learning Sciences at Soar Technology, Inc.(SoarTech). His research interests include using machine learning approaches to automate authoring, instructional management, and evaluation processes to enhance adaptive instructional tools and methods. Dr. Sottilare is the father and co-creator of the Generalized Intelligent Framework for Tutoring (GIFT), an open source AIS architecture. He has published over 250 technical papers in the training and learning sciences field, has over 2600 citations and is a senior member of the IEEE. He is the recipient of multiple lifetime and technical awards. Dr. Sottilare currently serves as the principal investigator for the MOTIVATE project.*

Angela J. Woods *is a Software Architect at SoarTech. with over 20 years of software engineering experience with emphasis on designing system architectures for real-time uses including intelligent training, agent-based simulation, data analytics and game development. She is an expert in dynamic adaptation techniques and her most recent work has included detecting semantically meaningful events in timeseries data streams and using machine learning techniques to build causal models that can augment adaptive intelligent training systems. She has extensive engineering experience including serving as the tech lead on over $5M in government funded research, where she successfully led teams of software engineers through full project lifecycles. She currently supports MOTIVATE as its chief architect.*

Nick Giranda *is the Tech Lead for the MOTIVATE project and has acted as the technical lead for multiple projects at SoarTech. These include simulation, automated tutoring, and autonomous vehicle based projects. Nick has worked with SoarTech for 6 years, and in that time has contributed to NAWC TSD, ADL, ARL, and DARPA research programs including multiple projects involving the Generalized Intelligent Framework for Tutoring (GIFT). Nick earned a Masters in Software Engineering from Loyola of Chicago, and a Bachelor of Arts in Mathematics from Knox College.*

Matthew Bertrand *is a Software Engineer at SoarTech and is responsible for the development of the MOTIVATE software baseline. He earned his Bachelor's degree in Computer Science at Eastern Michigan University.*

Dr. Eric Ortiz *serves as a Senior Program Manager at SoarTech, providing leadership and oversight for technical projects and multiple, multi-year DoD-funded programs. Dr. Ortiz has 25 years of experience in the development of digital media production, interactive web-based technologies, military simulations, serious games, and virtual environments. Over the course of his career, Dr. Ortiz has directed multiple design-related projects, including 3D modeling, video editing, and 2D graphic work for proposals, web presence, and print campaigns. He holds a Ph.D. in Modeling and Simulation from the University of Central Florida and a patent for Visual Correlation for Static Digital Images. He currently supports MOTIVATE as its project manager.*

Brad Friedman *currently leads the computing efforts for the Army Synthetic Training Environment (STE) Cross Functional Team (CFT) and has 20 years of experience as an engineer, scientist, test evaluator, and program manager supporting DISA, MARCORSYSCOM, Army NETCOM and AFC. He has occupied significant roles of increasing responsibility within many government organizations and continues to advance technology through collaboration with both government and industry to develop next generation training systems. He currently serves as the government technical point of contact (TPOC) and Contracting Officer's Representative (COR) for the MOTIVATE project.*

Enhancing GIFT AutoTutor

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

INTRODUCTION

AutoTutor (Graesser et al., 2003) has been one of the GIFT modules since the second release of GIFT. The current work proposes potential enhancements of the GIFT AutoTutor Module. Our work includes two enhancements: (1) Closed captioning in multiple languages and improved conversational avatars with multi-lingual support and (2) an Intuitive Authoring tool and Server-free implementation of Expectation-Misconception Tailored dialog (EMT).

- **Closed captioning in multiple languages and avatars with multi-lingual support**: In this work, a translation API was integrated with the AutoTutor module to enhance intelligent instructions. Any content can be delivered to the learner in their native language (as closed captions and avatars that speak multiple languages) with a few simple configurations. Additionally, real time conversations between human tutor and learner are done in diverse languages. An easy-to-develop technological solution for language adaptivity is powerful. The current enhancement of multilingual support uses the same methods that have been previously implemented (i.e., 3A enhancements; Ahmed, Shi et al., 2021; Ahmed, Shubeck & Hu, 2020; Ahmed, Shubeck, Andrasik et al., 2020; Ahmed, Shubek, et al., 2021).

- **Intuitive Authoring tool and Server-free implementation of Expectation-Misconception Tailored dialogue (EMT)**: Two of the bottlenecks for implementing EMT dialogue in AutoTutor is 1) to create new rules that manage the dialogue between the learner and the tutor and 2) reliance on a server-based AutoTutor Conversation Engine (ACE) that requires a web server. In this enhancement, we first implemented a simple authoring process for EMT. We simply used a table (spreadsheet) to organize the information necessary for the dialogue and then used a javascript library to manage the dialogue by connecting the scripts (content in the form of a table) and previous student's behavior (Experience Application Programming Interface (xAPI) statements in the Learning Record Store (LRS)). The new approach is flexible and easy to implement.

CLOSED CAPTIONINING

Learners interact with voice, text, and mouse clicks, using AutoTutor's learner interface (Nye et al., 2014). In the latest version learners interact with drag and drop exercises and via touch, particularly on tablets. Learners can receive content in their native language. In between these two way communications there is a google translation API (Groves & Mundt, 2015). Content created in a certain language is translated to the learner's language. Learner's feedback or answers as text input is translated back to base content language to verify correctness.

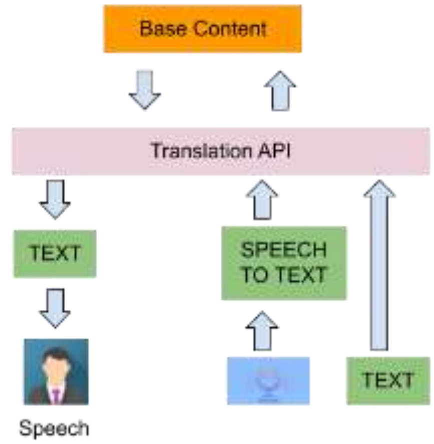


Figure 1. Two-way communication between AutoTutor and Learner, translation API in the middle.

The voice input of the learner is first converted to text and then translated to the base language. The text in the content is always spoken by the avatar. Figure 2 shows that the base content is translated in many different languages without having to manually recreate the content in the new language.

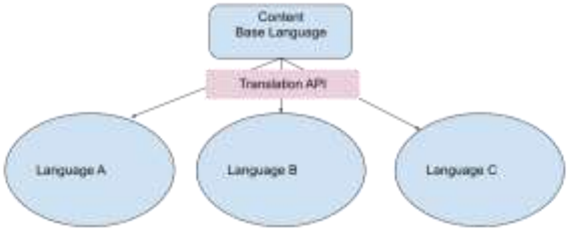


Figure 2. Content in base language is translated in different languages.

Sometimes AutoTutor is unable to answer a learner’s question. When this happens blended human tutors are necessary (i.e., behind-the-scenes human tutoring in combination with intelligent tutoring agents). The human tutor can answer by typing text or by speaking. This is also possible to do in the learner’s native language by using the translation API. In this case the human tutor does not speak directly to the learner but the speech is translated to the learner's language and then spoken by the avatar (see Figure 3).

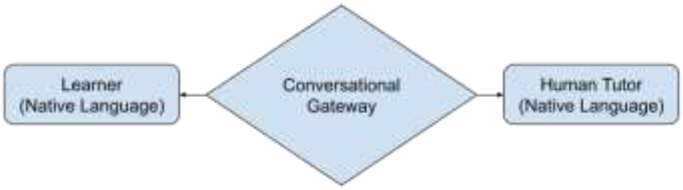


Figure 3. AutoTutor acts as a conversation gateway.

One shortcoming of this method is that the meaning can sometimes change through the translation process. A low translation accuracy may lead to a misinterpretation of the system when evaluating learner responses, which would lead to providing incorrect feedback to learners, ultimately affecting their learning outcomes. Determining the quality and accuracy of the translations requires further research. The so-called misunderstanding phenomenon may happen in machine-human communication. This actually pushes the discussion about the translation accuracy.

Regarding the accuracy of a translation, it involves the transition of different languages at multiple transfer nodes (Boitet et al., 2010), the inherent differences of various languages (Chakravarthi et al., 2021; Demirtas & Pechenizkiy, 2013), and the effectiveness of translation technologies/API (e.g., their algorithms) applied (Ekazuriaty, 2016; Johnson et al., 2017; Li et al., 2014) during the language delivery and translation process. Previous research has been implemented on the evaluation of the accuracy of Google Translate. Though the accuracy of Google Translate varies from language to language, Google Translate provided noticeably higher accuracy at some levels (Chen et al., 2016; Sutrisno, 2020).

A SIMPLIFIED EMT DIALOGUE

In this version of AutoTutor, we added a simplified EMT dialogue with traditional assessment tools, such as multiple choice questions. Consider the following implementation of this simplified EMT tutoring framework. The original *Operation ARA* is a game-based intelligent tutoring system (ITS) that uses an AutoTutor style conversation framework to teach scientific reasoning concepts (Cai et al., 2011). Operation ARA's content consists of several case studies or articles describing a scientific study, each with one to five scientific flaws (e.g., correlation does not mean causation, premature generalization of results, small sample size). These are considered "expectations" that must be met to provide a complete answer. Each case study also has several related misconceptions, which includes other scientific flaws that are not relevant to the case study.

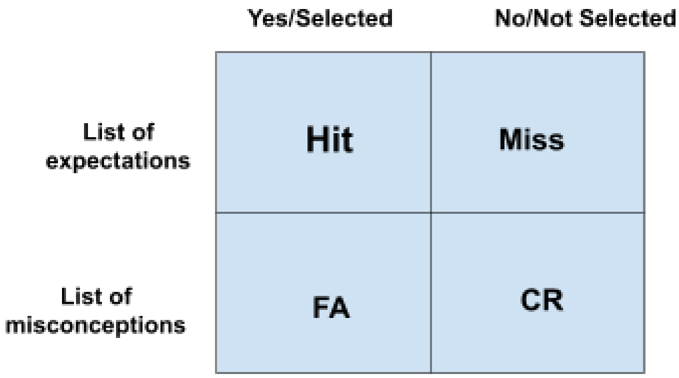
A typical dialogue flow would involve the user providing a response to a main question (e.g., What are the flaws in the current study?), which is compared to the possible explanations and misconceptions using latent semantic analysis (LSA) and regular expressions (RegEx). If the user provides an answer that does not match an expectation or misconception, they are provided a hint that covers one of the correct expectations (e.g., Consider that the parents who assisted in the study knew which condition each participant was in). The user would then provide more natural language input, which is again assessed using LSA and RegEx. If the response to the hint is incorrect the user is presented with a "prompt" which helps students articulate a missing key term (e.g., When researchers alter or affect the outcome based on how they interact with the participants we call this what?). If the user provides an incorrect response to the prompt, they would receive a feedback statement from AutoTutor which would then briefly describe the correct answer that covers that specific expectation.

A learner might express a misconception at any point during this hint → prompt → assertion cycle. If they express a misconception they receive a feedback statement describing why their statement was a misconception.

With the simplified EMT framework, users can select from a set of “select all that apply” multiple choice questions, each with several options. Each option would represent an expectation or misconception. Here, items that correspond to expectations are supposed to be selected and items that correspond to misconceptions should remain unselected. Learners can make two types of mistakes. a) unselected expectations, b) selected misconceptions.

Compared with the original AutoTutor, coverage of expectations in this case are “Hits”, and “observed misconceptions”. This dialogue model considers what students select as well as what students do not select. Figure 4 depicts the EMT dialogue mechanism. FA refers to “False Alarm”, CR refers to “Careful Rejection”.

We used three different interfaces to implement EMT dialogue: multiple-choice questions (MCQ), drag and drop objects, and hotspots (i.e., click on the correct objects/items located on the screen). It is both powerful and simple to capture misconceptions. In MCQ questions there are some items which are CORRECT answers and there are some items which are MISCONCEPTIONS. For drag and drop objects, the objects are mostly pictures. Learners are expected to choose a CORRECT object and drag it into a CORRECT area. There are some objects which are INCORRECT and can also be placed in the area. A similar mechanism is used for hotspots. Learners move objects and in real time AutoTutor captures where the objects are moved and how frequently. Tracking this information can help determine if a student is confused (e.g., moving an object to more than one location on the screen before submitting). Figure 5 shows examples of the user interface for drag and drop, and hotspots.



	Yes/Selected	No/Not Selected
List of expectations	Hit	Miss
List of misconceptions	FA	CR

Figure 4. Pictorial description of EMT dialogue mechanism

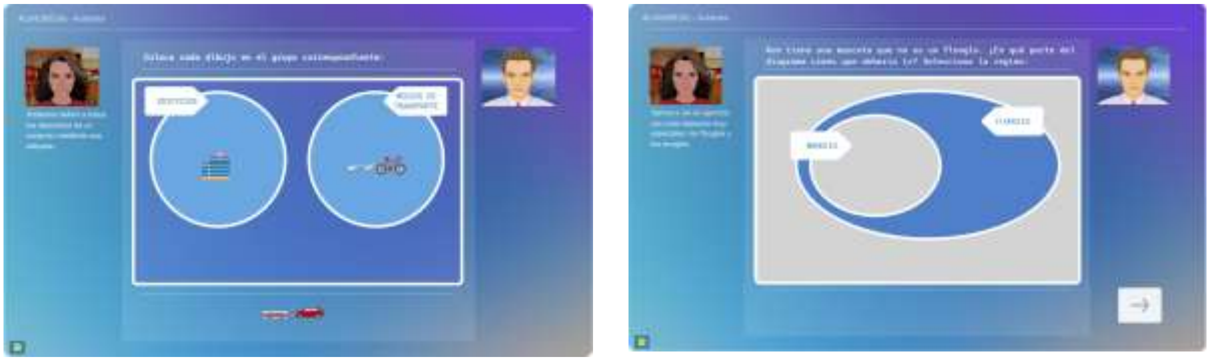


Figure 5. Drag and drop interface in AutoTutor

TECHNOLOGY

The work we present is primarily technological. We used commercial off-the-shelf (COTS) or open-source/open-access solutions for the current and previous enhancements. For multi-lingual support, we used Google's Translation API. The quality of the multi-lingual support will be as good as the API we use. Currently there are over 100 languages supported in Google translation. We use the API not only for closed captions for any of the 100 possible languages, we also use the API to produce paraphrases for any given language. For example, we can generate a huge number of variations for any given piece of text. This function enables us to dynamically generate semantically similar but syntactically different text in conversation-based tutoring. We used the most recent version of the Character API of Media Semantics. This API provides high quality characters that support all 100 languages. With the combination of Google’s Translation API and the Media Semantics Character API, we are able to produce conversation-based tutoring systems for learners with most of the world’s languages.

For the second enhancement, instead of using Extensible Markup Language (XML) as the original tutoring scripts, we used Google spreadsheet (which is a form of XML, but visually represented in the form of tables). With this solution, anybody can create interactions with a valid email address. All the scripts (content) are in the cloud and version controlled (Google Spreadsheets save a version for every 2 seconds). To manage dialogue, we have used javascript (nodejs) to manage communication among content (the Google spreadsheet), learner records (xAPI LRS), and the user interface (the web browser). The only server needed is a data server that stores LRS statements of the learners. The current system uses an instance of veracity LRS (see lrs.io). The current javascript library is on GitHub and it can be located on any web server.

PREVIOUS WORK

Previously, our team has made a version of AutoTutor that is Content-Aware, Context-Aware, and Learner Aware (3A). The 3A implementation was also “server-free” with the same general implementation principle (use COTS as much as possible). The current enhancement will work with the 3A enhancement. The previous enhancements reported are currently implemented as a standalone version of AutoTutor. We expect to have this integrated as part of GIFT to replace the earlier version of AutoTutor after we receive feedback from the participants of GIFTSym (Ahmed, Shi et al., 2021; Ahmed, Shubeck & Hu, 2020; Ahmed, Shubeck, Andrasik et al., 2020; Ahmed, Shubek, et al., 2021).

CONCLUSION

In this work we have extended the language capability of GIFT enabled AutoTutor 3A. We added heterogeneous language support using COTS technology. Additionally, we implemented a novel EMT dialogue framework to capture misconceptions of the learner and provide appropriate feedback.

REFERENCES

Ahmed, F., Shi, G., Shubeck, K., Wang, L., Black, J., Pursley, E., Hossain, I., & Hu, X. (2021). Collecting 3A Data to Enhance HCI in AIS. *International Conference on Human-Computer Interaction*, 499–508.

Ahmed, F., Shubeck, K., Andrasik, F., & Hu, X. (2020). Enable 3A in AIS. *International Conference on Human-Computer Interaction*, 507–518.

Ahmed, F., Shubeck, K., & Hu, X. (2021). Enhancement of GIFT Enabled 3A Learning: New additions.

Proceedings of the Ninth Annual GIFT Users Symposium (GIFTsym9), 151.

Ahmed, F., Shubek, K., & Hu, X. (2020). Towards a GIFT enabled 3A learning environment. *Proceedings of the Eighth Annual GIFT Users Symposium (GIFTSym8)*, 87–92.

Boitet, C., Blanchon, H., Seligman, M., & Bellynck, V. (2010). MT on and for the Web. *Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering)*, 1–10.

Cai, Z., Graesser, A., Forsyth, C., Burkett, C., Millis, K., Wallace, P., Halpern, D., & Butler, H. (2011). Trialog in ARIES: User input assessment in an intelligent tutoring system. *Proceedings of the 3rd IEEE International Conference on Intelligent Computing and Intelligent Systems*, 429–433.

Chakravarthi, B. R., Rani, P., Arcan, M., & McCrae, J. P. (2021). A survey of orthographic information in machine translation. *SN Computer Science*, *2*(4), 1–19.

Chen, X., Acosta, S., Barry, A. E., & others. (2016). Evaluating the accuracy of Google translate for diabetes education material. *JMIR Diabetes*, *1*(1), e5848.

Demirtas, E., & Pechenizkiy, M. (2013). Cross-lingual polarity detection with machine translation. *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, 1–8.

Ekazuriaty, N. A. (2016). *An Analysis on the google translation quality in translating english into Indonesia of the*

text" Cinderella". [PhD Thesis]. STAIN Ponorogo.

Graesser, A. C., Jackson, G. T., Matthews, E., Mitchell, H. H., Olney, A., Ventura, M., Chipman, P., Franceschetti, D., Hu, X., Louwerse, M. M., & others. (2003). Why/AutoTutor: A test of learning gains from a physics tutor with natural language dialog. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *25*(25).

Groves, M., & Mundt, K. (2015). Friend or foe? Google Translate in language for academic purposes. *English for Specific Purposes*, *37*, 112–121.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., & others. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, *5*, 339–351.

Li, H., Graesser, A. C., & Cai, Z. (2014). Comparison of Google translation with human translation. *The Twenty-Seventh International Flairs Conference*.

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

Sutrisno, A. (2020). The Accuracy and Shortcomings of Google Translate Translating English Sentences to Indonesia. *Education Quarterly Reviews*, *3*(4).

ABOUT THE AUTHORS

Dr. Faruk Ahmed *is a faculty member 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).*

Dr. Keith Shubeck *received his Ph.D. 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 Dr. Xiangen Hu in the Advanced Distributed Learning Partnership Lab at the Institute for Intelligent Systems. His dissertation work compared interactive and vicarious tutoring frameworks in a conversation-based intelligent tutoring system for critical thinking and scientific reasoning.*

Liang Zhang *is a Ph.D. Candidate of Computer Engineering at the IIS (Institute for Intelligent Systems) at The University of Memphis. His research interests include educational data mining and learning analytics. Now he works as a research assistant following Phil Pavlik in the Optimal Learning Lab of University of Memphis.*

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 III: Adaptive Instructional System (AIS) Architecture and Ontology

GIFT Giving and Receiving - Revisited

Austin Duncan[1], Gregory Goodwin[2], and Stacey Sokoloff[1]
Veloxiti, Inc.[1], US Army Combat Capabilities Development Command (DEVCOM) – Soldier Center[2]

INTRODUCTION

A Year Later

A year ago, the SMART (Simulation Monitoring and Reporting Tool) team discussed the challenges faced when creating Soldier training that leverages technologies developed by multiple organizations (Clarke et. al., 2021). The discussion included potential solutions for each challenge, noting that each potential solution came at a cost. A year later, the problems remain difficult, while the need for solutions has become even more critical.

Soldier training still requires a complex give and take between people and technology. Over the past year, the problem has only become more challenging. A reduction in force size and a renewed focus on near-peer adversaries has led to an increased reliance on technologies, including wearable devices, augmented reality, and Artificial Intelligence (AI) based solutions (Soldier Modernisation: Technology Trends). Soldiers are being asked to make complex decisions, sometimes with too little data available, but often with an overwhelming amount of data – so much data that finding the useful information is either practically impossible or requires using complex search technology. New tools are often a powerful enabler, but Soldiers must be trained to use the technology to complete missions. While mission and technological complexity are growing, a reduction in force size often requires Soldiers to fulfill multiple roles, both increasing job complexity and limiting the time available for training. The need to create broad and effective Soldier training has never been greater.

As the operational environment and state of the art technology have evolved, the Generalized Intelligent Framework for Tutoring (GIFT) has not remained static. GIFT is in an even better position to help address these issues due to improved Unity integration, more advanced condition classes, increased Experience Application Programming Interface (xAPI) support, and historic learner state tracking. These features provide training tool developers with a powerful platform that, among other things, encourages interoperability and promotes, but does not impose, structure.

Throughout 2021, the SMART team has updated SMART to add new features and to improve existing functionality. While addressing interoperability challenges has been on our minds, it has not been our primary focus. It has guided our decisions but has not had as strong of an influence as other factors, including feature requests. This is likely a common and often necessary occurrence. Still, over the course of the year, decisions were made that either promoted or detracted from interoperability. At this juncture, it seems important to revisit last year's discussion and re-assess each proposed solution.

SMART ARCHITECTURE UPDATES

The Year in Review

A year ago, SMART was finishing an initial implementation of a performance assessment capability for collective Battle Drill 5a, Attack a Bunker. This capability was integrated with GIFT in a limited manner;

limited because while the two technologies shared data, the integration did not provide significant value to either component. This is no longer the case.

Immediately following GIFTSym9 in 2021, the SMART team began leveraging GIFT's Game Master to offer an After-Action Review (AAR) for the bunker attack scenario. SMART focused on automatically producing a video that combines visuals from Virtual Battle Space 3 (VBS3) and audio from a verbal call for fire, and automatically adding the combined video to the GIFT domain session.

Starting in late 2021, the SMART team expanded its performance assessment capabilities to include CAS (Close Air Support). This capability differed from the other SMART domains in that it was quick and easy to run, making it a good candidate to explore and demonstrate GIFT's adaptive tutoring capabilities.

In addition to leveraging GIFT functionality, over the past year the SMART team refactored its assessment architecture to better encapsulate assessed tasks to ease implementation, increase code understandability, and to facilitate interoperability. The resulting Assessment Manager shared library simplified SMART course creation.

SMART Architecture Overview

SMART (Figure 1) contains a Service Oriented Architecture (SOA) of components that work together to provide collective task performance assessment. Services vary in complexity and responsibility, ranging from a simple Fire Direction Center (FDC) simulator to a complex performance assessment engine that leverages the Velox Toolkit. The SMART architecture is still relatively young—as it matures, commonalities are identified and addressed, often by abstraction, shared libraries, and/or new components. The SMART team follows an agile development process and views refactoring as expected and necessary.

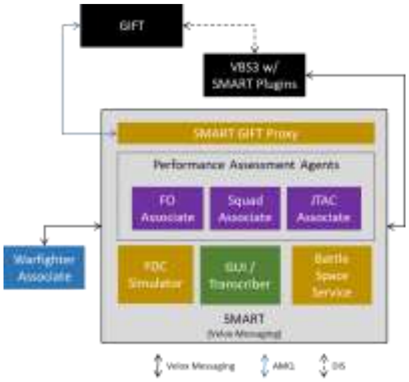


Figure 1. SMART Architecture

Since GIFTSym9, updates to the SMART architecture include a new associate, Joint Terminal Air Controller (JTAC), a new service, the Battle Space Service, and a new shared library, the Assessment Manager. Other major updates include AAR playback using GIFT's Game Master.

SMART GIFT Integration

A year ago, SMART used GIFT to obtain Distributed Interactive Simulation (DIS) data and SMART shared its performance assessments and explanations with GIFT. Since then, SMART has integrated more closely with GIFT, leveraging its Game Master AAR capability and SMART recently began to use GIFT to provide a simple example of adaptive tutoring. What once was a limited integration is now providing substantial benefit to SMART and opportunities exist to increase the benefit.

Run-From-Log Playback Integration

SMART initially integrated with GIFT's run-from-log functionality, thinking that run-from-log was how GIFT supported AAR playback. There were challenges integrating with the run-from-log functionality—particularly, SMART used real-time timestamps for tracking and grading its tasks, however run-from-log required SMART to use timestamps provided by GIFT. Abstracting the source of timestamps is good practice and we were able to overcome this issue. Using run-from-log for AAR requires SMART components to reassess the mission data, which has the benefit of enabling different results with different configurations. On the other hand, run-from-log AARs are not very visually interesting. To increase the visual interest, we added a map to the SMART User Interface (UI). The run-from-log feature provided a useful testing mechanism and enabled exploration of configuration value updates, but we were not yet using it in a way that would provide Soldiers with a simple and rich AAR experience. Then we realized that GIFT already provided the functionality that we wanted in Game Master.

GIFT Game Master AAR Integration

Following the run-from-log integration, SMART integrated with GIFT's Game Master AAR functionality. Using Game Master to conduct AARs is much simpler than run-from-log and likely provides a better user experience with access to tasks, grades, maps, and visual and audio playback of the simulation on GIFT's user interface, without requiring any SMART components. In the case of the Squad SMART Battle Drill 5A scenario, special care had to be taken to provide the correct audio/video recording to GIFT with minimal user intervention required.

The most complex scenario supported by SMART is the Battle Drill 5A bunker attack scenario. In addition to making heavy use of VBS3, it involves conducting a verbal fire mission, in which the trainee conducts simulated radio calls using the SMART Transcriber and receives verbal responses from a simulated FDC. To make the AARs as engaging as the missions, we wanted to include video from VBS3 and audio from the fire mission, including calls spoken by the Forward Observer (FO) trainee and the computer-generated audio from the simulated Fire Direction Center (FDC). Manually capturing this data and adding it to Game Master requires the user to follow a multi-step process, which we almost completely automated. We now record each verbal fire mission exchange and trigger video recording in VBS3 to start and stop appropriately. SMART uses a library called FFMPEG to merge the audio and video, and the file location of this video is sent to GIFT. The SMART VBS3 Interop in GIFT then copies the video into its most recent domain session and writes the appropriate metadata files to direct appropriate playback of the session in Game Master. All that is required of the user is selecting the pre-loaded video for display in GIFT.

SMART Architecture Components

Battle Space Service: The Battle Space Service is a new component, that acts as a hub for receiving location-based data out of GIFT via DIS or VBS directly via C++ Plugins. It contains logic to forward location updates to ATAK in configurable batches to alleviate stress on Graphical User Interface (GUI) components.

FDC Simulator: The FDC Simulator has not changed since GIFTSym9. It still initiates explosions in VBS3 by sending commands to a SMART VBS3 plugin and provides verbal responses to fire mission communications.

FO Associate: The FO Associate, which is responsible for performance assessment of adjust fire missions, was updated to use the Assessment Manager shared library. Its functionality has remained unchanged. It still assesses grid, polar, or shift from known point fire missions based on the communication between the FO and the simulated FDC and environment data (explosion locations, etc.).

GUI / Transcriber: While the purpose of the GUI remains unchanged, it still displays task grades in real time and contains a speech to text transcriber that enables simulated radio communication between the FO and FDC, the GUI's role as a source of mission information has expanded since GIFTSym9. The Mission Briefing tab contains scenario data necessary for CAS missions. The Map View tab contains a real-time map with DIS data (entity locations, explosions, etc.) and mission configuration data (ex. the covered path). The Mission State tab contains a tabular view of the order of battle, including the current location of each entity. The SMART GUI can be seen in Figure 2.

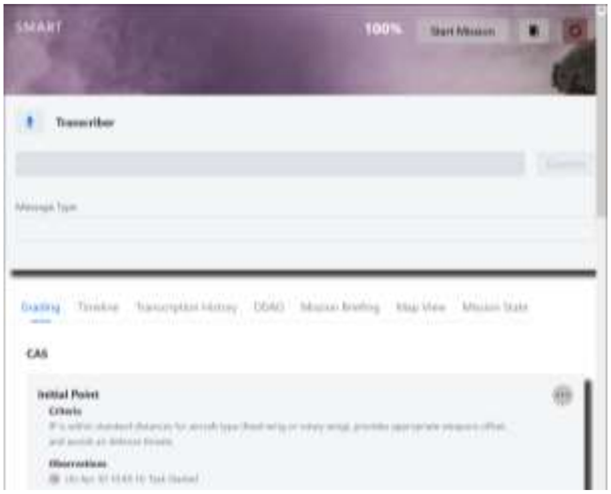


Figure 2. SMART GUI

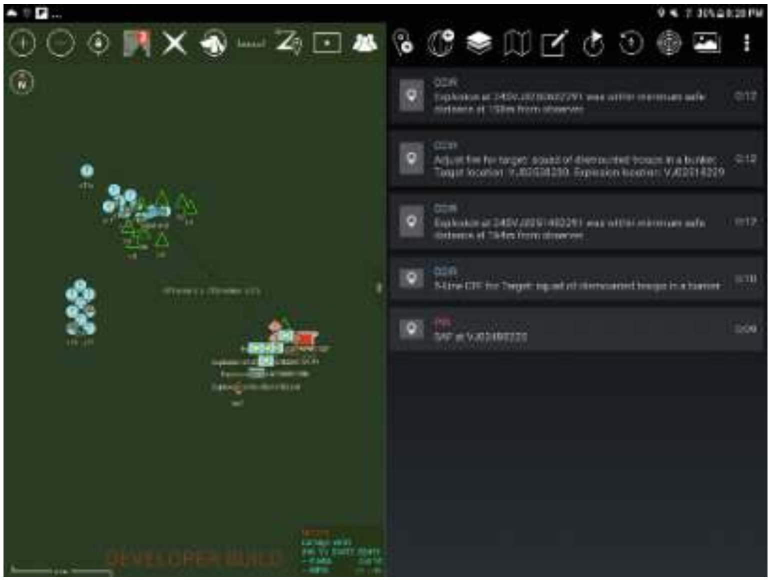
JTAC Associate: The new Joint Tactical Attack Controller (JTAC) Associate assesses performance for CAS missions. It uses position data from VBS3 and scenario description data simulating information that would be communicated to a JTAC on a Joint Tactical Air Strike Request form, shown to the user on the Mission Briefing tab of the SMART GUI, to assess how well the user filled out a CAS request form in the Android Tactical Assault Kit (ATAK) (Figure 3). The JTAC Associate uses the form data to command VBS3 to conduct an airstrike. Unlike other SMART mission types, CAS missions are quick, making it possible to conduct multiple repetitions in a short time. To take advantage of this benefit, the JTAC Associate contains logic to coordinate with other components, including VBS3, to enable the user to restart the mission with one button press.

SMART GIFT Proxy: While this component's main purpose remains unchanged (to translate between Velox Messaging, which is used by SMART components, and out of the box ActiveMQ which is used by SMART to communicate with GIFT), the SMART GIFT Proxy now assists with the automatic inclusion of the AAR video by commanding VBS3 to start and stop recording, combining the audio and video tracks by using FFMPEG, and sending the path of the resulting MP4 to GIFT.

SMART VBS3 Plugins: The changes to the SMART VBS3 Plugins over the last year have been minimal and mostly related to SMART's improved order of battle functionality. The SMART VBS3 plugins still enable SMART to obtain data that is not available through DIS, such as laser pointer and chem light use. The SMART VBS3 plugins also enable SMART to send commands to VBS3.

Squad Associate: The Squad Associate has been updated to use the Assessment Manager and has had minor logic improvements. Its timestamp handling was updated to support GIFT run-from-log. Its main purpose, to assess Army Battle Drill 5a tasks other than the call for fire, has remained unchanged.

Warfighter Associate: SMART continues to leverage the government owned Warfighter Associate (WA) to interact with ATAK. The WA provides enhanced situation understanding and decision aiding by mining tactical chat to identify events relating to configured information requirements. SMART leverages the WA mostly for its ability to interact with ATAK, providing SMART with access to CAS and indirect fire requests and providing SMART with a means of providing intelligent aiding in an operational tool, promoting train as you fight. While most of the changes that occurred in the WA did not impact SMART, relevant changes include updating to the latest ATAK version and adding the ability to display popup messages, used by the JTAC scenario for hints. ATAK with the WA plugin can be seen in Figure 3.



DDR
Explosion at [illegible] was within minimum safe distance at 158m from observer 0:12

DDR
Adjust fire for target: squad of dismounted troops in a bunker. Target location: [illegible]. Explosion location: [illegible] 0:12

DDR
Explosion at [illegible] was within minimum safe distance at 164m from observer 0:12

DDR
5-Line CFF for Target: squad of dismounted troops in a bunker 0:10

SAF at [illegible] 0:09

Figure 3. ATAK with WA Plugin

Scenarios

All SMART scenarios use the following components: GUI/Transcriber, SMART GIFT Proxy, and SMART VBS3 Plugins. The Battlespace Service and the WA are always required to display DIS data in ATAK. Scenario specific components are discussed in each scenario section.

Adjust Fire Mission

SMART was initially focused on providing performance assessment for grid, polar, and shift from known point adjust fire missions. While simple from a VBS3 scenario perspective, this mission type is highly interactive, enabling trainees to conduct the mission verbally using the SMART Transcriber, which leverages an external speech-to-text tool, such as Dragon Naturally Speaking. In addition to the common components, this scenario requires the FDC Simulator and the FO Associate.

Attack a Bunker

The bunker attack scenario is the most complex scenario supported by SMART. It was created to expand SMART from an individual training tool to a tool capable of assessing collective training. The bunker attack scenario provides collective training for Battle Drill 5a. It consists of several stages, including a Squad Leader (SL) reacting to contact, the SL requesting fire, and the FO calling for and adjusting fire. The scenario assumes that the SL requesting indirect fire support uses ATAK for communication, the FO has access to both ATAK and a radio, and the FDC only has radio access (Figure 4). In addition, SMART grades tasks pertaining to the Platoon Sergeant and Platoon Leader, including how successfully they join the squad in contact. All these tasks use configurable values that allow the course creator to modify the difficulty and realism of the course.

This scenario was developed to support collective training, but for practical reasons, it is often run by an individual. Roles other than the SL are automated as much as possible in VBS3. In certain parts of the scenario, when running as an individual, the SL user must assume another role. Performing collective task assessment has proven easier for SMART than the mechanics of running a collective scenario as an individual. The bunker attack scenario uses all components except the JTAC Associate.

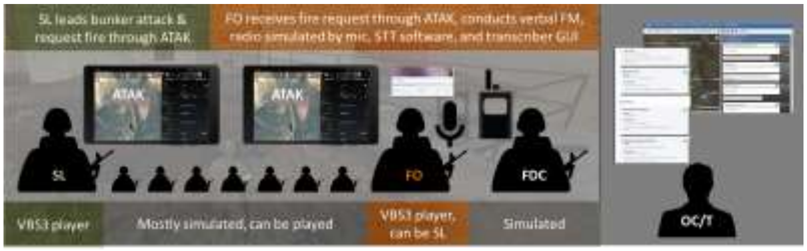

SL leads bunker attack & request fire through ATAK
FO receives fire request through ATAK, conducts verbal FM, radio simulated by mic, STT software, and transcriber GUI
ATAK
ATAK
SL
FO
FDC
OC/T
VBS3 player
Mostly simulated, can be played
VBS3 player, can be SL
Simulated

Figure 4. Squad SMART Scenario

Close Air Support

The newest SMART scenario provides training for CAS requests using ATAK's nine-line request form (Figure 5). It trains tasks such as Initial Point (IP) selection, egress route selection, and ordnance selection. Unlike other scenarios which typically take over 10 minutes to complete, the CAS scenario is relatively quick, providing the SMART Team with a good opportunity to explore adaptive tutoring. To make it easy for a trainee to explore the impact of changing items in the CAS nine-line form, the SMART GUI was updated to allow the scenario to be reset with one button press.

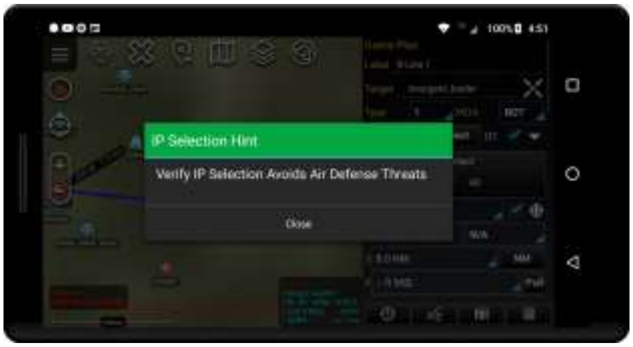
Figure 5. CAS Form

Resetting the scenario is different from restarting all scenario components. When the scenario is reset, the SMART components clear all data. The VBS3 scenario is reset as close to the original starting state as possible. Tasks in GIFT are set to At Expectations, but the GIFT course is not restarted, allowing GIFT to retain performance data, enabling it to trigger the necessary strategies based on performance trends.

There are two types of strategies/adaptations used by the CAS scenario, a scenario difficulty update and a progressive hint that provides more information when a user continues to get a No-Go on a task. These adaptations are managed using strategies in the GIFT Domain Knowledge File (DKF) for the course.

Difficulty adaption in this scenario is achieved using VBS3's teleport functionality triggered from a strategy in the course DKF. Currently, this adaption moves an enemy MANPAD to a location that makes selecting the appropriate IP, and relevant surface-to-air threats more difficult.

Hint adaptation (Figure 6) is achieved using a progression of ordered strategies in the DKF. The initial implementation focuses on IP selection since it is observable before the nine-line is submitted, enabling SMART to provide guidance when the user has time to react. There are three hint levels, each providing more detailed information. The hint level increases when a user gets a No-Go grade on the IP selection task, until the maximum hint level is reached. In addition to the common components, the CAS scenario uses the JTAC Associate.



IP Selection Hint

Verify IP Selection Avoids Air Defense Threats

Figure 6. Basic Hint in ATAK

TAKING OUR OWN ADVICE

The past year has been busy. We spent most of the year focusing on adding new functionality and had to balance between future focused design goals and the development of tangible features. While developing concrete functionality was foremost in our minds, the goal to develop interoperable components implicitly influenced each decision. We evolved the SMART architecture and added new features without referring to our GIFTSym9 paper, but writing a paper changes its authors, and the ideas offered by the paper influenced our decision-making process, begging the question: *how well did we take our own advice?*

Last year, the SMART team identified 4 major challenges: intermediate conclusion sharing, combining course objects, performance assessment categorization, and high-level attributes and competencies.

Intermediate Conclusion Sharing

Intermediate conclusion sharing refers to sharing domain-based assessments between tasks/concepts, between courses, and/or between training technologies. Potential solutions included micro-services, data sharing (macro-services), common libraries, and advanced GIFT interoperability plugins.

Each SMART mission type is handled by a different agent with unique needs. The need to share conclusions between agents has been surprisingly minimal; too minimal to justify creating separate services and/or GIFT interoperability plugins. This lack of overlap may be a result of the inherent differences between missions. FO, a SL, and a JTAC must draw similar types of conclusions, for example, maintaining minimum safe distances, the specifics vary.

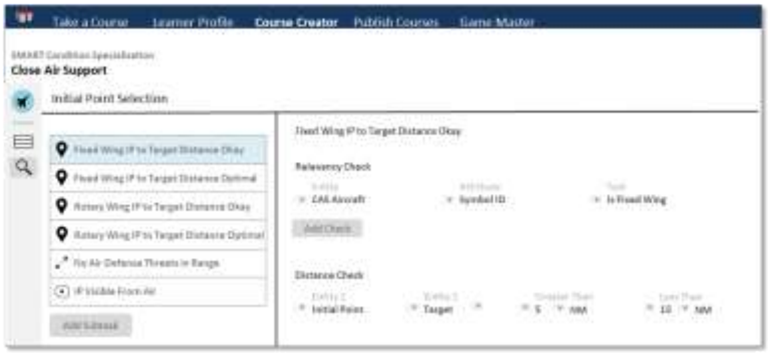
SMART uses a service oriented architecture (SOA) comprised of approximately 10 processes, each with its own responsibility. SMART has always used libraries for complicated functionality required by multiple processes. We have shared libraries for orders of battle, geometric calculations, and messaging types. Sharing libraries does not share conclusions but rather shares a means of reaching conclusions, which often may be better since it enables components to apply reasoning logic in a targeted manner.

Within an organization, using common libraries is usually considered good software engineering. There is a greater documentation cost when libraries are shared externally. SMART has not had the opportunity to share libraries with third parties but doing so should not be difficult. Shared libraries enable tasks, within or across agents, to apply the same logic to reach a conclusion. For example, sharing an order of battle library ensures that each SMART task applies the same logic to aggregate damage. Shared libraries share logic, not conclusions. For the most part, SMART tasks are independent and do not need to share conclusions. An exception is the mission phase.

The concept of mission phase originated from the first domain that SMART covered, fire missions. Fire missions can be divided into Call for Fire, Adjustment, Fire for Effect, and End of Mission phases. Applying a similar concept to each agent has proven beneficial. Checking for a mission phase in a task starting or stopping condition is more maintainable and often easier to understand than having individual tasks perform the assessments. Mission phase is tracked in memory by each assessment agent and for the most part, this is sufficient. The exception is that the Squad agent needs to know when the fire mission is complete. Sharing a messaging library makes sharing this conclusion straight forward. Mission phase is perhaps the most prominent intermediate conclusion shared by SMART.

Despite the current lack of overlap, SMART may benefit from having a common domain model that publishes conclusions to, and/or can be queried by, tasks. This domain model would be the central source of world state, including both raw data and assessed conclusions. Ideally, the model could save and visualize its state in a human readable format, which would help during development and during AARs by providing a snapshot of the context at the time each assessment was made.

Enhancing the model with built in functions that can draw conclusions based on specific types of data may help make certain model extensions available to non-technical users. Figure 7 shows a notional example of how a non-technical user could specialize a task. The UI would present the user with different options based on the type of check being performed. In this example, the user is specifying that a task must check whether the distance between the target and the IP is within the doctrinal boundaries for a fixed wing aircraft.



SMART Condition Specialization

Close Air Support

Initial Point Selection

- Fixed Wing IP to Target Distance Okay
- Fixed Wing IP to Target Distance Optimal
- Rotary Wing IP to Target Distance Okay
- Rotary Wing IP to Target Distance Optimal
- No Air Defense Threats in Range
- IP Visible From Air

Fixed Wing IP to Target Distance Okay

Relevancy Check

Entity: CAS Aircraft | Attribute: Symbol ID | Test: Is Fixed Wing

Distance Check

Entity 1: Initial Point | Entity 2: Target | Greater Than: 5 NM | Less Than: 10 NM

Figure 7. Condition Specialization Mockup

To make the problem tractable and keep the UI simple for non-technical users, the domain model would have to be specialized for the use case, in this case CAS. Specializing the model underlying the UI enables the selection space to be restricted. Referring to the example above, restricting the domain to CAS would limit the symbol ID tests to perhaps: *Is Fixed Wing*, *Is Rotary Wing*, and *Is Air Asset*. A general military domain editor would require a potentially overwhelming number of choices. Despite the potential benefits of a central domain model, the SMART Team has never been able to justify its creation.

Over the last year, SMART found more need to share functionality than conclusions and has accomplished this through shared libraries. Our needs have not justified the cost of implementing the other potential solutions. This is not to say that we do not think that the other solutions have merit, just that in our situation, the cost is not yet justified.

Combining Course Objects

Combining course objects refers to the challenge of enabling a course creator, ideally one who is not technical, to put together a training course comprised of course objects created independently. Potential solutions included verbose documentation and/or mapping to a domain specific data model.

This challenge is perhaps the most ambitious challenge discussed in this paper. If successfully addressed, non-technical course creators will be able to leverage assessment and instructional course objects created by different organizations to make a course that satisfies their needs. For our GIFTSym9 presentation, SMART designed a UI that showed how, if sufficient meta-data existed, GIFT could recommend course objects to assess a battle drill in a specific environment. While not as ambitious as the notional course creator, SMART was able to get insight into this challenge by attempting to replace the portion of the Squad mission involving indirect fire with CAS. This involved removing a handful of tasks in the middle of a scenario and replacing them with another set of tasks. The results were encouraging. Like conditions in GIFT, tasks in SMART are encapsulated and have clear starting and stopping conditions. The main point of communication between tasks is mission phase, which in this case could be set as easily and naturally by CAS as it could by the fire mission.

Despite the SMART team not following the recommendation to write verbose documentation, simple code inspection revealed that the starting condition of a task, *Approach the bunker using a covered path*, was dependent upon an output of the fire mission, namely the *end of mission* transmission. Conceptually, the covered path task should not be concerned with the specific detail that *end of mission* was transmitted over the radio but rather that it is an appropriate time to begin maneuvering toward the bunker because the combined arms attack (indirect fire or CAS) is complete. Updating the JTAC Associate to publish such a message was easy, because all SMART components use a common messaging framework.

SMART leverages Velox Messaging, which is a government owned messaging framework based on Google Protocol Buffers and Apache ActiveMQ. Velox Messaging allows the creation of XML data contracts to specify types, such as Entity Location, and service contracts to specify publications and subscriptions. Since all SMART components use a common messaging language, mixing and matching tasks, even across agents, only requires resolving logical differences.

The biggest challenge faced when replacing indirect fires with CAS was not updating the SMART components, but rather updating the VBS3 scenario. This was true despite the years between the creation of the fire mission assessment agent and the CAS assessment agent, suggesting that, encouraging developers to share source code, especially self-documenting source code, promotes interoperability.

Performance Assessment Categorization

The performance assessment categorization challenge involves determining the right level of nuance for task grading and promoting commonality or at minimum understandability of assessment logic. Proposed solutions include careful documentation and the use of metadata.

SMART uses a Go / No-Go classification. GIFT is slightly more detailed with Above Expectations, At Expectations, and Below Expectations. While SMART has not updated its implementation to deviate from the binary classification that is in use throughout the Department of Defense (DoD), sufficient information exists in most tasks to provide additional nuance, either in categorization (e.g. slightly above expectations) or in skill (e.g. a task to return timely well aimed fire was considered a No-Go even though the fire was well aimed because it was not timely). Providing clear and detailed assessment explanations, was always an aim of SMART, and the need has become even clearer over the past year as additional task types were added.

With the addition of the CAS scenario, use of the shared Assessment Manager library was essential. This approach allows each task, across assessment agents, to define a grading algorithm following a shared structure provided by abstract classes. Further, the library forces a common task structure and assessment methodology —providing for simpler understanding of assessment logic for developers.

Extending the SMART implementation to support a larger performance state space (grades other than Go and No-Go) would require updates to the architecture to add the additional grades, and to each task to set the grade appropriately. The architecture updates would be simple. The difficulty required to update each task to use a larger state space depends upon the necessary updates, which require domain research.

Many of the tasks are graded based on time and distance requirements, lending themselves well to a state space larger than Go and No-Go. As such, with the appropriate input from a domain expert, more nuanced grades could be assessed. For example, in the bunker attack scenario, trainees must return well-aimed fire on target within a configured time frame. The performance state space could match that of GIFT with Below, At, and Above Expectations—where Below Expectations is when either the well-aimed criteria or the time constraint is not met, At Expectations is when both criteria are met, and Above Expectations could be either based on outperforming either criterion.

A benefit of a larger performance state space is that it encourages continuous improvement. Many of the tasks assessed by SMART are inherently continuous. For example, in Battle Drill 5a, the task to approach the bunker using a covered path is currently assessed Go if a configurable percentage of approach locations are under cover and the approach is completed within a configurable timeframe. Providing more nuanced grades to encourage a safer and faster approach would be a natural extension.

As a Soldier proceeds through training, expectations rise. Performance that was once considered Above Expectations may be considered At or even Below Expectations as the Soldier advances. Perhaps adjusting performance state boundaries as Soldiers transition from novice, to journeyman, to expert may help motivate Soldiers to improve. Providing a method for driving deliberate practice and continual improvement could be as simple as modifying the current method of handling configuration values; rather than have a configured value that must be met, configure the standard values that mark performance requirements for the state space. In this example, , have the highest state, Above Expectations, be a dynamically moving value - having the score required for achievement increase until the trainee has perfected the scenario.

Capturing High Level Attributes and Competencies

Capturing high-level attributes, and competencies, refers to the challenge of uncovering aspects of performance that do not directly map to tasks, but relate to overarching skills (high-level attribute) or an externally defined competency. Proposed solutions included using GIFT to map tasks/concepts to higher level attributes or to assess aggregate data retrieved from a Learner Record Store (LRS).

The SMART domains are sufficiently complex that high level attributes—ranging from softer skills such as squad coordination (commanding shift fire, directing squad to cover/fall back/return to formation) for the SL and Platoon Sergeant to radio usage for the FO – can be identified, but the benefit of tracking each attribute depends upon the purpose of the assessment. For example, tracking radio usage may not be important for Soldiers who have not yet been trained in proper radio procedure but are conducting calls for fire to improve target location skills.

The mapping between tasks and high-level attributes is many to many. Each task may map to numerous high-level attributes and each high-level attribute may be tracked by numerous tasks. Currently, tasks in SMART are often multi-faceted, meaning that numerous skills must be exercised to receive a Go. For example, one facet of the bunker approach task is to remain under cover. Another is to arrive at the bunker in a timely manner. Each facet supports the assessment of different high-level attributes. In its current form, SMART could be enhanced to track high level attributes, but since facets are part of the internal task implementations, the mapping must be hard coded. SMART could be refactored to treat facets as subtasks, which would be necessary to make the mapping externally configurable.

Tracking competencies is like tracking high-level attributes, but competencies are defined by an authoritative external organization. SMART can easily be updated to send additional messages to map SMART tasks to competencies. It is hard to assess how well SMART tasks map to defined competencies without having domain specific examples, but assuming logical overlap exists, SMART could send the necessary data to GIFT. The open-source Competency and Skills System (CaSS) allows developers to define competencies and to establish relationships between them. The inclusion of CaSS in GIFT makes integration with CaSS a logical progression for developers in the assessment/tutoring realms.

SMART is starting to explore using GIFT to adjust task difficulty based on performance trends within one GIFT session. This setup enables the trainee to execute multiple CAS requests, and benefit from adaptations without requiring a LRS. All progress is lost when the GIFT course is restarted. Going forward, the ability to track and view performance trends across sessions and courses will allow for informed modifications to scenarios to focus a trainee’s learning to areas where he is deficient, driving toward an expert level of performance. To do this well, SMART and/or GIFT would need to track domain specific scenario metadata such as duration, locations traveled, time of day, and weather. To make informed decisions about performance trends, this data would need to be stored over time to provide insight into what factors caused each assessment. While SMART could store this data, likely the better solution is to use GIFT to store the data in a LRS, perhaps through xAPI statements.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

Interoperability between training tools and technologies remains a difficult challenge, but one worthy of attention. It will not be solved quickly, nor will it be solved by an individual organization, but rather requires collaboration between different groups. GIFT is in an excellent position to support continuing discussions and provide potential solutions, since as a general adaptive tutoring framework it sits between the parties and components that need to interoperate. These are difficult challenges, and the answers are currently unknown, but hopefully, by asking the right questions, GIFT can shape the discussion that improves training technology interoperability.

REFERENCES

Clarke, T., Duncan, A., Goodwin, G., Merrihew, J., & Sokoloff, S. (2021). GIFT Giving and Receiving: Helping Vendors Share Appropriately. In Proceedings of the 9th Annual GIFT User Symposium. Orlando, FL.

Soldier Modernisation: Technology Trends. (2022, April 13). Army Technology. https://www.army-technology.com/comment/soldier-modernisation-technology-trends/

ABOUT THE AUTHORS

Austin Duncan *is a Software Engineer with Veloxiti, Inc. He is a Navy Information Technician veteran having extensive experience in Naval software, communications, and messaging systems. Since joining the company in 2018, he has contributed to numerous projects using his experience in AI, UIs, and software design to help warfighters train, maintain situational awareness, and perform their roles with increased expertise. Before joining Veloxiti, Austin earned a Bachelor or Science (BS) from the Georgia Institute of Technology with concentrations in Computational Theory and AI.*

Dr. Gregory Goodwin *is a senior research scientist with U. S. Army Combat Capability Development Command Soldier Center at the Simulation and Training Technology Center (DEVCOM-SC-STTC), in Orlando, Florida. For the last decade, he has worked for the Army researching ways to improve training methods and technologies. He has a PhD in Psychology from Binghamton University and an M.A. in Psychology from Wake Forest University.*

Stacey Sokoloff *is a Senior 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.*

Synthetic Training Environment Experiential Learning for Readiness (STEEL-R) Demonstration

Kevin Owens[1], Benjamin Goldberg[2], Robby Robson[3], Michael Hoffman[4], Fritz Ray[3], Alex Colburn[3], Mike Hernandez[3], Mile Divovic[3], Shelly Blake-Plock[5], and Cliff Casey[5]
Applied Research Lab, University of Texas at Austin[1], US Army Combat Capabilities Development Command (DEVCOM) – Soldier Center[2], Eduworks Corporation[3], Dignitas Technologies[4], Yet Analytics[5]

INTRODUCTION

The Synthetic Training Environment (STE) Experiential Learning for Readiness (STEEL-R) team will present an introduction and give a live demonstration of STEEL-R at GIFTSym 2022. The STEEL-R Team intends to provide a live, web-based demo version of this system to the public in 2022.

STEEL-R is a combination of existing and new software that captures and evaluates the experiential competence of individuals and teams (Goldberg et al., 2021). STEEL-R incorporates modified versions of the US Army Combat Capabilities Development Command (DEVCOM) Soldier Center's Generalized Intelligent Framework for Tutoring (GIFT), the Competency and Skills System (CaSS; an open-source Learning Record Store (LRS) provided by Yet Analytics), and several other existing components. It enhances and extends the Total Learning Architecture (TLA) (ADL Initiative, 2019) learning ecosystem and adds several new components, including an Experience Design Tool (XDT) and an Observer/Controller Trainer (O/CT) analytics dashboard.

The STEEL-R project was initiated in 2020 (Goldberg, 2020). Previous research funded by the US Army Research Laboratory (ARL) used the TLA Experience Application Programming Interface (xAPI) (ADL Initiative, 2017) to capture learner data. STEEL-R goes beyond these previous efforts by (a) gathering data from multiple and diverse sources with a focus on immersive experiences, (b) measuring performance in multiple echelons (ranging from individuals to squads) and at multiple levels of proficiency, (c) using an xAPI profile that enables full traceability of evidence, and (d) providing a dashboard that shows competency and performance trends (Robson et al., 2022). The STEEL-R program includes research inspired by previous educational research on experiential learning throughout the 20th century, and previous ARL research from direct experience, interoperable performance assessment, measuring performance at multiple echelons (specifically the individual, fire team, and squad) (Owens et al., 2020), and establishing a foundation for adaptation of the team experience and individual experience.

A key aspect of STEEL-R is the incorporation of competency-based experiential learning (CBEL) (Owens & Goldberg, 2022). Based mostly on Experiential Learning Theory (Kolb & Kolb, 2017), and other related learning theories, CBEL was developed for volatile, uncertain, complex, and ambiguous (VUCA) performance training applications such as those that frequently occur in military operations. CBEL is an active approach for building expertise based on andragogy, in which learners (actors) engage in training sessions based on their own unique performance needs, their assigned team, competency states and/or the inherent needs of their unit. Actors achieve competence through long-term mental-models developed over time as they accumulate experience performing targeted tasks under various conditions. This approach gradually builds experiential expertise (Owens & Goldberg, 2022) that builds the "sense-making" competence required by all warfighters in VUCA conditions. STEEL-R focuses on building competence with respect to team and individual tasks and their related affective, behavioral, and cognitive competencies within the "crawl" phase of US Army training (US Army TRADOC, 2021).

The STEEL-R demonstration that will be shown at GIFTSym is the culmination of two-years of research that involved multiple parallel tracks. Underlying this work is a competency-based approach to measuring and orchestrating experiential learning. Originally intended to demonstrate a competency-based data-collection and modeling strategy for the US Army's Synthetic Training Environment (STE) program, STEEL-R now includes experience design. The conceptual high-level architecture in Figure 1 shows how STEEL-R fits within the larger STE and Army Training Information System (ATIS) domains. To manage its initial scope, the STEEL-R project has focused on the training environment for a new version of the existing US Army Games-for-Training synthetic environment (US Army PEO-STRI, 2005).

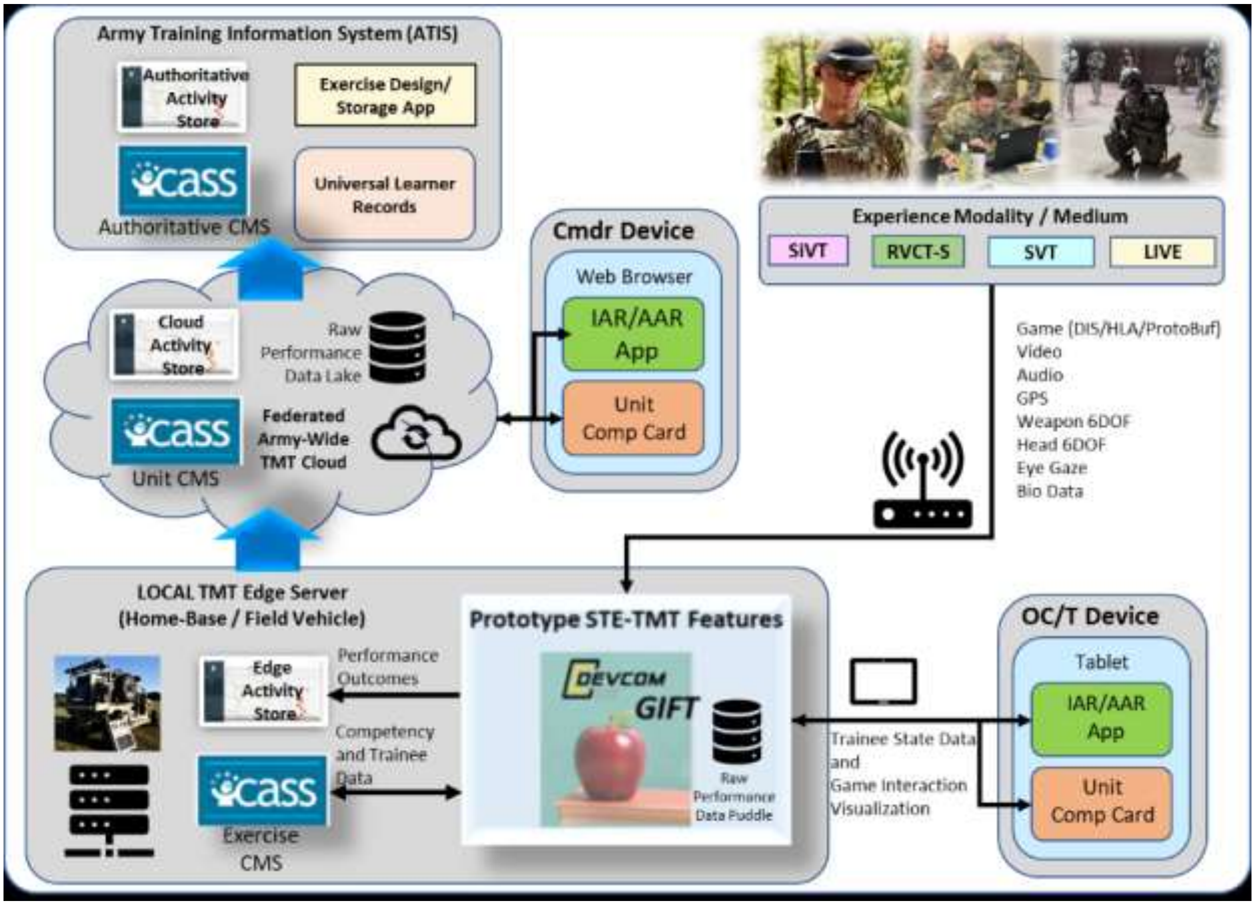

Army Training Information System (ATIS)
Authoritative Activity Store
Exercise Design/ Storage App
cass
Authoritative CMS
Universal Learner Records
Cloud Activity Store
Raw Performance Data Lake
cass
Unit CMS
Federated Army-Wide TMT Cloud
Cmdr Device
Web Browser
IAR/AAR App
Unit Comp Card
Experience Modality / Medium
SIVT
RVCT-S
SVT
LIVE
Game (DIS/HLA/ProtoBuf)
Video
Audio
GPS
Weapon 6DOF
Head 6DOF
Eye Gaze
Bio Data
LOCAL TMT Edge Server (Home-Base / Field Vehicle)
Edge Activity Store
Performance Outcomes
Competency and Trainee Data
cass
Exercise CMS
Prototype STE-TMT Features
DEVCOM
GIFT
Raw Performance Data Puddle
Trainee State Data and Game Interaction Visualization
OC/T Device
Tablet
IAR/AAR App
Unit Comp Card

Figure 1. Functional Diagram of STEEL-R in a Future Army STE-based Training Architecture

The STEEL-R project engaged in regular working sessions with an active US Army regiment to develop a GIFT-based after-action and analysis review (A3R) dashboard referred to as Game Master (see Figure 2). Game Master is used for controlling the execution of the synthetic exercise and displaying and analyzing the individual and team task execution outcomes. It uses existing GIFT capabilities and new features that are being added to GIFT. To test the capability of real-time task data collection and processing, a Unity-based synthetic training application for a fixed Army battle drill exercise was created. This work was conducted with the STE Cross-Functional Team. It focused on building interfacing protocols and scripting capabilities that were not tied to the US Army's Virtual Battlespace (VBS) product so that a more agnostic application interface could be developed. With this initial prototype, multiple engineering experiments were conducted representing actual Soldier touchpoints to establish a baseline real-time assessment capability. This work will continue in partnership with a future US Army training facility as part of the STEEL-R capability validation and verification process.

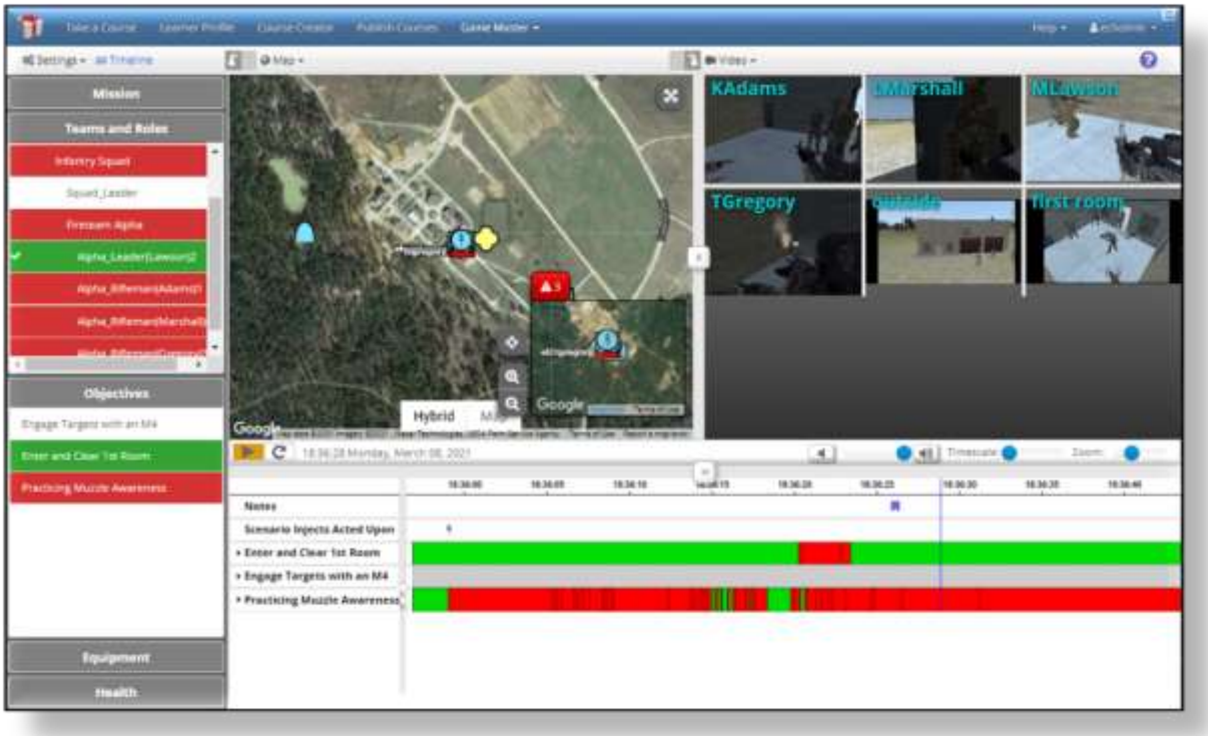


Figure 2. Game Master Dashboard

The STEEL-R based exercise demonstrated for GIFTSym is associated with a US Army Infantry Battle Drill, set within a variable experiential context. The required team and individual team-role tasks and measures are pre-selected and designed to be part of the Plan and Prepare phases of the CBEL process, which is discussed next.

The Plan, Prepare, Execute, and Assessment Process

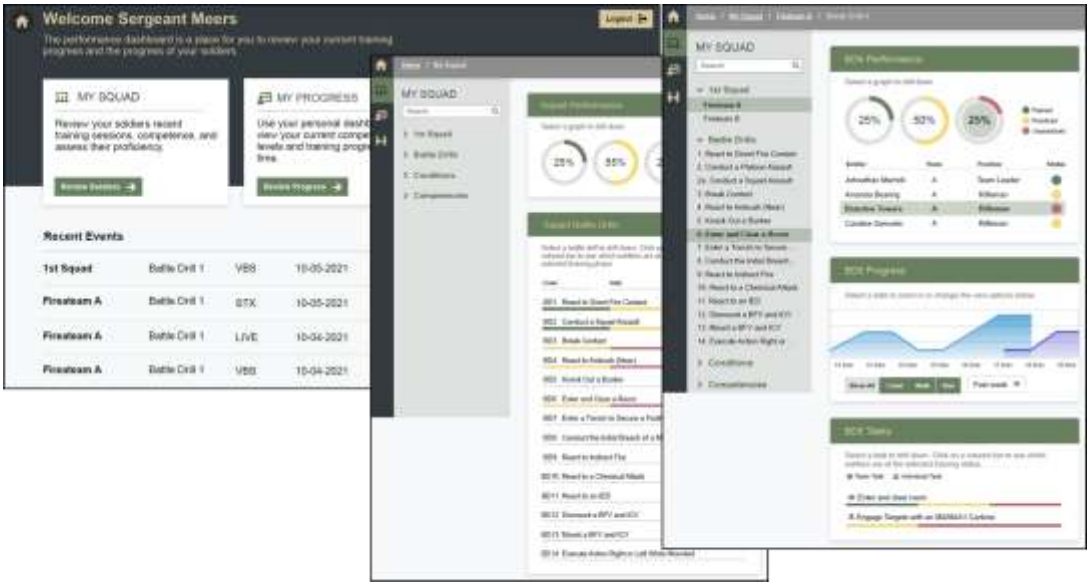
US Army Doctrine Publication 5-0 describes the Plan, Prepare, Execute, and Assess (PPEA) process as a "...workflow focused on enhancing and improving mission command by more fully incorporating the official doctrinal approach towards training and assessment of its Soldiers within a given Operational Environment (OE)" (US Army TRADOC, 2019). STEEL-R is architected to support a learning environment within a realistic OE, where training is actively conducted to build warfighting situational experiences through synthetic and live stimulus and feedback. The GIFTSym demo will demonstrate how STEEL-R is used within this PPEA process.

Each phase of the PPEA workflow revolves around the decision-making needs of the unit's commander and the ability to train the unit's individual Soldier and combat team tasks to support success on the battlefield. In STEEL-R, data about each Soldier is sampled during real-time assessment and used to produce incremental and summary overviews of trainee readiness under varying task conditions.

Plan and Prepare Elements of STEEL-R

CBEL training planning is a multivariate design and development activity. Depending on the need and source of the training requirement, a different set of experiential source data is selected through automatic indexing and filtering. This process produces a series of exercise design decision-support recommendations that target the specific competencies, tasks, and experiences of a unit team or unit Soldier (a "learning-actor"). STEEL-R is currently focused on training for platoons and below echelons (tactical small-units), a domain of Army training planning that is not well supported today. Currently, the US Army uses multiple tools within the Army Training Network to help Commanders produce training plans and strategies, but at the tactical small-unit level such tools and strategies do not exist. It is a goal of STEEL-R to provide these capabilities in a manner that will be available in the future STE-Training Management Tool (TMT).

Within STEEL-R, the concept of operations for training planning incorporates tools that provide training leader /manager decision-support in the exercise design process. The first tool is a competence dashboard or leaderboard shown in Figure 3 that also can provide feedback to a team or individual. This tool provides training leaders / managers with a means to view, compare, and select the competencies that need to be trained. Information from the dashboard and other inputs from a unit's commander will be combined with an experience index (XI). An XI maps and filters design variables that are used to configure experiential training events.



Welcome Sergeant Meers

MY SQUAD

Review your soldiers recent training sessions, competence, and assess their proficiency.

MY PROGRESS

Recent Events

1st Squad	Battle Drill 1	VBS	10-05-2021
Fireteam A	Battle Drill 1	STX	10-05-2021
Fireteam A	Battle Drill 1	LIVE	10-04-2021
Fireteam A	Battle Drill 1	VBS	10-04-2021

Figure 3. Competence Dashboard

Other elements that help define the experience decision-making process are the mission order and its subordinate fragmentary orders that cue experience-events at the task execution level. Each mission is expected to represent operating orders used in previous live exercises or real-world events. The mission defines the environment, force, and other plot contexts such as the enemy, terrain and weather, available troops or support resources, time to perform, and civil variables (known as a METT-TC format). Once the experience context is specified, the unit echelon, team(s) and/or role(s) to be trained on a target competency or task are determined. These decisions further filter down to the specific sub-competencies and tasks that can be measured within the selected mission, which is where the actual experience design process can begin.

Experience Design Tool (XDT)

As noted earlier, the STEEL-R project was expanded to include a research thread focused on developing the experiences that enable data to be collected and specify Soldiers will conduct CBEL to build competence in warfighting tasks. This is done in STEEL-R through the XDT. The XDT expands upon concepts developed in a previous US Army project called Squad Overmatch (Johnston et al., 2017). This project produced a variable stress-condition based series of classroom, synthetic, and live "lane-based" events intended to build levels of teamwork, advanced situation awareness, and combat resilience. XDT uses a similar approach in that its function is to create experiential exercises consisting of multiple mission-oriented conditional variables (including difficulty and stress) and including one to many serial or actor-triggered episodic competency/task performance prompts, referred to as Experience Events (or xEvents). xEvents can be conceived of as "competency test-questions;" they measure a target actors' ability to perform a specific targeted task / competency at specified conditions of difficulty (e.g., volatility, uncertainty, complexity, or ambiguity) and stress (cognitive load, physical load, environmental load, etc.).

What makes the XDT different from a traditional exercise design tool is that in addition to creating the selected scenario plot (i.e., the mission), it is configured to help design the various xEvents along with their associated measures and criteria. XDT also supports selecting the source data required to serve as evidence of a task's formative and summative measured outcome. As will be discussed more during the GIFT discussion, as learning-actors encounter and perform against xEvents in the synthetic environment, they produce objective performance data that is reported by sensors and/or the training environment itself. XDT also helps the designer not only test if the appropriate data is produced but test if the target task/competency is optimally being measured, and then defining an appropriately defined criteria for each resulting performance level, as it fits within an xEvent's conditions.

The XDT (using GIFT capabilities) will also design strategies that either provide actor interventions (e.g., feedback) or exercise adaptations (e.g., different artificial intelligent actor behaviors) to further develop the learning actor's experience and task-competence. It is the long-term exposure to these kinds of dynamic conditions and interactions, over many of these experiential opportunities, that enables a learning-actor to develop the ability to automatically respond to similar combat events while later performing in a live training or real-world environment. The result of these experiences, and later feedback of their performance, are what stimulates subsequent reflection by the actor (or their leaders), which further stimulates long-term experiential learning and expertise beyond what can be learned in a didactic learning environment. This process is also what allows the learning-actor to develop ideas for constructing new performance strategies to help them improve their outcomes in future dynamic xEvents focused on the same task/competency. An xEvent will usually follow a design structure similar to that shown in Figure 4 below.

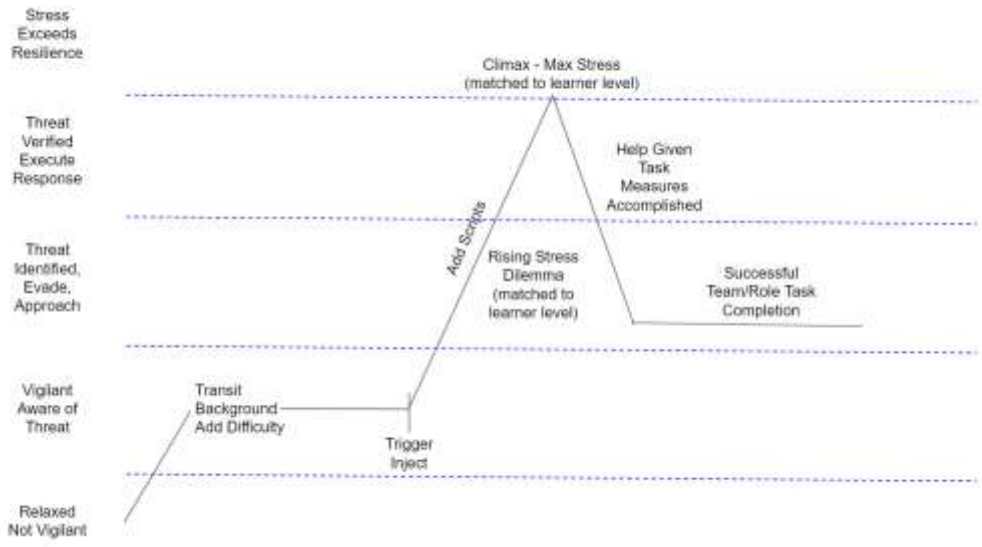

Stress Exceeds Resilience
Threat Verified Execute Response
Threat Identified, Evade, Approach
Vigilant Aware of Threat
Relaxed Not Vigilant
Climax - Max Stress (matched to learner level)
Help Given Task Measures Accomplished
Add Scripts
Rising Stress Dilemma (matched to learner level)
Successful Team/Role Task Completion
Transit Background Add Difficulty
Trigger Inject

Figure 4. General Experience Event Design Structure

The results of the XDT based design process are saved, stored, and shared within what is referred to as an experiential training support package (XTSP). An XTSP is a semantically normalized structure that can exist in the cloud or saved off as a JavaScript Object Notation (JSON) artifact. Key is that the xTSP is a machine-readable format that helps Soldiers rapidly and automatically set up complex training environments, data collection devices, as well as automate much of the re-designed real-time assessment capability provided by the GIFT architecture. In this way, the XTSP essentially provides GIFT much of the initial parameters used in its Domain Knowledge File (DKF) for its task-based assessments and strategies for a given synthetic or live training application. This includes the domain mission, the mission-based actor organization, and the competency related tasks, concepts (measures) and automated or manually data-informed measurement criteria. The XTSP also provides GIFT with the training experience difficulty and stress points used by GIFT to calculate an xEvent's overall difficulty or stress level that is reported with the assessment result via the xAPI statements GIFT produces.

The prepare phase of the PPEA process is used just before a designed XTSP is to be executed. During this stage, the actual learning-actors (teams or individuals) are selected and assigned to pre-defined mission defined teams and their respective roles. In addition, the preparation phase is when each learning-actor is assigned to a specific training environment device so the training application can report to GIFT what synthetic performance belongs to which actor. Preparation is also when the exercise mission is provided and discussed so the learning-actors understand the context, restrictions and/or support resources they must perform with. In addition, during the prepare phase, it will provide the designated training leader the option to modify or add additional parameters to the XTSP based on specific attributes of a given actor's learner-profile that may not have been designed for originally in the XTSP. Once these activities are completed, the execution phase can begin.

xAPI Profiles and DATASIM

xAPI is the primary method within the STEEL-R architecture to transfer the data describing assessments between systems. The xAPI Profiles (ADL Initiative, 2017) specification, a companion to xAPI itself, is a specification for describing the statements and patterns of xAPI data. This allows an author to describe the concepts, statement templates, and expected flow of statements in an xAPI dataset produced by a system or collection of systems.

For STEEL-R, an xAPI Profile has been written for GIFT, whereby the xAPI data emitted follows these rules (Blake-Plock et al., 2021):

- As a domain session is requested, data is cached for use in the creation of the xAPI statements
- As a domain session starts, an xAPI statement is generated which identifies the user and the course selected
- As a knowledge session is created, one or more xAPI statements indicate that either a session host created and started a session or that upon creating the session lobby, other users joined the lobby and then the host started the session
- As an updated request passed through a knowledge session, the team position of the session member provides information for use in the creation of xAPI statements
- The knowledge session begins for the team and statements are emitted
- The learner state is derived from the relevant GIFT components regarding: cognitive state, affective state, and performance state; user interaction within the course causes an update to these attributes
- As formative assessment is completed, a request is made to publish the lesson score, and summative results are recorded as xAPI statements
- The session is closed if an xAPI statement is emitted indicating that the user has exited the course

Following the recommended data flow of the TLA, xAPI data emitted from GIFT is validated and captured by a Noisy LRS — the Edge Activity Store in Figure 1 above. This xAPI data then is filtered through LRSPipe, an open-source middleware that provides the ability to govern the business logic of the data flow through the TLA by means of an xAPI Profile (https://github.com/yetanalytics/xapipe). This filtered data, governed by the master xAPI Profile is then forwarded and made available to an LRS in the transactional layer — the Cloud Activity Store as described above. The data collected through this process of filtering is used by CaSS as immutable evidence in the assertion of competencies.

One of the uses of the Data and Training Analytics Simulated Input Modeler (DATASIM) in the context of the STEEL-R project is to quickly model changes to xAPI datasets and to evaluate the design of the xAPI Profiles used by the system. Because xAPI Profile modifications can be performed and resulting data can be simulated very quickly using DATASIM, we have been able to rapidly evaluate the effects of hypothetical changes to GIFT xAPI data on CaSS assertions without performing code changes on source systems beforehand. This is because DATASIM generates synthetic statements that reflect those changes and sends it to CaSS in much the same way GIFT produces statements from active training sessions with learners.

DATASIM is also used to test components of the system in isolation and evaluate system stability at scale. DATASIM is capable of generating tens of thousands of xAPI statements per second, which can result in datasets that can be used to evaluate system performance at or above realistic maximum production scale conditions.

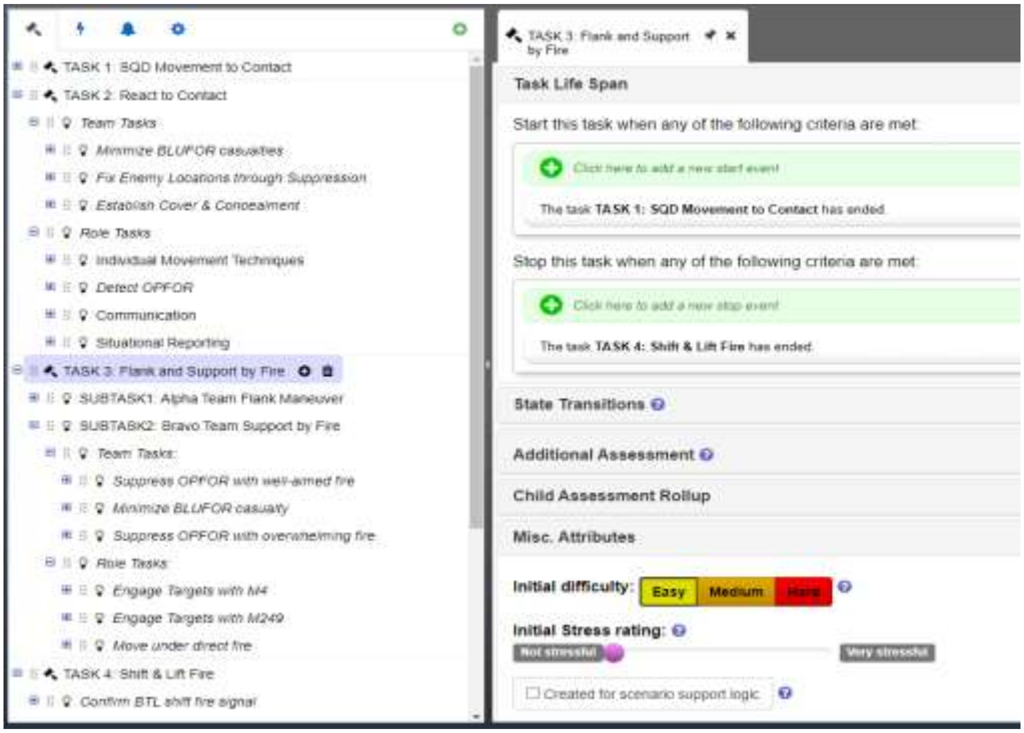
Execute and Assess Elements of STEEL-R

In STEEL-R's process, each action by each participant is captured and stored for further analysis. This allows the OC/T to discern adjustments that must be made and how their plans must be modified in real time or in an after action review (AAR).

To facilitate this, commanders are provided with easily digested real-time assessments of individual and team performance and competencies. Assessment results enhance the commander decision making and help commanders and the staff keep pace with constantly changing situations.

GIFT

In the STEEL-R solution, GIFT (Sottilare et al., 2012) is used to produce evidence of training experiences in the form of xAPI statements and session logs generated from the learning experiences that are being executed.



- TASK 1: SQD Movement to Contact
- TASK 2: React to Contact
  - *Team Tasks*
    - *Minimize BLUFOR casualties*
    - *Fix Enemy Locations through Suppression*
    - *Establish Cover & Concealment*
  - *Role Tasks*
    - Individual Movement Techniques
    - *Detect OPFOR*
    - Communication
    - Situational Reporting
- TASK 3: Flank and Support by Fire
  - SUBTASK1: Alpha Team Flank Maneuver
  - SUBTASK2: Bravo Team Support by Fire
    - *Team Tasks:*
      - *Suppress OPFOR with well-aimed fire*
      - *Minimize BLUFOR casualty*
      - *Suppress OPFOR with overwhelming fire*
    - *Role Tasks*
      - *Engage Targets with M4*
      - *Engage Targets with M249*
      - *Move under direct fire*
- TASK 4: Shift & Lift Fire
  - *Confirm BTL shift fire signal*

TASK 3: Flank and Support by Fire

Task Life Span

Start this task when any of the following criteria are met:

Click here to add a new start event

The task **TASK 1: SQD Movement to Contact** has ended.

Stop this task when any of the following criteria are met:

Click here to add a new stop event

The task **TASK 4: Shift & Lift Fire** has ended.

State Transitions

Additional Assessment

Child Assessment Rollup

Misc. Attributes

Initial difficulty: Easy | Medium | Hard

Initial Stress rating: Not stressful — Very stressful

☐ Created for scenario support logic

Figure 5. Screenshot of the GIFT DKF authoring tool with the React to Contact related tasks and concepts.

GIFT accomplishes this by first aligning its assessment model to the experiences outlined in the xTSP via the DKF. The DKF shown in Figure 5 contains the tasks and concepts related to a React to Contact scenario. Notice that 'TASK 3: Flank and Support by Fire' is selected and that some of its attributes are shown, including when the task should start or end and the initial difficulty and initial stress ratings of the task. After the DKF is created, GIFT works with the integrated training application (e.g. Virtual Battlespace, SE Sandbox, RIDE) to assess the various actions of the learners under different tasks and conditions.

The OC/T can utilize GIFT's Game Master interface to monitor the ongoing assessments, override Artificial Intelligence (AI) assessments and provide observed assessments as needed during the training. As the training unfolds, GIFT frequently updates its representation of the individual and team learner state. These updates are delivered to a LRS via xAPI statements that conform to the GIFT xAPI profile. This profile follows TLA standards and best practices.

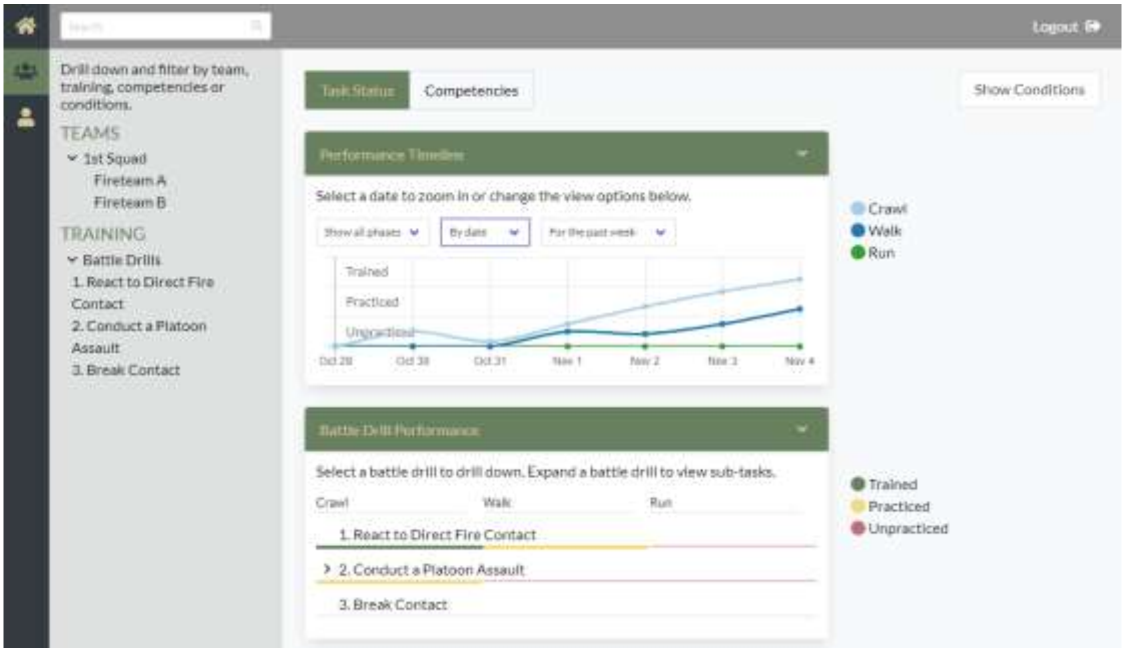
When the training scenario is completed, GIFT produces summative xAPI statements that contain the overall assessment. This is generated automatically using production rules. The OC/T can then use the Game Master Past Session interface to make final assessment decisions that can further augment existing xAPI statements and produce new xAPI statements. The Game Master can be used to conduct an AAR as well using a timeline to playback, synchronize numerous data streams, and customize the delivery of key events to the target audience. At the end of the training session, a robust, supervised, evidence-based data set exists that CaSS can then use for readiness and talent tracking.

CaSS

CaSS provides a key capability to STEEL-R by gathering evidence, authoring and storing competency frameworks, and generating assertions of competency based on the gathered evidence. These assertions are presented via dashboards or as data to other systems so observers or other trusted agents can judge the effect on an individual or team's proficiency. CaSS has been evolving since 2016 via the Advanced Distributed Learning Initiative (ADL Initiative, 2017), US Navy (Gafford et al., 2019), US Army (Goldberg et al., 2021), and US Air Force (US DoD ADL Initiative, 2020) investment, and is now a comprehensive, reusable, open-source system for incorporating competency-based learning in a training or education ecosystem (ADL, 2019).

In STEEL-R, CaSS stores competency frameworks that represent the knowledge, skills, abilities, and other attributes related to the training exercises that STEEL-R manages. CaSS receives GIFT-generated xAPI statements from an LRS and translates these into assertions about the competencies held or demonstrated by individuals and units. These are used to estimate competency states, which in turn are displayed in a dashboard during the Assess phase of a training event. The model used to estimate competency states takes factors such as repetition and skill decay into account and is being modified to require that competency be demonstrated under a variety of conditions that induce stress or add difficulty to a task. The goal is to provide a tool to STE that can be used to optimize the effectiveness and efficiency of training interventions that support squad development. All data produced by CaSS are available through APIs so that CaSS and STEEL-R can be used in a Multi-Open System Architecture.

The current STEEL-R dashboard, depicted in Figure 6, is built to expose the assertions identified in CaSS. The STEEL-R dashboard is built with the Vue javascript framework allowing for rapid and iterative development as requirements and designs evolve. The dashboard was designed with scalability and performance in mind to be able to handle large amounts of data flowing into it from CaSS, and also with extensibility such that it could be further modified to display additional data as it becomes available.



Drill down and filter by team, training, competencies or conditions.

Select a date to zoom in or change the view options below.

Select a battle drill to drill down. Expand a battle drill to view sub-tasks.

Figure 6. STEEL-R Dashboard

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

STEEL-R's capability to homogenize data from a range of sources and to estimate learning gains over time is intended to have a positive impact on experiential US Army training. In the third year of STEEL-R, the team will coordinate with US Army units to collect data from a range of synthetic and semi-synthetic training experiences. This data will be used to verify the STEEL-R architecture and models, and to mature dashboard interfaces.

During evaluation of the system and discussion with Subject Matter Experts (SMEs), it was identified that STEEL-R could be operated by a variety of training personnel, from the personnel engaging in the training themselves, to an OC/T, to a set of training staff as a part of a highly structured formal training event. STEEL-R's design follows PPEA, which aligns with the current Army model for training. However, STEEL-R is not limited to the PPEA paradigm and can be used as a standalone scenario design tool or a means to collect data and assess competency and readiness in conjunction with other STE training regimens.

To enable this flexibility, it is necessary for STEEL-R to support the rapid execution of pre-canned scenarios, to enable scenarios to be constructed as part of training event planning, and to enable scenarios to be rapidly developed over short periods of time between training days. To this end a library of reusable experiences in the form of xEvents are linked to scenarios that implement them. In addition, these xEvents are associated with measures and competencies that are used to elicit performance. While some of this linked data resides in CaSS, it is advantageous to store and serve experience and scenario data separately. This is done using the XI which has been expanded to include experiences at multiple levels of granularity together with conceptual, instantiated, and instrumented experiences. This evolution of the XI concept requires further research and development.

REFERENCES

ADL Initiative. (2017). Experience API Specification. Retrieved from https://github.com/adlnet/xAPI-Spec

ADL Initiative. (2019). Total Learning Architecture. Retrieved from https://adlnet.gov/publications/2020/04/2019-Total-Learning-Architecture-Report/

Gafford, W., Kitchens, J., & Ray, F. (2019). Use of Natural Language Processing NLP to Extract Technical Competency Frameworks from Maintenance Task Analyses (MTA), Interservice/Industry Training Simulation and Education Conference.

Goldberg, B (2020). STE Experiential Learning-Readiness (STEEL-R) Strategy. STTC White Paper.

Goldberg, B., Owens, K., Hellman, K., Robson, R., Blake-Plock, S., Hoffman, M. & Gupton, K. (2021). Forging Proficiency and Readiness through an Experiential Learning for Readiness Strategy. In Proceedings of the 2021 Interservice/Industry Training Simulation and Education Conference (I/ITSEC). Orlando, FL.

Johnston, J, Townsend, L, Gamble, K, et al. (2017). Squad Overmatch (SOvM) Phase II Final Report. Army Research Laboratory.

Kolb, A.Y., & Kolb, D.A. (2017). The Experiential Educator. EBLS Press, Kaunakakai, HI.

Owens, K.P. , & B. Goldberg (2022). Competency-Based Experiential-Expertise. Design Recommendations for Intelligent Tutoring Systems, Volume 9, Chapter 3 · Feb 22, 2022.

Owens, K.P., et.al. (2020). Re-Thinking the Tactical Small Unit Synthetic Training Model. Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC).

Robson, R., Goldberg, B., Blake-Plock, S., Owens, K., Ray, F., Purviance, A. Hoffman, M., Hernandez, M, Hoyt, & W. (2022). Mining Artificially Generated Data to Estimate Competency. To appear in proceedings of Educational Data Mining 2022, Durham, UK.

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

US Army PEO-STRI (2005-2018). Games-for-Training. https://www.peostri.army.mil/games-for-training-gft

US Army TRADOC (2019). TP 350-70-1 and ADP 5-0.

US Army TRADOC (2021). FM 7-0.

US DoD, ADL Initiative. (2020). Competency Framework Development Process Report. Ft. Belvoir: Defense Technical Information Center.

ABOUT THE AUTHORS

***Kevin Owens** is an Engineering Scientist in Modeling and Simulation at the Applied Research Laboratories: The University of Texas at Austin (ARL:UT). Following a career in the US Navy, Kevin has been conducting learning engineering, and applied research in new warfighter training technologies and learning models for 21-years. Currently Mr. Owens is a member of the US Army PEO-STRI engineering team for the future Synthetic Training Environment (STE) program. He is also a principal researcher for the U.S. Army Combat Capabilities Development Command (DEVCOM) Soldier Center, Simulation and Training Technology Center (STTC) STEEL-R project.*

***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. Robby Robson *is a researcher, entrepreneur, and standards professional who is co-founder and Chief Science Officer of Eduworks Corporation, a member of the IEEE Standards Association Board of Governors, and Principal Investigator on the STEEL-R project. He has made contributions to areas ranging from semi-algebraic geometry and computational number theory to web-based learning, digital libraries, and applications of AI to learning, education, and training. His recent efforts focus on competency-based approaches to talent management and experiential learning and providing effective and equitable reskilling opportunities for current and future workers. Robby holds a doctorate in Mathematics from Stanford University.*

Michael Hoffman *is a senior software engineer at Dignitas Technologies and the technical lead for the GIFT project. For over a decade he has been responsible for leading the engineering of GIFT, collaborating with the ITS community, and supporting ITS related research. Michael manages and contributes support for the GIFT community through various mediums including the GIFT portal (www.GIFTTutoring.org), annual GIFT Symposium meetings and technical exchanges with Soldier Center and their contractors. He is also the Project Manager on the Flexible and Live Adaptive Training Tools (FLATT) project which is providing a new and intuitive way to leveraging GIFT and the technical lead on helping to integrate GIFT into TSS/TMT.*

Fritz Ray *is the Chief Technology Officer (CTO) at Eduworks Corporation. He has spent his 15 year career architecting, designing and leading development of software used by the US Advanced Distributed Learning(ADL) Initiative, the US Army Research Laboratory, the US Navy, the US Air Force and industry customers in the fields of aviation, financial services, and intellectual property. He currently contributes to several open-source projects, standards efforts surrounding learning engineering, and task forces seeking to understand and model the learning experience in software. He has a strong background in E-learning including training systems, practical aspects of cyber-security, information technology and artificial intelligence.*

Alex Colburn *is an Instructional Designer and Analyst at Eduworks Corporation. An Army veteran and former university ESL instructor, he has assisted with competency-based learning research and analysis efforts for projects with the US Army, the US Navy, the US Marines, the US Air Force, the National Park Service, and the National Cooperative Extension's Impact Collaborative.*

Mike Hernandez *is the Director of Business Development at Eduworks and acts as a contributor on the STEEL-R Project. He brings over a decade of Defense experience and has been a contributor to the Total Learning Architecture (TLA) effort since 2016. Throughout his career, he has stewarded the technical and programmatic development of multiple projects related to competency-based learning, performance tracking, and data analytics for the US Navy, US Army, US Air Force, and OSD.*

Mile Divovic *is a Software Engineer at Eduworks Corporation. He holds a degree in Computer Science from New York University and has worked on multiple xAPI and competency-based software projects with the Department of Defense, the National Park Service, and the National Cooperative Extension's Impact Collaborative.*

Shelly Blake-Plock *is CEO of Yet Analytics. Key projects include the implementation of an xAPI-enabled platform for the U.S. Air Force under a program sponsored by the Air Force Research Laboratory as well as the design and development of the Data and Training Analytics Simulated Input Modeler (DATASIM) for the Advanced Distributed Learning Initiative. Shelly is a Senior Member of the IEEE and is an officer of the Learning Technology Standards Committee (LTSC) where he chairs the P9274.4.2 Working Group on Cybersecurity for xAPI as well as the Technical Advisory Group on xAPI.*

Cliff Casey *is the CTO of Yet Analytics, Inc. He obtained a Degree in Computer Engineering from McGill University and has over 15 years of software engineering, system architecture, and technical leadership experience including hiring, training, and coordinating large software development teams, vendors, contractors, infrastructure teams and QA resources. His technical experience is primarily in architecting, building, and deploying enterprise-scale applications and data solutions, both on-premises and in the Cloud. As the technical leader of Yet Analytics he has overseen and contributed to the development of the Data and Training Analytics Simulated Input Modeler (DATASIM), the Data Analytics and Visualization Environment for xAPI (DAVE), and the Open Source SQL LRS and LRSPipe projects.*

A Review of GIFT Network Interface Paradigms and Potential Future Directions

Christopher H. Meyer[1], Nicholas Roberts[2], Mike Kalaf[1], Zach Heylmun[1], and Anne M. Sinatra[3]

Synaptic Sparks, Inc.[1], Dignitas Technologies[2],
US Army Combat Capabilities Development Command (DEVCOM) - Soldier Center[3]

INTRODUCTION

This paper focuses on the Generalized Intelligent Framework for Tutoring's (GIFT's) complex network communication paradigms. There are a number of lessons learned by the authors from their work with GIFT, and with integrating GIFT with network architectures such as those provided by the Google Web Toolkit (GWT); a core software framework that GIFT utilizes which supports a variety of different communication methods. Some of the communication methodologies that GWT includes, are Remote Procedure Calls (RPC), web sockets, and HTTP operations. GIFT has also implemented networking solutions with Apache's ActiveMQ and Kafka applications, specific Transmission Control Protocol (TCP)/User Datagram Protocol (UDP) socket-based messaging, and with external hardware/firmware/sensors. GIFT, as it stands at the time of this writing, currently uses the above network interface solutions to communicate with a multitude of external software suites in support of the US Army's Synthetic Training Environment Experiential Learning for Readiness (STEEL-R) (Goldberg et al., 2021) community. The combination of GIFT and external software systems such as the University of Southern California's (USC's) Rapid Integration and Development Environment (RIDE), the Eduworks Competency and Skills System (CaSS), and the YetAnalytics xAPI Data Visualizations (DAVE) project offer additional functionality integrated together when compared to the individual systems alone. Other examples of external software suite categories that GIFT interfaces with include support software systems such as resource health monitoring services, learning management systems, learner record stores, and select Internet of Things (IoT) devices.

Based on the review of existing technologies, it is possible that implementing a generalized networking solution in GIFT, or preemptively creating middleware translation utilities, may enhance the ability to easily integrate with future educational technologies with little risk or development effort.

Background Information and GIFT APIs

GIFT is a highly flexible Intelligent Tutoring System (ITS) framework. It is not limited to one domain topic area, and its modular design allows for reuse (Sottilare et al., 2017). The GIFT framework includes several types of interfaces, notably RPC Application Program Interfaces (APIs), GWT communication protocols, and the Gateway module. All of these communication protocols are utilized in specific GIFT use cases. For instance, one of the Gateway Module's responsibilities is to enable communication with software that is involved in active training scenarios such as Unity or Virtual Battlespace 3 (VBS3). This section continues the explanation of the above protocols and more, and provides information on a number of currently available interface concepts that GIFT fully supports.

GIFT and the Google Web Toolkit (GWT)-Provided Client and Server APIs

The GIFT architecture relies on many modern interfaces in order to both produce and consume network messages. Following GIFT's beginning use of RPC communication methods, another primary communication method that GIFT employed was a paradigm provided by GWT software. The concept of Create, Read, Update, and Delete (CRUD) served as an initial methodology by which GIFT interfaced with users' training content. These interactions between users and training content in GIFT were enabled primarily via server-to-client communication APIs in GIFT's instantiation of GWT software. The services that are implemented using GWT provide many interfaces into and out of GIFT software modules, which are reached by user interactions via a variety of functions in the web client.

For the rest of this section we will focus only on GWT interfaces that GIFT has active implementations for, understanding that there are far more communication protocols available in GWT for future use. Other layers of communication will be described in the following subsections.

One of the two primary protocols provided by GWT that GIFT utilizes is the Hypertext Transfer Protocol (http / https). GIFT uses the Hypertext Transfer Protocol to communicate with the GIFT server from browser-based clients (such as Chrome or Firefox) to name one communication path. The HTTP protocol is primarily used by a GIFT client to create synchronous communication paths with the server, in order to perform any respective CRUD operation mentioned above between the user and the content they are accessing (2022 GIFT Developer's Guide, 2021).

A second primary protocol that GIFT utilizes in GWT is that of Web Sockets. While anyone familiar with Web Sockets can note some overlap in capability with HTTP interfaces, GIFT can use web sockets to initiate server requests with parameterized data. This capability is more naturally in-line with web socket functionality, and thus GIFT can freely alternate between HTTP and web socket protocols depending on the nature of the client-server communication requirement (2022 GIFT Developer's Guide, 2021).

Of final note; both of these message protocols utilize the TCP connection types after the formation of the initial HTTP or web socket message. TCP allows for nearly any message that can be converted to bytes to be sent, lossless and reliably, between networked systems.

Additional detail about GWT in general can be found at:
https://www.gwtproject.org/doc/latest/DevGuideServerCommunication.html

Additional detail about how GWT is implemented in GIFT can be found at:
https://www.gifttutoring.org/projects/gift/wiki/Developer_Guide_2022-1#GWT

GIFT and External Training Application Interfaces

GIFT interfaces with a number of different external training applications. This is a feature that is fairly unique about GIFT – it can be used for intelligent tutoring in existing programs such as simulation based games. This feature requires a gateway between the external training application and GIFT. Some have already been created, and others can be created by those with a knowledge of computer programming. Assessments and adjustments to the real-time gameplay can occur based on individual performance in the game VBS3. Similar simulation based games can be integrated with GIFT as well. Other external training applications GIFT is currently integrated with include PowerPoint, Unity, and the TC3Sim game. Adaptive training using these external training applications can be authored in the GIFT authoring tool by dragging already existing course objects for them onto the main timeline. Figure 1 below is a current architecture diagram that illustrates these concepts of GIFT communicating with external applications and systems.

While most of the communication that GIFT performs with external training applications is done using TCP or UDP sockets managed by the Gateway module, Figure 1 depicts a use case with Test Harness hardware that utilizes an ActiveMQ message bus rather than direct TCP socket operations. The full list of configurable settings for certain training applications is included with a 'default.interopConfig.xml' file included with a standard GIFT server deployment setup (2022 GIFT Configuration Settings, 2021).

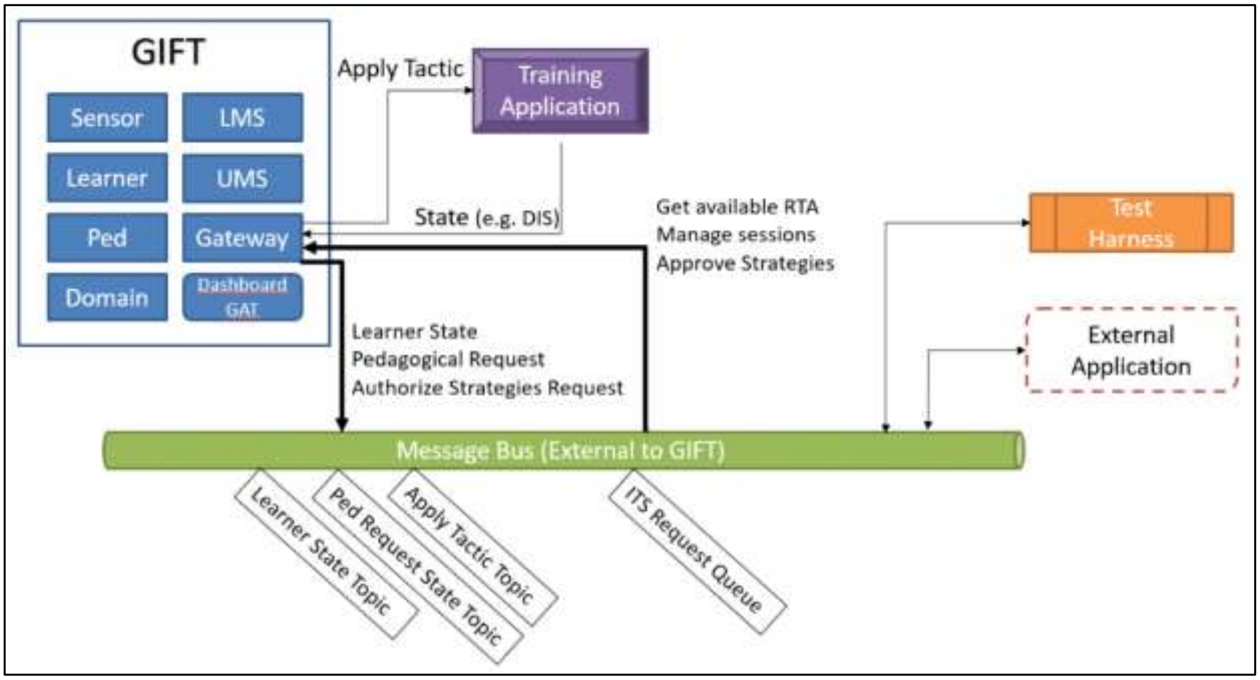


Figure 1. Adaptive Learning Service API. Reprinted with permission from Hoffman et al., 2021.

Further information on Training Applications and GIFT can be found at: https://www.gifttutoring.org/projects/gift/wiki/Configuration_Settings_2022-1#Training-Applications

GIFT Wrap

GIFT Wrap is one of the capabilities provided by the GIFT software suite, and was created with the goal of developing a user-friendly tool for authoring individual adaptive training content (Davis et al., 2018). As the reader is likely aware, this need for adaptive tutoring systems such as GIFT to not only consume varied training content in different formats is tightly coupled with the need to be able to create and edit training content that is also compatible with the hypothetical training application.

While there are some existing challenges to address, overall the inclusion of GIFT Wrap with the official release of GIFT has allowed content creators to better-author training content utilizing external applications such as Unity, as well as provide one of the necessary technical pieces to integrate with mobile hardware that was used in LandNavHD (a land navigation training application).

Additional details about GIFT-Wrap can be found in Davies et al., 2018 and at: https://www.gifttutoring.org/projects/gift/wiki/Developer_Guide_2022-1#Adding-External-Applications-to-GIFT-Wrap

LMS / UMS Interface Methods

GIFT also provides interfaces for various Learning Management Systems (LMSs) and User Management Systems (UMSs). GIFT uses SQL-based databases to track learner performance, course progression, assessments, and many other forms of user- and course-centric information. The LMS and UMS modules in GIFT are configured similarly to other modules, in that GIFT is able to send data relating to potentially multiple LMSs and UMSs depending on configuration file settings.

In addition to industry standard LMS and UMS interface capabilities, GIFT also provides connection options to a Learner Record Store (LRS). By providing this connection option, GIFT is able to support Experience API (xAPI). As of this writing, GIFT has been tested with the Advanced Distributed Learning (ADL)'s open source LRS, and WaxLRS. Other LMS/UMS/LRS software suites that GIFT has been integrated with to some level include Moodle, Blackboard, and Canvas.

Additional information about the UMS Database in GIFT can be found at:
https://www.gifttutoring.org/projects/gift/wiki/Configuration_Settings_2022-1#UMS-Database-Connection-Configuration

Additional information about the LMS Module settings in GIFT can be found at:
https://www.gifttutoring.org/projects/gift/wiki/Configuration_Settings_2022-1#LMS-Module-LMS-Connections-Settings

GIFT and External Sensor Integration Methods

As of this writing, GIFT currently offers integration with several external sensors listed in the gifttutoring.org Documents-tab website. Exploring all past sensor integrations is beyond the scope of this paper, but more information is available through examining past implementations that are publicly available in the GIFT code baseline for contributors. If there is not a prior example to use as a baseline for any new sensor integration, then more information on how to integrate new sensors may be obtained by following this link:

https://www.gifttutoring.org/projects/gift/wiki/Developer_Guide_2022-1#Integrate-a-Sensor

XML-RPC APIs

Ajax (originally an acronym for Asynchronous JavaScript And XML) is a programming practice built around using XMLHttpRequests to update a page's elements dynamically. In Java-based servers such as those GIFT implements, Ajax requests are typically handled via servlets. For the version of GWT that GIFT implements, the RPC mechanism is used to perform Ajax within GWT operations.

Fully exploring XML-RPC operations in GIFT can be accomplished by searching files for java classes extending the RemoteServiceServlet class provided by GWT. Alternatively, classes that have the same filename as the server-side files mentioned above but with a suffix of 'Async' designate a client-side version of the corresponding RPC service name.

An example that illustrates basic GIFT RPC interfaces and how to configure the XML RPC Python Server with GIFT is provided at:

https://www.gifttutoring.org/projects/gift/wiki/Developer_Guide_2022-1#Configuring-the-XML-RPC-Python-Server-With-GIFT

xAPI Syntax and Grammar Vocabularies in GIFT

As xAPI-compatible messages are a current area of study and development in GIFT, additional online resources will provide the most up-to-date information.

Additional information on xAPI interfaces in GIFT is available at:
https://www.gifttutoring.org/projects/gift/wiki/XAPI_Statements_2022-1

GIFT and Overloading ActiveMQ Servers, and REST Interfaces

As mentioned in previous sections, GIFT maintains ActiveMQ servers during operation and can be configured to provide and interact with external software applications via RESTful interfaces. One such example of GIFT operating with an external software application was in the entertainment industry with a high-tech escape room (Sayer, 2016). Developers of the escape room technology integrated the escape room software with similarly flexible communication protocols. By using available GIFT interfaces, the escape room software was able to inject direct performance data into GIFT's ActiveMQ topics, and create new scenario-specific topics that GIFT could reference during a "lesson." As the escape room lesson continued, GIFT was able to monitor and respond to participant success and failure. Based on GIFT's assessment of the performance as defined by course creators, the escape room software offered interfaces to GIFT to be able to adapt scenario difficulty, onboarding processes, and offer additional hints.

While this method of integration between GIFT and external training applications was not common, it showed that external training applications that have active software developers can use quick, modern, or direct connection methodologies to communicate with GIFT with little development cost in cases where new communication protocol experiments are desired.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

As explained above, GIFT offers a complex variety of communication protocol options that any developer may utilize. As GIFT has continued to mature and as work supporting the STEEL-R mission progresses, interface design and definition will continue to evolve. Based on the authors' knowledge of industry, software frameworks should allow for integration with an ever-increasing number and types of software, hardware, and firmware or risk complete obsolescence given time. By continuing to focus development efforts on GIFT networking, integrating GIFT with complimentary software frameworks such as CaSS and RIDE, and understanding military training requirements' impact on software communication paradigms, GIFT should be able to maintain and potentially improve its proven effectiveness as a Department of Defense (DoD) Adaptive Instructional System (AIS), and be of even greater value to the diverse range of GIFT users.

ACKNOWLEDGEMENTS

This material is based upon work supported by the US Army Combat Capabilities Development Command Soldier Center (DEVCOM SC), Training and Simulation Division (TSD), SFC Paul Ray Smith Simulation and Training Technology Center (STTC) under contract No. W912CG-22-C-0015. 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 DEVCOM SC-TSD-STTC.

REFERENCES

2022 GIFT Configuration Settings. GIFTTutoring.org. (2021, December 16). Retrieved May 15, 2022, from https://www.gifttutoring.org/projects/gift/wiki/Configuration_Settings_2022-1.

2022 GIFT Developer's Guide. GIFTTutoring.org. (2021, December 16). Retrieved May 15, 2022, from https://www.gifttutoring.org/projects/gift/wiki/Developer_Guide_2022-1.

Davis, F., Riley, J., Goldberg, B. Extending GIFT Wrap to Live Training (2018). In Proceedings of the Seventh Annual GIFT Users Symposium (GIFTsym7) (Paper #5). US Army DEVCOM–Soldier Center.

Goldberg, B., Owens, K., Gupton, K., Hellman, K., Robson, R., Blake-Plock, S., & Hoffman, M. (2021). Forging Competency and Proficiency through the Synthetic Training Environment with an Experiential Learning for Readiness Strategy. In *the annual Proceedings of the Interservice/Industry Training, Simulation, and Education Conference [CD-ROM], Orlando, FL. Arlington, VA: NTSA*.

Sayer, H. (2016, August 3). Ex-Lockheed Martin employees launch tech-driven escape room in Avalon Park. *Orlando Weekly*. https://www.orlandoweekly.com/arts/ex-lockheed-martin-employees-launch-escape-room-in-avalon-park-2513578

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*, 1-19.

ABOUT THE AUTHORS

***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 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 together 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 before co-founding a 501c3 Nonprofit Organization called Synaptic Sparks and supporting the GIFT program.*

***Nicholas Roberts** received his Bachelor of Science degree in Computer Science from the University of Central Florida. After graduation, he began working at Dignitas Technologies as part of its GIFT team and has worked on the GIFT program for the last 9 years. He has worked on building and maintaining several of GIFT's websites, managing GIFT's internal networking, communicating between GIFT and external training applications, creating GUIs to control GIFT's modules, and coordinating testing for official software releases.*

***Michael 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.*

***Zach Heylmun** graduated from the University of Florida with a degree in Digital Arts and Science engineering. After graduation, he worked for Lockheed Martin on low-level, high performance graphics as well as virtual reality rendering for flight simulation. Zach currently supports the GIFT program as he manages his co-founded company Voidstar Solutions, as well as helps to support Synaptic Sparks, a 501c3 charity dedicated to STEM education. Through a combination of efforts, both for- and nonprofit, he has worked on web technologies, mobile applications, and Amazon Web Service (AWS) server infrastructure in support of GIFT.*

***Dr. Anne M. Sinatra** is a Research Psychologist at US Army Combat Capabilities Development Command (DEVCOM) Soldier Center. She received her Masters and Ph.D. in Applied Experimental and Human Factors Psychology from the University of Central Florida. Dr. Sinatra has been on the GIFT team since 2012, and her research focuses have included team tutoring, educational psychology, and cognition.*

THEME VI: MEASUREMENT AND ASSESSMENT

FLATT: A New Real-Time Assessment Engine Powered by GIFT (2022 Edition)

Steven Harrison[1], Mitchell Gorsd[1], Michael Hoffman[1], Kimberly Pollard[2], and Benjamin Files[2]
Dignitas Technologies[1], U.S. Army Research Laboratory (ARL)[2]

INTRODUCTION

The Flexible and Live Adaptive Training Tool (FLATT) builds upon the benefits and advantages provided by the Generalized Intelligent Framework for Tutoring (GIFT) (ARL GIFT, 2022) as a third-party assessment engine that enables trainers and researchers (operators) to rapidly prototype and evaluate rule-based, real-time adaptive interventions. Operators require an intuitive interface to capture and visualize information for learner states, and to then act on that data to adapt the training in real-time. FLATT is able to accomplish these adaptations (i.e., trainee state-driven customizations [TSCs]) by allowing the operator to rapidly author rules that specify *when* and *how* to adapt a virtual training environment (VTE) during a live session. *When* to adapt a VTE can depend on one or more independent sources of data, such as human-worn sensors, VTE states, trainee behaviors, and survey results. *How* to adapt a VTE varies for each VTE; only features that the VTE currently supports and provides API (Application Programming Interface) access to can be utilized. The previous publication (Harrison & Burmester, 2021) discussed the domain analysis research decisions and the design for the FLATT architecture. Over the past year, the FLATT application started development based on those decisions and has gone from a purely theoretical concept to a viable application. In this paper we will discuss how GIFT can provide its supported data sources and supported VTE TSCs to FLATT for authoring, how the authoring tool is designed to maximize performance and ease-of-use, how FLATT operates and communicates with GIFT during a live session, and the live session services provided to GIFT by FLATT. This paper will also provide a use-case scenario in which FLATT is used to supplement a GIFT domain knowledge file (DKF) and adapt a scenario in real-time.

FLATT ARCHITECTURE

FLATT was designed to be modular so each piece could be reused to save the author time and simplify the authoring experience. In order to fully comprehend the information presented in this paper, the application's modular hierarchy needs to be understood.

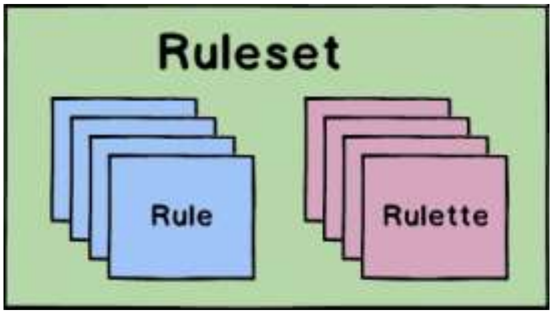


Figure 1. Ruleset Hierarchy

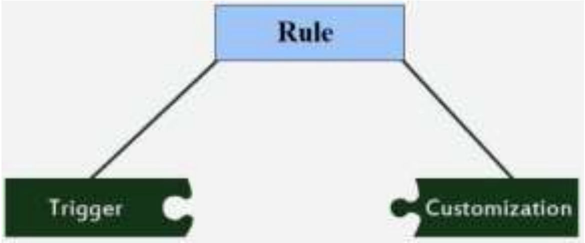


Figure 2. Rule Components

Ruleset Hierarchy

Ruleset – This is the top-most level in the hierarchy (**Error! Reference source not found.**). A ruleset is a c ollection of authored rules and rulettes. Each ruleset requires a Service Definition indicating what capabilities are possible within the ruleset authoring tool and during a live session.

Rule – This is used to determine when and how to adapt a training environment. Each authored rule consists of a trigger (when) and a customization (how) as seen in **Error! Reference source not found.**. E.g., One s uch rule might be: *When* the player's heart rate exceeds 150% of baseline, change in-game weather to sunny (*how*).

Trigger – The first half of the rule (**Error! Reference source not found.**). Determines when a rule should b e executed. For example, to execute a change in the VTE when the trainee reaches a certain location or attains a certain physiological state. The Service Definition defines what conditions can be used to make up the trigger.

Customization – The second half of the rule (**Error! Reference source not found.**). Specifies how to adapt t he training environment. For example, additional adversaries might be spawned in the VTE, guideposts might illuminate, or a weapon jam might be simulated. The Service Definition defines which TSCs are supported.

Rulette – This is an authored piece of a trigger designed to be reused by multiple rule triggers. This allows the operator to create reusable conditions that can then be quickly added to different rule triggers without having to duplicate the work, saving valuable time.

AUTHORING TOOL

FLATT provides a powerful and straightforward authoring tool to the operator. In FLATT, the operator uses a robust if/then format via an intuitive graphical user interface to account for any number of conditions and to designate a unique situation in which the desired VTE adaptations will be applied. When authoring the rule, chart objects can be dragged and dropped onto the chart and quickly connected to allow for rapid prototyping. This builds a logical *flow* of data and conditions to form the trigger, which can be seen in Figure. Authoring the customization is just as intuitive with placing the adaptation events into a linked sequence, which can be seen in Figure 2. The user can easily visually assess the steps involved in triggering a desired customization. Each chart object has specific input and output types that are uniquely identified with color and shapes; text connects to text, numbers connect to numbers, etc. These specifications make it easy for the operator to keep track of each object's requirements at a glance and reduces the possibility for user error in connecting incompatible triggers and rules. Since chart objects can have different type inputs and outputs, and connections can only be made between them if they are the same type, it could be overwhelming for users to find potential connection points. Fortunately, FLATT makes this easy. While creating the connections between chart objects, the FLATT authoring tool intelligently focuses on available possible connection points to increase the ease of authoring. To further improve the authoring experience, the left editor panel's view can be toggled to show the options as icons or textual lists. All the features built into the FLATT authoring tool serve to provide an easy-to-use interface that allows for rapid rule prototyping.

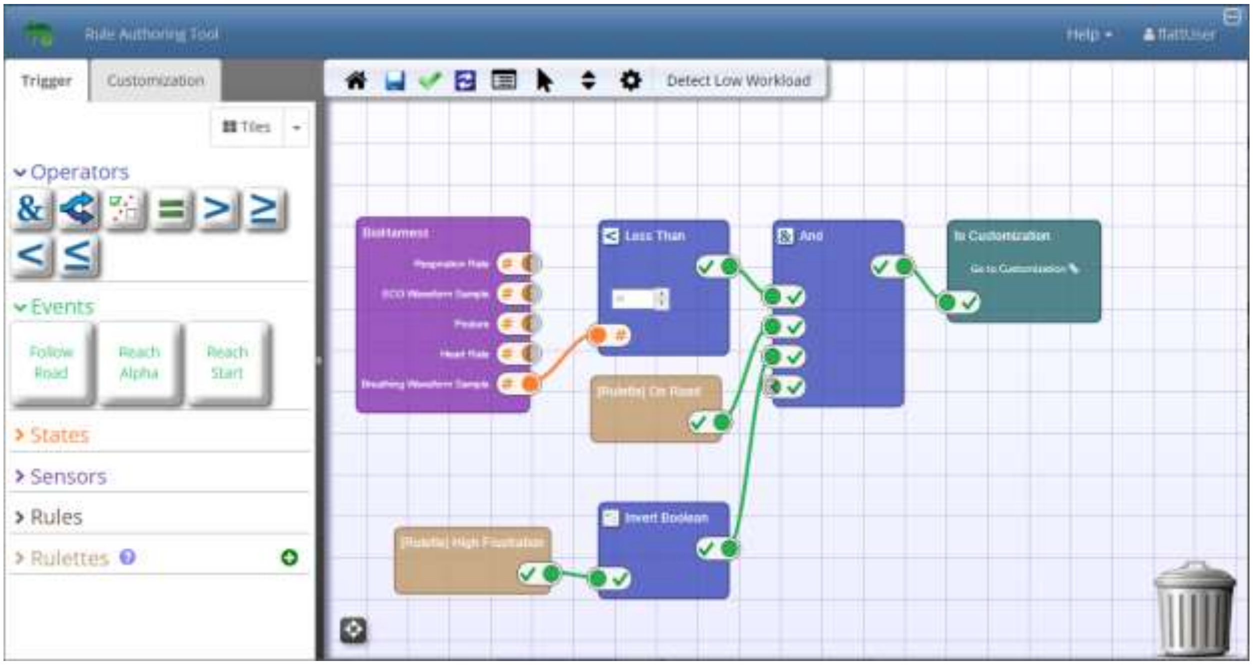


Figure 1. Chart Objects in Rule Trigger

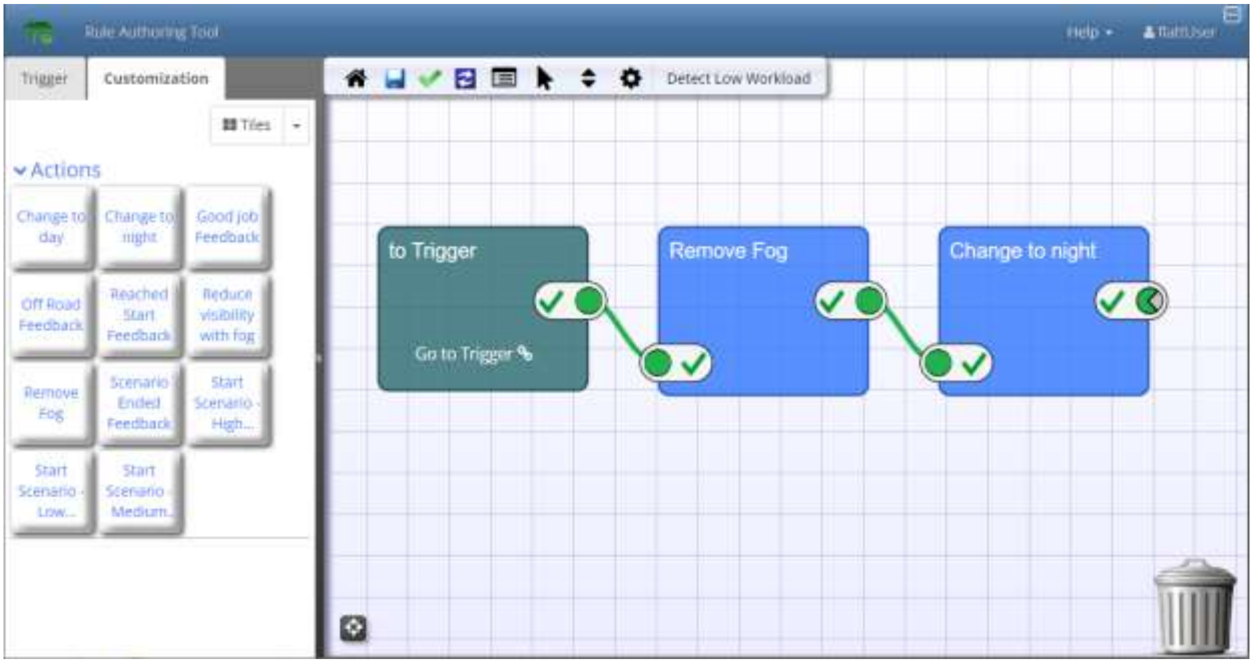


Figure 2. Chart Objects in Rule Customization

The FLATT authoring tool is much simpler to use and understand than GIFT's DKF. When designing a DKF, the user currently has to navigate through several tabs in the authoring tool to configure the scenario adaptation and state transition individually to create a *rule* – for example, if the user wants multiple adaptations to apply after one state transition, they would have to author each adaptation individually (one adaptation takes an average of 5-6 mouse clicks), then author the state transition in a different tab and select the adaptation set to apply (one state transition takes an average of 4-5 mouse clicks). In FLATT, the user has access to all VTE specific adaptations that are available in one simple drag and drop interface (Figure 2). Having all available data sources in one place results in a much easier interface to explain and understand than the complicated DKF structure. This can be essential particularly when collaborating with a team or another group of users.

To further assist the user, the FLATT authoring tool has a built-in validation feature that is automatically executed whenever the operator saves their current work. This validation will detect and notify the operator if any invalid states exist within the ruleset, such as disconnected chart objects, circular dependencies, or unauthored triggers and customizations. A dialog will appear and inform the operator what issues were found and provide guidance on how to resolve them in order to get into a working state.

SERVICE DEFINITION

The power behind FLATT is its ability to receive and evaluate a stream of information and quickly determine if the data meets the trigger conditions for any of the authored rules. This is done efficiently because it can utilize GIFT as a service to perform the complicated and intensive task of processing data sources, such as human-worn sensors, VTE states, and assessed learner characteristics and states. Since GIFT is already capable of handling this data, FLATT benefits by only receiving post-processed data to be evaluated. FLATT developed an approach, the Service Definition API, to define the types of data that will be made available by GIFT. This API is intelligent tutoring system (ITS) agnostic, so another ITS can define its own Service Definition to become FLATT compatible. Each Service Definition is specific to the ITS+VTE pairing for the scenario being created.

Table 1. Chart Object Service Definition Metadata

Field	Description
Name	A unique name for the data source (e.g., Reach Alpha, Bioharness)
Category	• Action – represents a TSC that can be applied to the VTE. • Event – an assessed event that occurred in the scenario. • Sensor – supported sensors that the trainee could use in the scenario. • State – individual characteristics (e.g., grit, motivation) or assessed trainee states
Expected Outputs	The data that can be received from this source (e.g., heart rate – number; follow road – text: Below/At/Above Expectation)
Description	An optional description used to aid the author by providing additional details about this data source

The Service Definition includes metadata, shown in Table 1, that gives context to a user interface for each data source. For GIFT, the data sources include the authored concepts and strategies, learner state attributes, and supported sensors. These items in the Service Definition instance are dynamically added as available chart objects that can be dragged into a FLATT rule – since authoring is done graphically by connecting objects on a chart into a *flow* (see Figure 1). Since chart objects are dynamically generated in FLATT based on the metadata provided, FLATT can easily stay updated with future GIFT features, such as newly supported sensors, VTE integrations, and DKF assessment conditions, with no need for additional labor or development in FLATT. For example, GIFT currently supports connecting with the Zephyr Bioharness (Zephyr, 2012); the BioHarness chart object seen in Figure 1 was produced by an entry in the Service Definition as follows:

- **Name**: BioHarness
- **Category**: Sensor
- **Expected Outputs**:
  - Respiration Rate (Number)
  - ECG Waveform Sample (Number)
  - Posture (Number)
  - Heart Rate (Number)
  - Breathing Waveform Sample (Number)
- **Description**: The Zephyr Bioharness is a wireless chest-based wearable device, capable of real-time recording of various physiological parameters.

When running a FLATT Ruleset during a live session, GIFT sends the associated chart object metadata alongside the processed data to FLATT. This allows FLATT to easily identify the data source and its associated chart object so that only the necessary data elements are delivered to the appropriate rules. FLATT will then perform evaluations on all authored rules that contain the chart object to determine if the rule's trigger conditions have been met and the customization should execute. A similar process is used to execute the customization's TSCs; the Service Definition metadata is sent from FLATT to GIFT for all TSCs that should be applied. GIFT associates the metadata with its known VTE adaptations and requests the VTE to apply them.

The FLATT user interface is accessible as a standalone application or as an embedded service through GIFT's DKF FLATT Condition. As seen in Figure 3, when authoring with FLATT as a standalone application, the Service Definition can be imported as a file by the FLATT User, whereas when accessing FLATT through GIFT, it is sent automatically via FLATT's API. This allows the user the option of authoring FLATT rulesets with or without an active connection to GIFT.

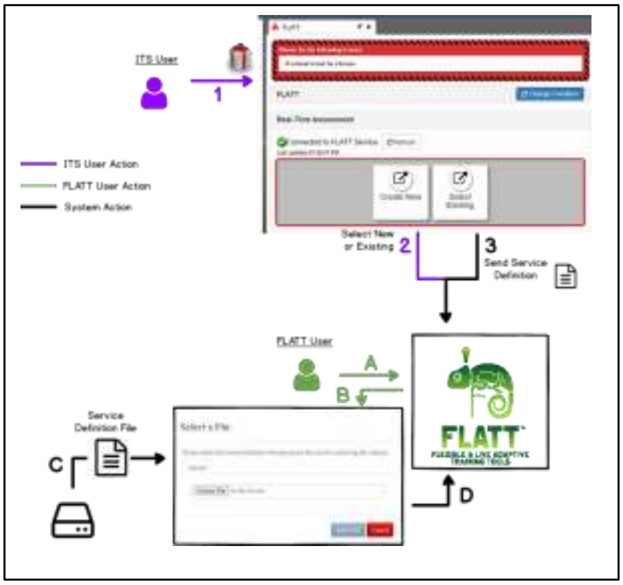


Figure 3. Service Definition by User

GIFT/FLATT LIVE SESSION OPERATION AND COMMUNICATION

Once the authoring is complete, FLATT is ready to be used during a live session to adapt the VTE in real-time. By utilizing GIFT as a service, FLATT delegates the complicated and intensive task of needing to receive, analyze, and process incoming data streams. GIFT performs the heavy-lifting and sends FLATT only the information that is relevant, such as assessed learner states, individual characteristic levels (e.g., high grit, low motivation), scenario data, and processed sensor data. The FLATT Condition was created within the GIFT DKF architecture in order to support this communication with the FLATT application.

GIFT DKF: FLATT Condition

The DKF is an XML file that contains elements, such as learner identification, waypoints, assessments, and scenario injects which allow the author to structure the interactions and instruction that a learner experiences while executing a scenario or mission in an external training application, such as Virtual Battlespace 3 (VBS3), (Burmester, 2021). This new condition class in the DKF provides an access point to connect to FLATT's API and allows an open line of communication between the two applications, permitting GIFT to pass information such as trainee, sensor, scenario, and assessment information to FLATT. FLATT receives this consolidated dataset and determines if any of the authored rules have been satisfied. Once a FLATT rule is triggered, FLATT can use the bidirectional connection to the FLATT Condition in order to send GIFT the TSCs to be applied; GIFT then redirects those TSCs to the attached VTE to perform the adaptations. The connection also allows FLATT to share its capabilities and features; using FLATT as a service, GIFT can gain access to new features such as rule and sensor data monitoring which are discussed in the next section.

FLATT as a Service for GIFT

FLATT's live-session services offer operators access to information not visualized by GIFT yet. FLATT can provide the collected live session data as a service to an external consumer, such as GIFT's Game Master (see Figure 4) and embed it into the consumer's own dashboard. The services are ITS agnostic so any application with access to FLATT's API can utilize these features.

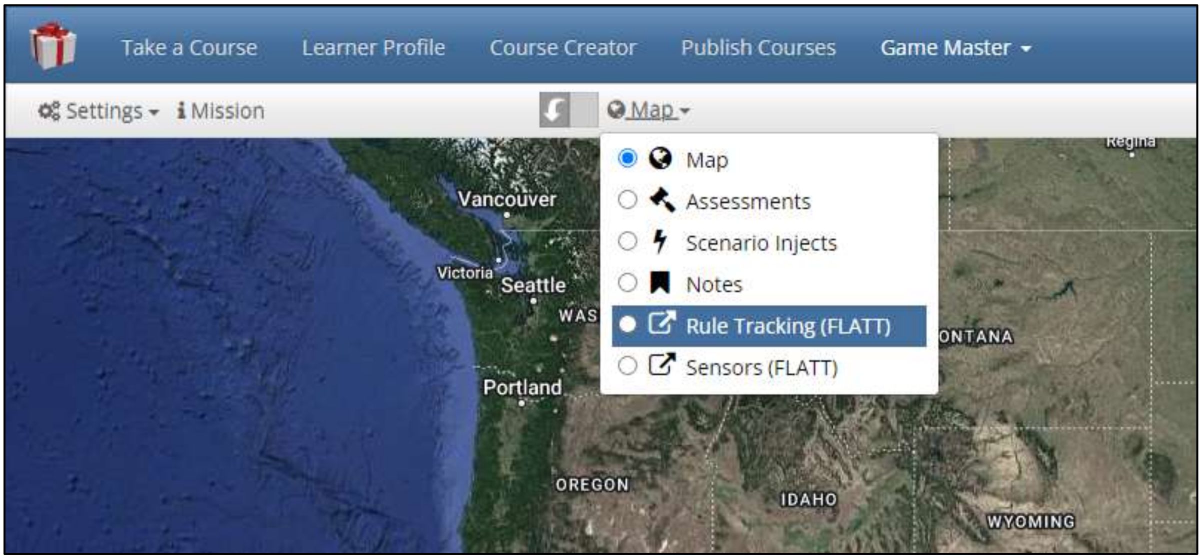


Figure 4. Game Master FLATT Services

Rule Monitoring Service

During a live session, the operator can use the Game Master tool to embed FLATT's live monitoring system for the authored FLATT rules (see Figure 5). This allows the operator to view and track the progress of each rule to see which items in the triggers, including the trainee and game state, have been met and which have not. Using Figure 5 as a reference, chart objects outlined in green, such as Less Than, are evaluating to true, meaning the flow is progressing through to the next chart object. Objects outlined in red, such as the On Road Rulette or And, are evaluating to false and are preventing the TSC from executing.

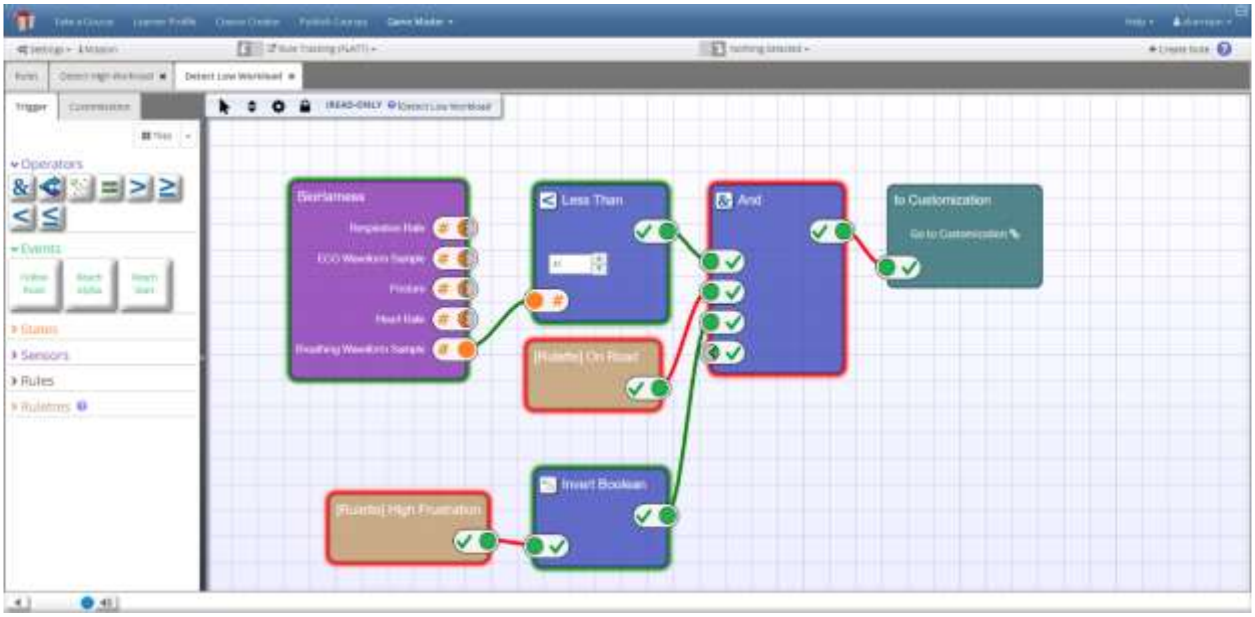


Figure 5. Live Session Rule Tracking

The rule tracking service also enables the operator to modify the existing rules on the fly, avoiding the need to account for all possibilities ahead of time. For example, if the trainee's breathing waveform level does not reach the value defined in the trigger (see Figure 5), the author could choose to alter the threshold value while the VTE training is running, and thus, allowing the customization to proceed. This sort of real-time calibration capability is invaluable for pilot testing for experiments or highly dynamic training programs. The complete authoring tool is available to modify the rule in real-time as needed. These modifications will take effect immediately without needing to pause or stop the current training session. The benefits of this are far reaching and include decreasing authoring and planning time, reducing trainee wait times, improving assessment and/or customization rehearsal capabilities, and improving operator control over the session to name a few. Future improvements to this service will allow the operator further control over the scenario by allowing new rules to be created during a live session instead of just modifying existing ones and having the option to disable/re-enable rules so that the operator can prevent rules from executing at any given point.

Sensor Data Monitoring Service

FLATT also provides a service for visually representing the trainees' connected sensor data (see Figure 6). The sensor data is displayed in real-time graphs as they are received. Since a single sensor may have multiple measurable data types (e.g., a bioharness can measure heart rate and breathing rate), each measurable data stream is contained within a separate graph for readability and ease of comprehension.

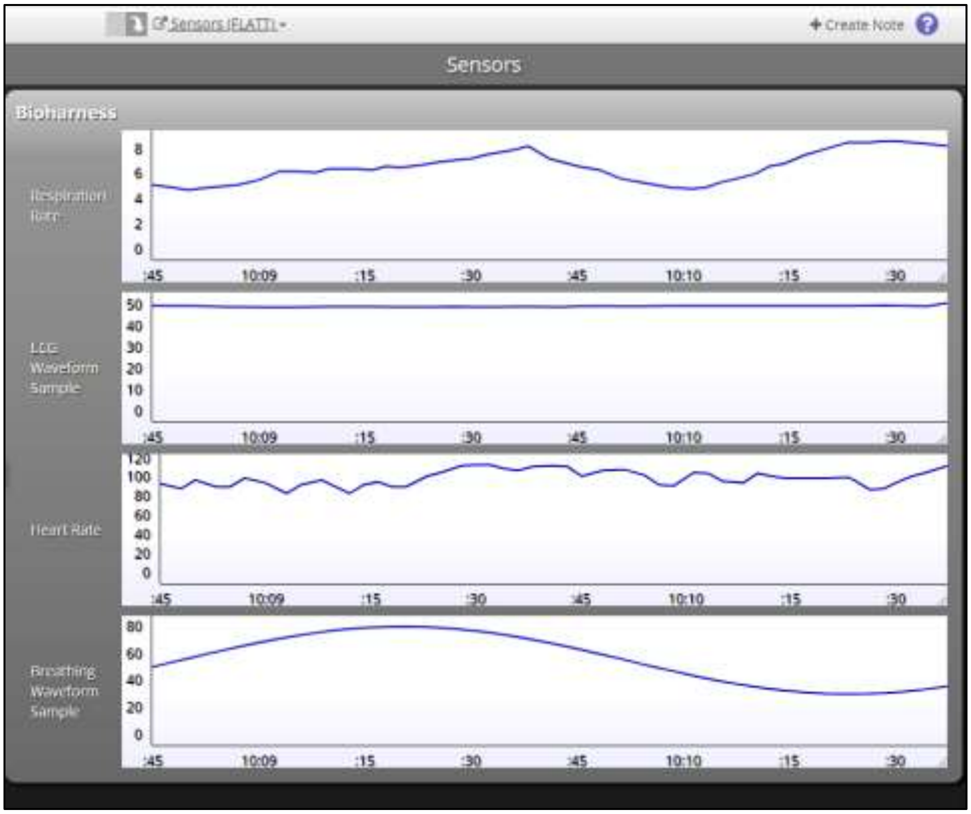
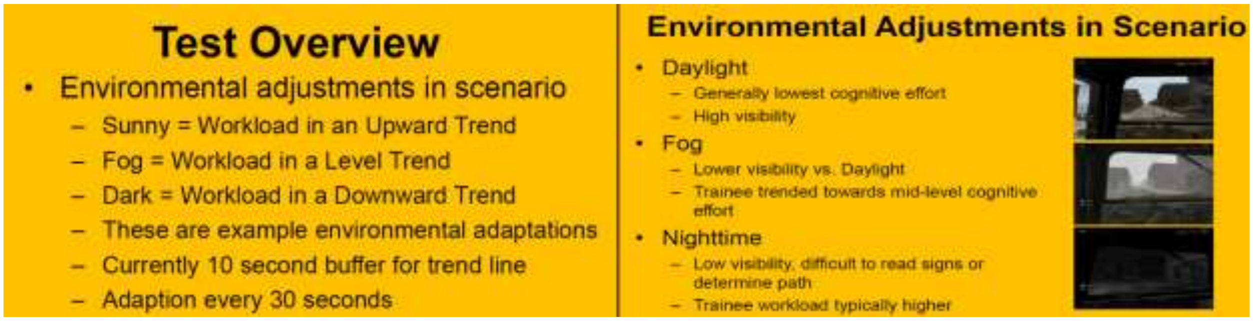


Figure 6. Live Session Sensor Data Visualization

By embedding FLATT's sensor data service, GIFT can expand its Game Master capabilities beyond what is currently available. Being able to monitor the trainee's real-time physical state provides the operator with a hitherto unseen understanding of the trainee's condition during the scenario. This information can be used by the operator to better adapt the rules and/or scenario to the trainee's specific circumstances. Future improvements upon this service will allow the operator to hide undesirable data streams, reorder the visual graphs for ease-of-use, and display data classification thresholds within the graph (e.g., heart rate is slow; heart rate is elevated) to make readability easier than using the raw data.

USE CASE EXEMPLAR

One of FLATT's objectives is to increase training effectiveness; a way of doing this is to utilize the trainee's state to personalize the adaptive training capabilities within the training scenario. It was decided to leverage existing published literature to demonstrate that objective. The scenario found in *Physiological Based Adaptive Training* (Schnell et al., 2017) demonstrates the effectiveness of using workload, performance state (how the trainee is doing in the scenario), and subject state (physical, emotional, or mental) to change the difficulty of the scenario by performing real-time adaptations. This type of exercise is self-paced based on the trainee's needs, which can lead to a much greater effectiveness in the trainee's learning while also reducing the need for human instructors, which are a limited resource. FLATT was able to reproduce the exemplar scenario using GIFT and VBS to build out the actual scenario and create FLATT rules (see Figure 5) to identify when adaptations should occur and which real-time TSCs should be used to adapt VBS. The rules created imitate the test overview found in *Physiological Based Adaptive Training*, which can be seen in Figure 7 – when the trainee has a high workload (upward trend) the scenario environment changes to day to reduce the scenario difficulty to help reduce the workload; if the workload is medium (trend is level) the environment changes to foggy to maintain a mid-difficulty level; and lastly, if the workload is low (downward trend) the environment changes to nighttime to increase the difficulty. FLATT determines workload by the trainee's breathing rate, performance during the scenario, and trainee's frustration level.



Test Overview

- Environmental adjustments in scenario
  - Sunny = Workload in an Upward Trend
  - Fog = Workload in a Level Trend
  - Dark = Workload in a Downward Trend
  - These are example environmental adaptations
  - Currently 10 second buffer for trend line
  - Adaption every 30 seconds

Environmental Adjustments in Scenario

- Daylight
  - Generally lowest cognitive effort
  - High visibility
- Fog
  - Lower visibility vs. Daylight
  - Trainee trended towards mid-level cognitive effort
- Nighttime
  - Low visibility, difficult to read signs or determine path
  - Trainee workload typically higher

Figure 7. Physiological Based Adaptive Training - Test Overview and Environmental Adjustments

Breathing Rate

The trainee's breathing rate is measured using a Zephyr Bioharness. According to the paper *Respiratory Changes in Response to Cognitive Load: A Systematic Review* (Grassmann et al., 2016), a person's breathing rate is a highly effective indicator of their cognitive load state. Since GIFT already supports receiving data streams from a Zephyr Bioharness, it was an easy determination to use this device. The authored FLATT rules in this exemplar used the average breathing rate from collected Bioharness data to determine what the threshold levels would be for low, medium, or elevated breathing. These levels could easily be adjusted during a live session to accommodate a trainee with unusually low or high limits.

Performance

In-game performance is measured as the trainee progresses through the VBS scenario. The scenario itself is fairly simple; the trainee is instructed to drive a vehicle from one waypoint to another while staying on the road. Their performance is calculated as "good" or "bad" if the vehicle is on or off the road respectively. The FLATT rules can trigger the TSCs described above to modulate the difficulty of this otherwise simple scenario.

Frustration

Lastly, the trainee's frustration level is assessed prior to the scenario using the GIFT survey course object available in the course authoring tool. The Self-Assessment Manikin survey (Bradley & Lang, 1994) is presented to the learner to assess their current frustration level. When the scenario starts, GIFT sends the assessed learner state to FLATT to be used in its rule evaluations.

Future use cases

This use case highlights many of the exciting and powerful features of FLATT. Ongoing basic research is developing new ways to use unintrusive measurements to opportunistically sense Soldier states (e.g., Neubauer et al., 2020). Other work is discovering relationships between trainee characteristics and how feedback framing (Files et al., 2019) and different levels of immersion (Pollard et al., 2020) affect learning in lab-designed tasks. FLATT's capabilities will enable researchers to observe how these states, characteristics, and TSCs relate to task performance and learning during naturalistic, simulation-based training. Moreover, FLATT makes it easy to experimentally vary elements of the simulation to test hypotheses about relationships between ongoing events and states. Because FLATT is aimed at both researchers and trainers, promising applications from research can easily be adopted and evaluated by trainers.

FUTURE IMPROVEMENTS

One of the main benefits of FLATT is its ability to add new services and features that can easily be utilized by GIFT with minimal development. Since the data provided by FLATT is consumed by GIFT as a service, FLATT can produce new data feeds as development continues, and GIFT can host and display them as it sees fit. For example, a future FLATT service could be to allow the operator to view live video streams such as webcams or wearable cameras. These data streams could be received by FLATT and provided to GIFT as a live-stream service. This will give the operator a visual way to track the trainee during an exercise and personally monitor their progress through the scenario. An example of a new feature is to give the operator the ability to create new rules during a live session. This is a natural progression from the current feature of being able to modify existing rules. The newly created rules will be able to take effect during the live session without interruption, and FLATT will be able to immediately begin applying the collected game state, trainee state, and sensor data to the new rule. This feature would require no additional effort from GIFT as it already supports embedding the rule monitoring service into its Game Master page.

FLATT is still relatively young in its development and there are plans to add a lot of great new features.

- Incorporating user management and permissions. This will allow users to create rulesets and share them with other users on the system, which should increase productivity as collaboration becomes more involved.
- Multi-user collaboration such as concurrent editing, mirror/instructor mode, change logs and user-history, etc.
- Integrating FLATT support into more ITS/Learning Management System (LMS) applications
- Automated/Guided installer.
- Ability to integrate with sensor data stream interpreters.
- Advanced live-session operator features such as manually overwriting states for authored rule chart objects.
- After action review (AAR) metrics and services that could be used for research analysis such as when and how many times rules were triggered or TSCs were applied.

CONCLUSION

FLATT has come a long way since it started development a year ago, and it will continue to evolve over the next year. While experienced trainers have deep expertise and knowledge in the kinds of adaptations and interventions that work, those adaptions might not be easily executed during a live session or cannot be adapted on the fly. With FLATT, trainers can easily create rules to perform the adaptations while also providing the trainer flexibility in managing those adaptations during a live session. Researchers can now be provided with the data to develop new models of learners and determine effective training customizations that are appropriate for each learner state. FLATT’s features allow for rapid prototyping which enables the user to avoid wasting precious time and resources building a stovepiped ITS solution that may become obsolete after a few data collections. These resources can instead be focused on researching new scenario adaptation ideas, determining which adaptations are most effective, and improving the overall training experience for the learner.

REFERENCES

ARL GIFT. (2022). *GIFT Tutoring*. Retrieved from https://gifttutoring.org/projects/gift/wiki/Overview

Bradley, M., & Lang, P. (1994). Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry*, 25(1), 49-59.

Burmester, E. (2021). Authoring Collective Training Demonstrations in GIFT, 2021 Update. *Proceedings of the Ninth Annual GIFT Users Symposium*, (pp. 44-51). Retrieved from https://www.gifttutoring.org/attachments/download/4104/giftsym9_proceedings_FINAL_1.0.pdf

Files, B. T., Pollard, K. A., Oiknine, A. H., Passaro, A. D., & Khooshabeh, P. (2019). Prevention focus relates to performance on a loss-framed inhibitory control task. *Frontier in psychology*, 10, 726.

Grassmann, M., Vlemincx, E., von Leupoldt, A., Mittelstadt, J. M., & Van den Bergh, O. (2016). Respiratory Changes in Response to Cognitive Load: A Systematic Review. *Neural Plasticity*. Retrieved from https://www.hindawi.com/journals/np/2016/8146809/

Harrison, S., & Burmester, E. (2021). FLATT: A new real-time assessment engine powered by GIFT. *Proceedings of the Ninth Annual GIFT Users Symposium*, (pp. 129-138). Retrieved from https://www.gifttutoring.org/attachments/download/4104/giftsym9_proceedings_FINAL_1.0.pdf

Neubauer, C., Schaefer, K. E., Oiknine, A. H., Thurman, S., Files, B., Gordon, S., . . . Gremillion, G. (2020). Multimodal Physiological and Behavioral Measures to Estimate Human States and Decisions for Improved Human Autonomy Teaming. *CCDC Army Research Laboratory Aberdeen Proving Ground United States*.

Pollard, K. A., Oiknine, A. H., Files, B. T., Sinatra, A. M., Patton, D., Ericson, M., . . . Khooshabeh, P. (2020). Level of immersion affects spatial learning in virtual environments: results of a three-condition within-subjects study with long intersession intervals. *Virtual Reality*, 24, 783-796.

Schnell, T., Reuter, C. J., Gunnink, E. D., Parker, B. M., Richey, C. H., Hoke, J. A., & Moss, J. D. (2017). Physiological Based Adaptive Training. *Proceedings of the Fifth Annual GIFT Users Symposium.* Retrieved from https://www.gifttutoring.org/attachments/download/2136/10_Learner%20Modeling%20Paper_Schnell%20et%20al.pdf

Zephyr. (2012). Retrieved from https://www.zephyranywhere.com/media/download/bioharness3-user-manual.pdf

ABOUT THE AUTHORS

***Steven Harrison** is a software engineer at Dignitas Technologies and the technical lead for the FLATT project. He has over 10 years of professional software research and development experience. Steven has been a member of the GIFT development team since 2017 and was also the technical lead for the Squad Advanced Marksmanship Trainer (SAMT) effort that integrates GIFT intelligent tutoring with marksmanship training simulators, used on systems such as the Engagement Skills Trainer (EST). The Marksmanship effort supports research and human experimentation on providing ITS capabilities for training marksmanship, including research for capturing various sensor and device data streams. Steven has been responsible for ensuring the development of FLATT and SAMT.*

***Mitchell Gorsd** is a junior software engineer at Dignitas Technologies working on the FLATT project. He graduated from the University of Central Florida and has been working on the FLATT project for over a year. Mitchell has experience with simulation software, including Unity and Augmented Reality, along with creating user experiences in React, React Native, and JavaScript.*

***Michael Hoffman** is a senior software engineer at Dignitas Technologies, the Project Manager for the FLATT contract with ARL, and the technical lead for the GIFT contract with DEVCOM. Through more than a decade he has been responsible for leading the development of GIFT, collaborating with the ITS community, and supporting ITS related research. 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 Soldier Center and their contractors. His background has helped guide the direction of FLATT development.*

***Dr. Kimberly Pollard** is a biologist at DEVCOM ARL Humans in Complex Systems division, where she specializes in research on training and human-technology interaction. Her current work examines the ways in which information presentation and individual differences come together to affect performance in human-technology domains. Topics include learning in virtual reality, human-agent teaming, and human adaptation to changing technologies. She earned her Ph.D. from UCLA and her B.A. from Rice University.*

***Dr. Benjamin Files** is a biologist at DEVCOM ARL Humans in Complex Systems division, where he uses methods from cognitive and perceptual neuroscience and experimental psychology to better understand how individual characteristics relate to the effectiveness of training interventions and information interfaces. Topics include reasoning with, interpreting, and expressing uncertainty; mutual human-autonomy adaptation; and transferability of practice effects. He earned his Ph.D. from the University of Southern California and his B.A. from the University of California, Berkeley.*

Authoring Collective Training Demonstrations in GIFT, 2022 Update

Elyse Burmester
Dignitas Technologies

INTRODUCTION

The Generalized Intelligent Framework for Tutoring (GIFT) has served a number of unique purposes since its release in May of 2012, but the most recent shift of interest in GIFT's usage has been focused on collective training in virtual training environments. As a frequent developer of collective demonstrations using the training applications integrated with GIFT, I have developed the streamlined process provided in this paper to create training content quickly and effectively. While GIFT has several assessment tools in its suite, this paper will focus only on the Real-Time Assessment tool, which is powered by the Domain Knowledge File (DKF). As a follow up to previous GIFT Symposium authoring guides for team-based training exercises (Burmester, 2020; Burmester, 2021), this version expounds on the fundamental components of the DKF authoring process and provides example use-cases for each section to give further context and inspiration for techniques that readers' can employ on their own.

DKF OVERVIEW

The DKF is an Extensible Markup Language (XML) file that provides all the necessary information in order to execute a lesson with an external training application (Domain Knowledge File GIFT Wiki, 2021). DKFs are read by the Domain module and used by the Domain and Pedagogical modules. This schema contains elements such as scenario name, learner identification, waypoints, assessments, etc. You can edit a DKF using a text editor (although this is not recommended for non-development GIFT users) or the more popular DKF Authoring Tool.

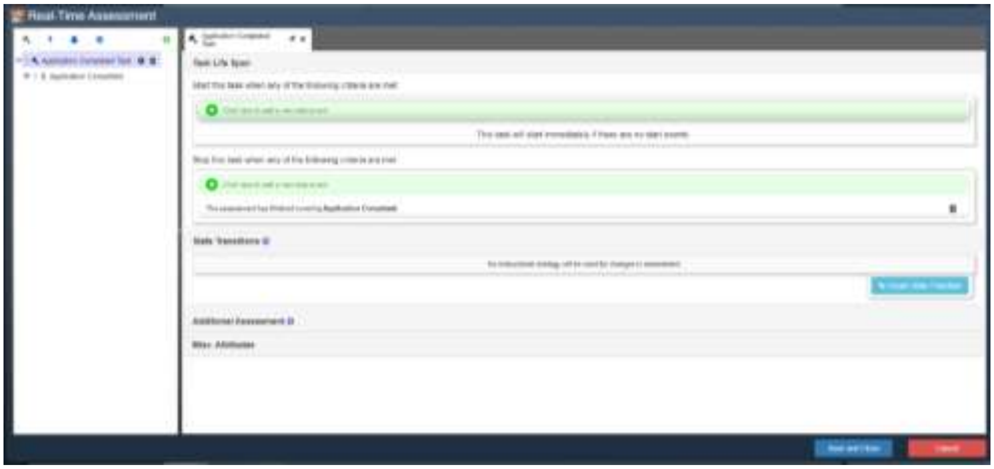


Figure 1. Task Authoring Panel.

When you open the DKF editor, also called Real-Time Assessment, your screen will 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.

Tasks and Concepts

Tasks and concepts outline the assessments, measures of performance, and/or task steps that are required in a training scenario. This list can be defined from any number of sources; for example, an author might gather measures of performance from an existing Army Battle Drill and task steps from a Training and Evaluation Outline (T&EO). On the other hand, tasks and concepts can also be created entirely from scratch using any well-defined metrics or guidelines the author wishes to utilize.

Tasks have a lifecycle that must be defined by Start and End triggers. These triggers are used to structure the flow of assessments throughout the training scenario - the concepts listed under each parent task will only remain active for the duration of the parent tasks' lifecycle. Start and End triggers can be authored using one or any combination of the following: 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, 4) when a Learner Action is selected (explained in future sections), or 5) when a strategy is applied (also explained below).

Creating this Task lifecycle structure is important when considering any dependencies that certain parts of the training may require. For example, before an Infantry team can properly Enter and Clear a Building (Department of the Army, 2016) they must perform a specific set of task steps when moving to their objective (the building). The movement behaviors required when approaching the objective versus entering/clearing the objective greatly vary and would not be assessed at the same time or in the same manner given the different contextual dependencies, i.e. environment and spatial conditions outside of the building vs. inside the building/its rooms. Therefore, the task steps, or Concepts, for approaching the objective should be defined within their own Task, separate from the Task that contains the Concepts for entering and clearing the objective.

Condition Classes

GIFT has a number of automated assessment options, or Condition Classes, that provide specific scoring rubrics for pre-defined tasks/measures and output performance assessments based on information consumed from game state messages. Condition Classes are the backbone of the task and concept list - housed in their own individual java classes, they contain logic to assess the learner's actions in the domain against pre-defined (but malleable), research-based measurements. There are currently twenty-nine Condition Classes implemented in GIFT at the time of this paper's publication, including a unique (but not technically automated) assessment type which will be explained in the next section. Some examples of automated Condition Classes include: 'Entered Area', which checks whether a learner has entered a location among one or more defined locations; 'Health', which checks whether team members in the team organization are not healthy or have some type of damage state; and 'Spacing', which checks whether two objects/players controlled by learners are maintaining proper spacing or formation.

In addition to its automated Condition Classes, GIFT provides a manual assessment option that requires human input to assess learner performance - a condition class called Observed Assessment. Observed Assessments can be used in place of measures that are required as part of the training but cannot be automatically assessed by any of the existing Condition Classes, which essentially cover any behaviors or actions that GIFT cannot pick up through its currently supported messaging logic (i.e. DIS, Protobuf, etc.). An example use case of an Observed Assessment might be an assessment of communication within a team during a training exercise; GIFT does not currently support Natural Language Processing (NLP) and therefore would not be able to assess any spoken communication from the learners during the training exercise. The Observed Assessment condition class, however, would allow some outside observer of the training exercise to manually input an assessment based on their perceived judgement of the team's ability to communicate.

For more information on Condition Classes, including how to build your own, check out the "Domain Knowledge File" documentation on www.gifttutoring.org (tip: use the search bar to quickly find the latest version of this documentation; Wiki page search results are delivered in chronological order based on publication date).

Strategies

Strategies, also called Scenario Injects, are another key component of the DKF. Strategies can be used to adapt the virtual training environment during a scenario and/or apply remediation tactics to provide further instructional support for the learner based on their performance or current physiological state. Strategies can also be used as triggers to start or end a Task, providing additional flexibility for managing assessments during training. GIFT currently supports five strategy types: Feedback, Present Media, Modify Scenario, Present Survey, and Start Conversation. The following sections will explain each strategy type in detail.

Feedback

The Feedback strategy type is used to deliver a message to the learner. There are four types of feedback currently available to choose from: Present a Message, Local Webpage, Play Audio, and Play Avatar Script.

Present a Message allows you to send a message via text and/or dynamic speech to all learners by default, or just to one/many learners if specified, with optional presentation features such as displaying the message in the training application (if supported), playing a beep sound through the browser when the feedback message is displayed, and flashing a yellow background behind the text when the feedback is displayed.

Local Webpage and Play Audio allow you to upload a file of the respective type to display to the learner during training. Play Avatar Script allows you to upload an Agent File and Speech Key, plus the option to present a message via text along with the script.

This strategy type can serve many purposes for both the instructor and the learner; for example, an instructor might use Present a Message to deliver instructions before a scenario begins (i.e. If the learner must perform a set of operations to set up the training application) rather than doing so themselves, which in turn saves them time and allows them to address any individuals that might have specific issues during the process. As another example, a team conducting a distributed training mission together might benefit from hearing feedback delivered via audio file rather than having to read text on their screen which might distract them from the supporting tasks they are executing with other teammates.

Present Media

The Present Media strategy type is used to deliver a media resource to the learner. GIFT currently supports six media types: Slide Show ("PowerPoint 97-2003 Show (*.pps)" only), PDF, Local Webpage, Local Image, Web Address, and YouTube Video. Each media type can be uploaded directly into the authoring tool interface and the author can provide a unique title that is displayed as a header on the webpage while the learner is viewing the media content.

This strategy type is useful for presenting specific examples or instruction using established lesson materials at key points during a training scenario, providing individualized learning experiences in real time. An instructor might find this useful when curating the flow of their learners' experience; for example, if they wanted to deliver YouTube videos that give context for an upcoming portion of the training before the actual Tasks, or assessments, begin.

Modify Scenario

The Modify Scenario strategy type is used to adapt the scenario or virtual training environment the learner is in. There are fourteen types of adaptations currently supported: Create Actors, Remove Actors, Teleport, Highlight Object, Remove Highlight, Breadcrumbs, Remove Breadcrumbs, Fog, Overcast, Rain, Time of Day, Endurance, Fatigue Recovery, and Script.

Create Actors and Remove Actors is used to add or remove an actor/vehicle/object in the training environment respectively. For example, an instructor might use this strategy type to increase the difficulty of a scenario by adding additional enemies for learners that are showing signs of boredom, or to decrease the difficulty for a learner that is not performing well by removing enemies or physical barriers in the environment.

The Teleport strategy type is used to physically transport a learner to a location in the virtual environment, which is specified by the author in the authoring tool interface.

Highlight Objects and Remove Highlight will add or remove a visual indicator (a red, green, or blue floating arrow) above a specified team member or location in the virtual environment. This strategy type might be used, for example, to identify important people/locations or provide visual cues to draw a learner's attention to an event in the simulated environment.

Fog, Overcast, Rain, and Time of Day are environmental adaptations that change the weather conditions within the simulation. Time of Day provides the author with four options: dawn, midday, dusk and midnight; while Fog, Overcast, and Rain offer a selection between 0 and 1 for Density and a time value for Duration.

The Endurance strategy type is used to set the endurance value for learners in supported training applications. The Fatigue Recovery strategy type is used to increase or decrease a learner's fatigue recovery rate during the scenario.

Finally, the Script strategy type is used to apply any scripting logic the author chooses to provide, allowing the author to utilize scenario modifications and actions offered by the training application that are not included in the pre-canned strategy types listed above.

Present Survey

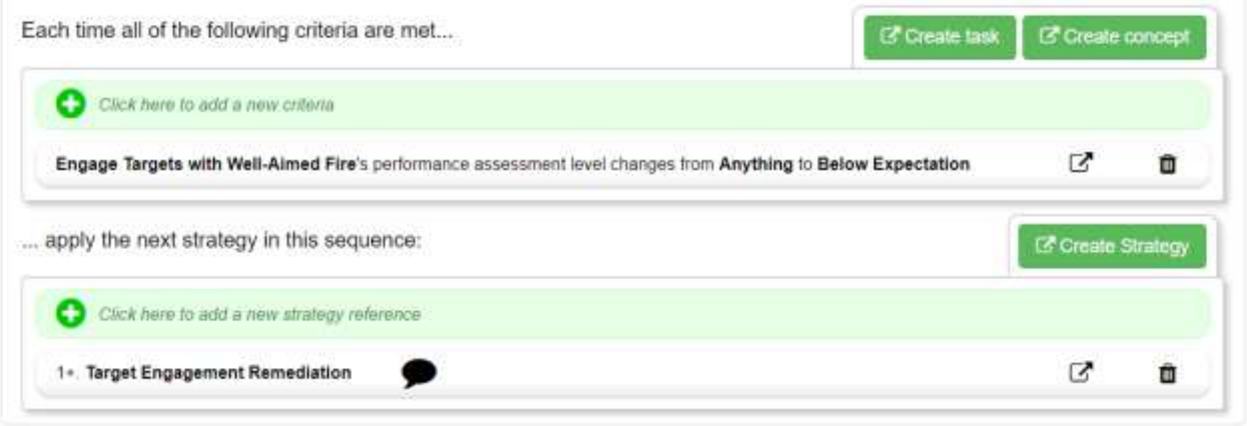
The Present Survey strategy type uses the results of a survey (which can be authored on the fly from the DKF authoring interface or the author can choose an existing survey within the course) to update the assessment of the task or concept that is identified. This strategy type is unique because it is the only type of strategy that directly influences the assessment of a task or concept. This could be useful to an instructor for many reasons; for example, one might use this strategy type to present a survey that checks the learners' knowledge of the skills they are performing in a virtual training environment to make sure they understand the fundamentals and are not just using muscle memory. Based on those survey results, that instructor could further modify the scenario for low scoring individuals or provide remediation for those that answered incorrectly on certain survey questions. This strategy type opens up even more flexibility in the survey authoring tool itself; and while this paper does not go into detail on that authoring tool, Anne Sinatra's "The 2021 Authoring Guide for GIFT", published in last years' GIFTSYM9 Proceedings, provides a thorough walkthrough of that process (Sinatra, 2021).

Start Conversation

The Start Conversation strategy type is used to start an AutoTutor or Conversation Tree conversation with the learner. AutoTutor is a computer-based tutor that helps students learn by holding a conversation in natural language. The AutoTutor Script Authoring Tool (ASAT) is a third-party based service which is not integrated in the GIFT authoring tool directly (Hoffman & Ragusa, 2015) but superficial collaboration during the authoring process in GIFT is being explored (Wang et al., 2020). Conversation Tree conversations, on the other hand, can be authored directly in GIFT from both the course authoring and DKF authoring level or you can choose an existing conversation that was previously authored. Conversation Tree conversations can be useful for increasing engagement in the learner by delivering instruction or eliciting information in an interactive format.

State Transitions

State Transitions are an important component of the DKF authoring tool because they provide the foremost mechanism to deliver a Strategy to its intended audience or application. Outside of State Transitions, the only other way a strategy can be applied *automatically* is through a Task start or end trigger, as discussed in previous sections. State Transitions apply their referenced Strategies based on some evaluation of criteria defined in the DKF authoring tool interface. Currently there are three evaluation criteria options to choose from: an authored Task, an authored Concept, or a possible Learner State. Once the criteria option is chosen, the author can set the state change that should trigger the application of strategies for that State Transition. An example of this operation in its final form would be: "If the learner's performance in the concept named 'Engage Targets with Well-Aimed Fire' changes from Anything to Below Expectation, apply the strategy titled 'Target Engagement Remediation'" (shown in Figure 2).



Each time all of the following criteria are met...

Engage Targets with Well-Aimed Fire's performance assessment level changes from **Anything** to **Below Expectation**

... apply the next strategy in this sequence:

1. **Target Engagement Remediation**

Create task
Create concept
Click here to add a new criteria
Create Strategy
Click here to add a new strategy reference

Figure 2. Example State Transition

State Transitions can be as complex or as simple as needed; it is in the hands of the author to shape the experiences and paths they want for their learners' during execution. Creating a blueprint of any expected implications or reactions for the assessments in your DKF ahead of time greatly reduces the workload of this process. This is often times the most daunting aspect of DKF authoring for new users, but in my experience the learning curve does not take long to overcome.

Assessment Properties

The final elements of the DKF 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 but are not usually required.

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.

Places of Interest

This section defines the global list of points, paths, or areas associated with locations in a virtual environment, e.g. Virtual Battlespace 3 (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.

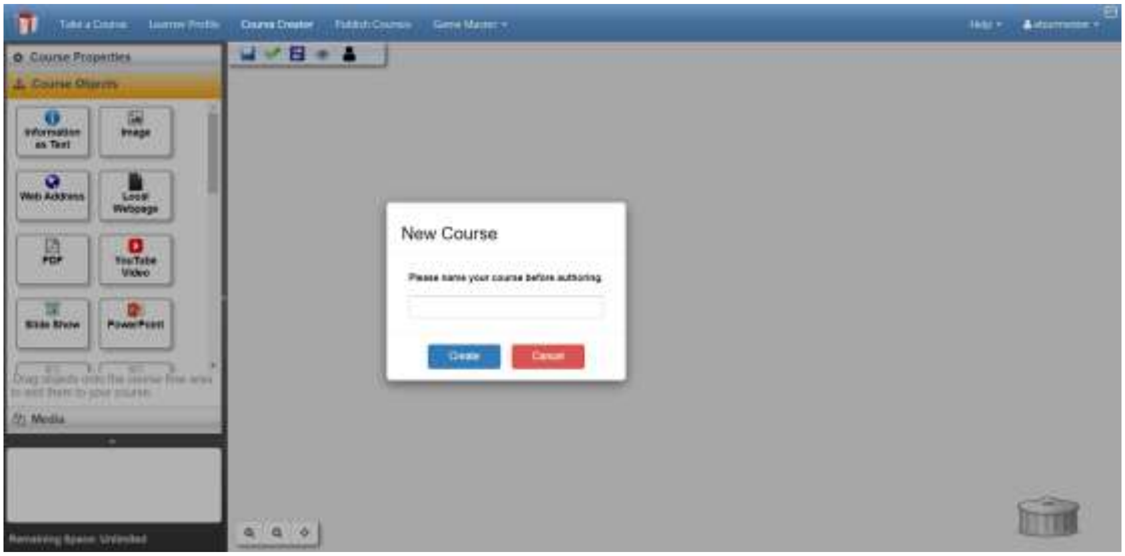
AUTHORING A DKF

Now that you are familiar with the main components of a DKF, you can follow the steps below to create a DKF on your own. This guide 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. It is intended to be simple and easy to use - it will not discuss advanced features of the DKF, such as course concepts. Readers are encouraged to explore the documentation provided on gifttutoring.org for more in-depth, technical information.

In order to access the full capabilities in this guide, you will need to run GIFT locally using the desktop version. The DKF Authoring tool can be accessed in GIFT Cloud but at the time of this papers publication training applications can only be used in the desktop version of GIFT.

1. Start by launching GIFT and logging in with your GIFT account. Once the Take a Course page loads, click on the “Course Creator” tab at the top of the webpage. Enter a useful course name and save it.

2. Scroll through the list of Course Objects on the left side of the webpage (shown in Figure 3) and find the appropriate training application for your scenario. For the purpose of this guide, we will use the Virtual Battlespace course object.



Please name your course before authoring

Figure 3. Create a Course page

3. Add all of the known entities within your scenario to the Team Organization list found under the "Assessment Properties" tab (shown in Figure 4). Consider also adding objects or entities that may not be played by a human but might be utilized as a trigger for some event in your scenario.

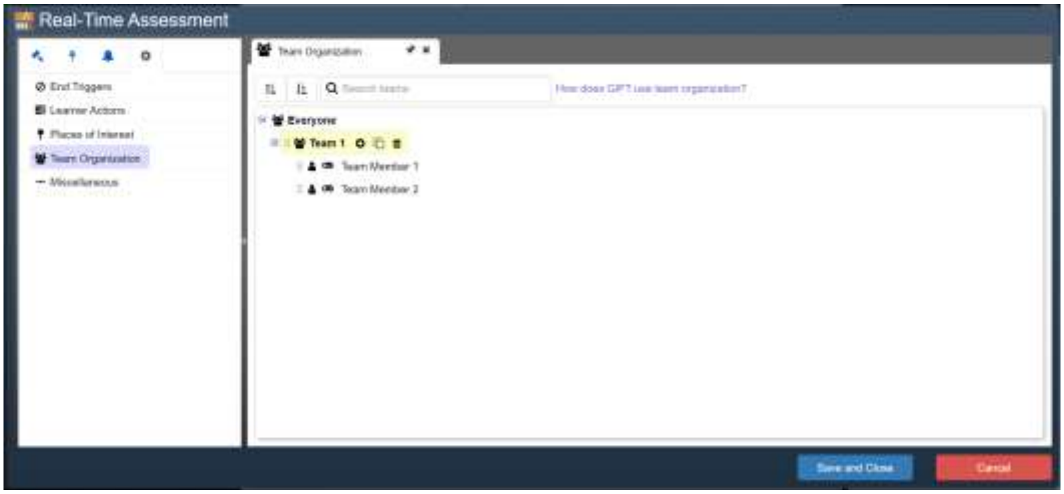
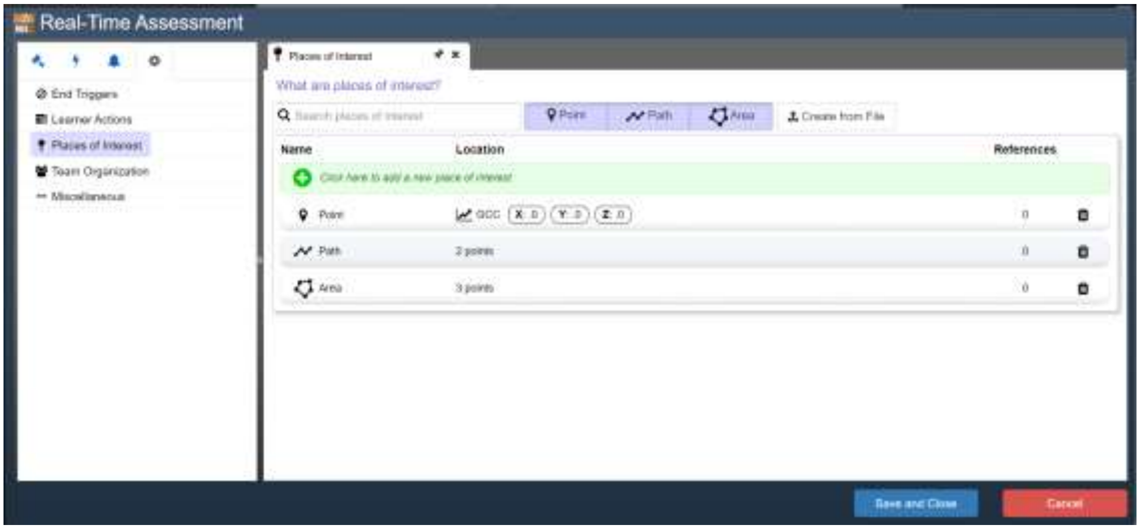


Figure 4. Team Organization Panel

4. After all of your entities are accounted for, open the Places of Interest tab in the Assessment Properties menu (shown in Figure 5). Use the green plus sign to add all relevant locations within your scenario - locations might be used to trigger events or provide context for an assessment.



Name	Location	References
Point	GCC X: 0 Y: 0 Z: 0	0
Path	2 points	0
Area	3 points	0

Figure 5. Places of Interest Panel

5. Now you are ready to populate the Tasks and Concepts. Open the Task panel by clicking on the hammer icon in the top left corner of the screen. Add an item to this task list using the green add button located next to the tabs at the top of the panel (see Figure 6). 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.

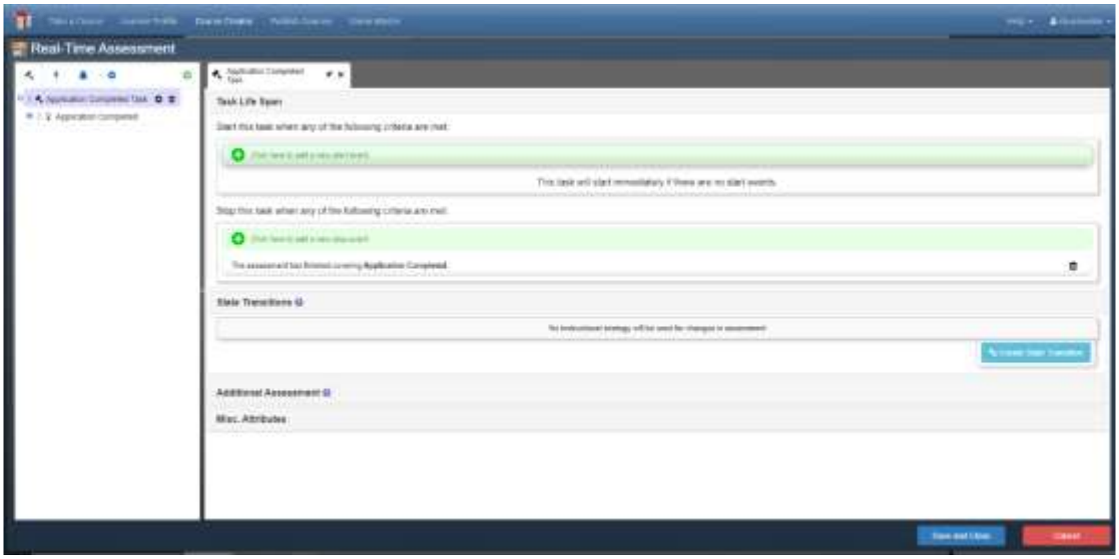
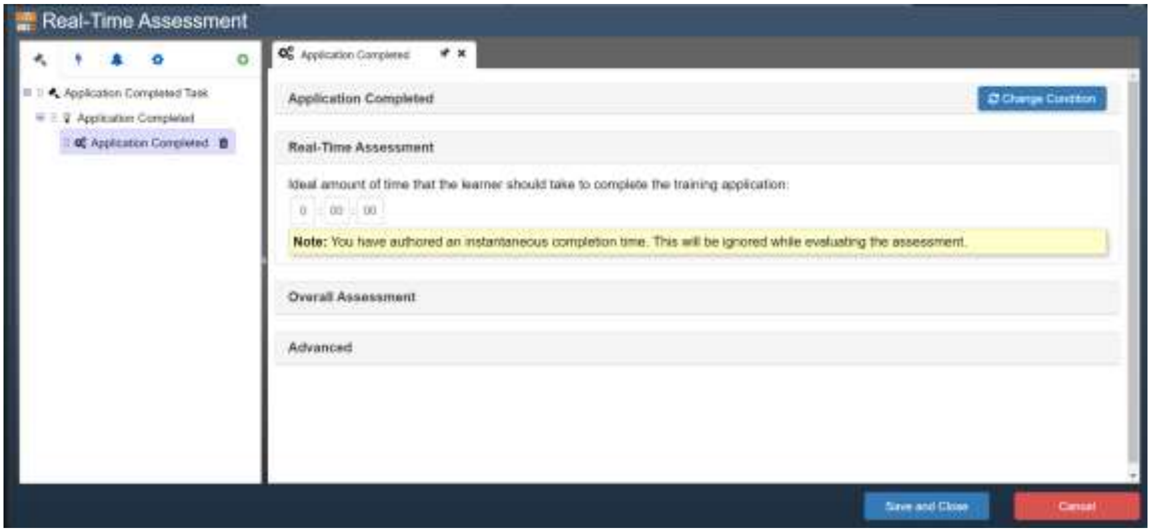


Figure 6. Task Authoring Panel.

6. 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. The "Overall Assessment" and "Advanced" sections within this panel contain additional options to customize assessments, but these options are not required for validation. An example of a condition class can be seen in Figure 7. Repeat steps 3 and 4 for all assessments required in your scenario.



Ideal amount of time that the learner should take to complete the training application:

0 : 00 : 00

Note: You have authored an instantaneous completion time. This will be ignored while evaluating the assessment.

Figure 7. Application Completed Condition Class

7. After adding all of your tasks and concepts, click on the Strategies tab (shown in Figure 8). 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 select an activity for the strategy to implement.

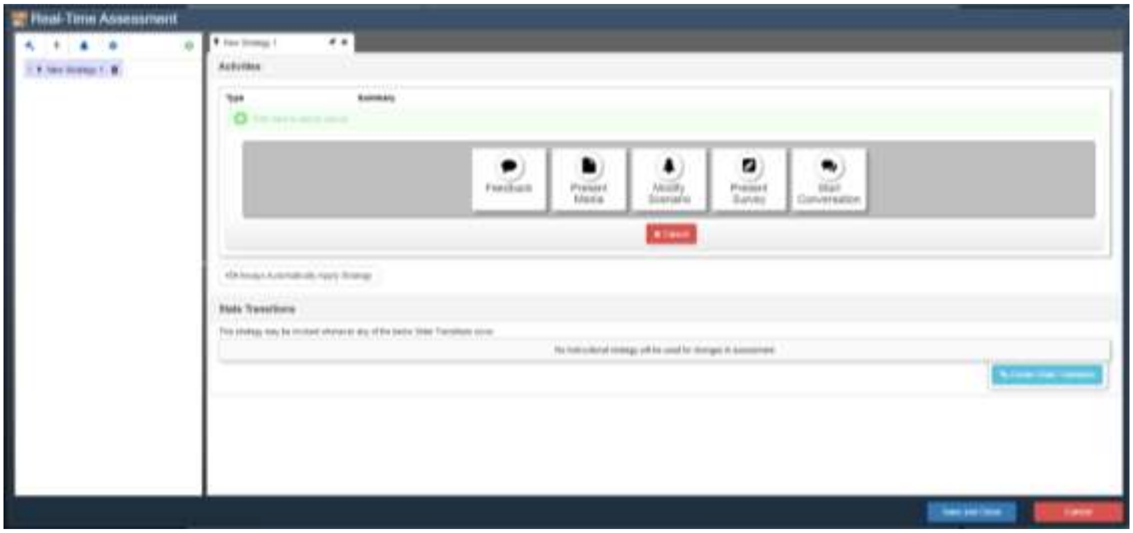


Figure 8. Strategy Authoring Panel

8. 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 9). (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.)

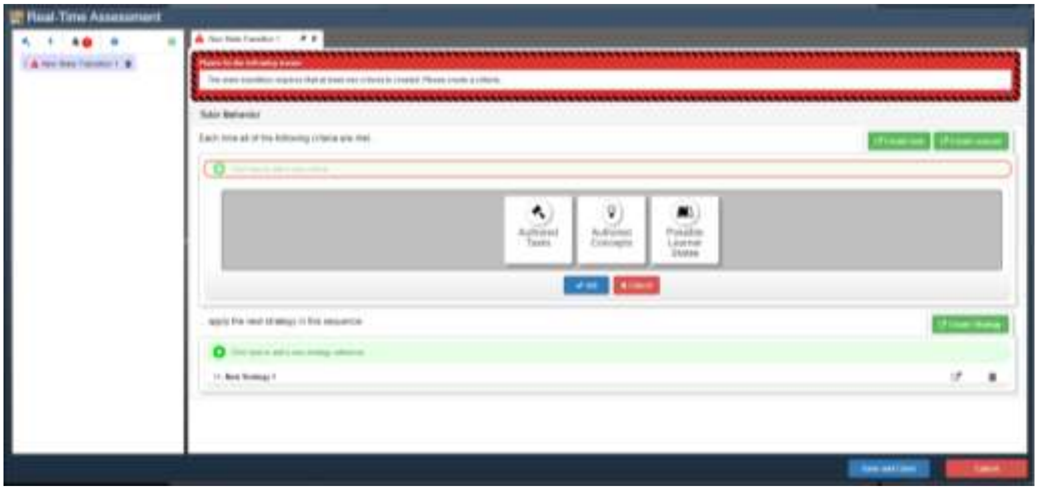


Figure 9. State Transition Panel

9. 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.

10. 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 event to complete the core assessment logic for your training scenario.

Now that you have all of the pieces of your DKF put together, it is very important to perform as many iterations 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

Authoring a DKF can seem like an intimidating task, especially when you consider all of the elements involved and the level of complexity that is possible. While this paper only touches on the basic DKF functionality and capabilities, it is intended to serve as a user-friendly introduction to the full-suite of tools available with this software. Whether you are looking to improve your current team-based training methods or serving some other domain, GIFT has the tools to help alleviate instructor workload and improve training outcomes.

REFERENCES

Burmester, E. (2021). Authoring Collective Training Demonstrations in GIFT, 2021 Update. Proceedings of the 9th Annual GIFT Users Symposium (GIFTSYM9). 28 Apr. 2021, https://tradem.gifttutoring.org/attachments/download/4104/giftsym9_proceedings_FINAL_1.0.pdf.

Burmester, E. (2020). Authoring collective training demonstrations in GIFT. Proceedings of the 8th Annual GIFT Users Symposium (GIFTSYM8). 28 Apr. 2020, https://gifttutoring.org/attachments/download/3708/giftsym8_proceedings.pdf

Department of the Army. (2016, August 23). Army Training Publication (ATP) 3–21.8: Infantry Platoon and Squad. Army Publishing Directorate. Retrieved April 26, 2022, from https://armypubs.army.mil/epubs/DR_pubs/DR_a/pdf/web/ATP%203-21x8%20FINAL%20WEB%20INCL%20C1.pdf

Domain Knowledge File, GIFT Wiki. (2021, May 12). Retrieved May 9, 2022, from https://gifttutoring.org/projects/gift/wiki/Domain_Knowledge_File_2021-2

Hoffman, M., & Ragusa, C. (2015, February). 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).

Sinatra, A. M. (2021, May). The 2021 Authoring Guide to GIFT. In Proceedings of the 9th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym9) (p. 25). US Army Combat Capabilities Development Command–Soldier Center.

Wang, L., Shubeck, K., Shi, G., Zhang, L., & Hu, X. (2020, May). CbITS authoring tool in GIFT. In *Proceedings of the 8th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym8)* (p. 69). US Army Combat Capabilities Development Command–Soldier Center.

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

***Elyse Burmester** is a Test Engineer 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 (DEVCOM-SC) SFC Paul Ray Smith Simulation & Training Technology Center (STTC). Ms. Burmester has a bachelor degree in Political Science from Florida Gulf Coast University.*

THEME V: COLLECTIVE/TEAM BASED METHODS

Formalizing Adaptive Team Feedback in Synthetic Training Environments with Reinforcement Learning

Andy Smith[1], Randall D. Spain[1], Jonathan Rowe[1], Benjamin Goldberg[2], and James Lester[1]
North Carolina State University[1]
U.S. Army Combat Capabilities Development Command (DEVCOM) - Soldier Center[2]

INTRODUCTION

The importance of training teamwork skills is well documented. Teams that share information, provide backup behaviors and support, provide guidance as situations change, and communicate efficiently are more effective than teams that do not engage in these behaviors. A significant challenge faced by designers of adaptive instructional systems (AISs) is devising computational approaches that can mimic the behaviors of a human coach during team training scenarios to provide teams with feedback, coaching, and instructional decisions that can be used to improve taskwork, teamwork, and team effectiveness.

There is growing evidence that machine learning techniques, including reinforcement learning (RL), provide an effective data-driven approach to tutorial planning in AISs (Chi, et al., 2011; Rowe & Lester, 2015; Shen et al., 2018). RL-based tutorial planning systems have 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. RL-based techniques have been employed across multiple educational and training domains, ranging from tutorial planning for narrative-centered learning environments (Rowe & Lester, 2015; Sawyer et al., 2017; Wang et al., 2018), and intelligent tutoring systems for logic proofs (Shen et al., 2018), to sequencing concepts for elementary mathematics education (Mandel et al., 2014), and to adaptive remediation to support cognitive engagement during online training (Fahid et al., 2021; Spain et al., 2021a; Spain et al., 2021b). However, these efforts have focused on individual learners, leaving an opportunity to extend these approaches for team-based training scenarios.

In this paper, we investigate how an adaptive team feedback framework (Spain et al., 2021a; Spain et al., 2021b) can be operationalized and extended using RL to support automated coaching for teams during team-based synthetic training exercises. Using a course we designed in the Generalized Intelligent Framework for Tutoring (GIFT) to provide feedback and coaching to support gunnery crews completing crew gunnery training tables in Virtual Battlespace 3 (VBS3), we examine how key components of RL-based tutorial planners, such as state definition, action definition, and reward function, can be constructed to operationalize key components of the adaptive team feedback model, including learner and team attributes, feedback content and timing, scenario adaptations, and within-task and longitudinal performance outcomes. We discuss challenges and tradeoffs of different approaches for developing automated coaching policies, such as technical constraints relating to the amount of data required to train models depending on the granularity of representations, as well as practical constraints associated with the nature of crew gunnery training tasks, which serves as the use case for our work. Finally, we describe efforts to integrate computational models of adaptive team feedback and coaching into the Learning Effect Model (Sottilare et al., 2018), highlighting capabilities that currently exist to support RL-driven coaching and feedback, as well as features that should be prioritized in future GIFT development efforts to support a generalizable RL-driven coach for team-based training exercises.

RESEARCH CONTEXT

AISs offer a number of affordances for automatically delivering feedback to support learning in simulation-based training environments. 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 student learning needs, assessing and diagnosing problems, and providing coaching and assistance as appropriate (U.S. Army, 2017).

At last year's GIFT Users Symposium, our team presented an adaptive team training feedback framework that describes a set of feedback strategies and tactics that could be implemented in an AIS for teams (Spain et al., 2021a). The framework is devised towards explicit feedback and outlines a broad range of feedback variables that can shape the training experiences of both individuals and teams, as well as impact the overall effectiveness of adaptive team training exercises. The outcome of this framework has been the identification of a set of instructional techniques that are compatible with a data-driven coaching framework that can be encoded as adaptable event sequences that are generalizable across training contexts. We have identified three generalizable strategies to investigate:

- Feedback and coaching on individual task performance (role execution),
- Feedback and coaching on team coordination, and
- Scenario adaptations that increase the difficulty or restart training tasks based on crew performance.

These strategies build upon the team's prior work investigating data-driven tutorial planning in an adaptive hypermedia course (Spain et al., 2019) and can be generalized across simulation-based team training scenarios. Each of these three instructional strategies is mapped to a lower-level adaptive event sequence that encodes a set of corresponding instructional tactics. The instructional tactics include coaching and feedback statements and adaptation decisions that address the following questions:

- Upon observing an error, should the AIS provide feedback or wait and continue to observe performance?
- If the decision is to provide feedback, what type of feedback should be presented to the trainee?
- Should the system provide corrective feedback (e.g., "You did not include X in your plan, which would result in Y") or probe the trainee to engage in self-directed reflection to diagnose the error (e.g., "Take another look at your plan; did you include everything?")?
- How can the scenario be adapted to promote mastery learning?

The feedback literature suggests that decisions about when and how to provide feedback are highly contextual. Although it is important that adaptive feedback be deeply informed by theory, existing models of team training are often not sufficiently prescriptive to guide how feedback decisions should be made in every particular situation. Devising empirical, data-driven models that drive key decisions about team feedback variables is critical for delivering effective training that enables learners to achieve and maintain skill and collective proficiency.

CREW GUNNERY TRAINING IN VIRTUAL BATTLESPACE 3

To support design and develop adaptive team feedback models with GIFT, we are using crew gunnery training in VBS3 as a testbed. VBS3 is an immersive virtual training environment that the U.S. Army uses to support individual and collective training requirements. Specifically, we are using a set of VBS3 missions that emulate real-world training and qualification courses that gunnery crews use for live-fire training and qualification. The virtual scenarios, which were developed at the Warrior Skills Training Center (WSTC) in Fort Hood, Texas, offer crews the opportunity to practice and rehearse crew coordination activities prior to engaging in live-fire qualification exercises. The VBS3 crew gunnery scenario involves a series of six engagements that allow gunnery crews to practice coordinating action in order to carry out the direct fire engagement process. Each engagement requires crew members to detect, identify, and engage one or more moving or stationary targets. A key component of the direct fire engagement process is the coordination of actions and behaviors among the vehicle commander, gunner, and driver. Once a threat has been identified, crews engage in a fire command sequence (Figure 1), which is a well-defined protocol for communicating information and actions to facilitate a coordinated response to a threat. Our crew gunnery assessment model, which is represented in GIFT as a domain knowledge file (DKF), includes a series of concepts for each engagement that can inform coaching at the crew and individual level to remediate errors in the direct fire engagement process.

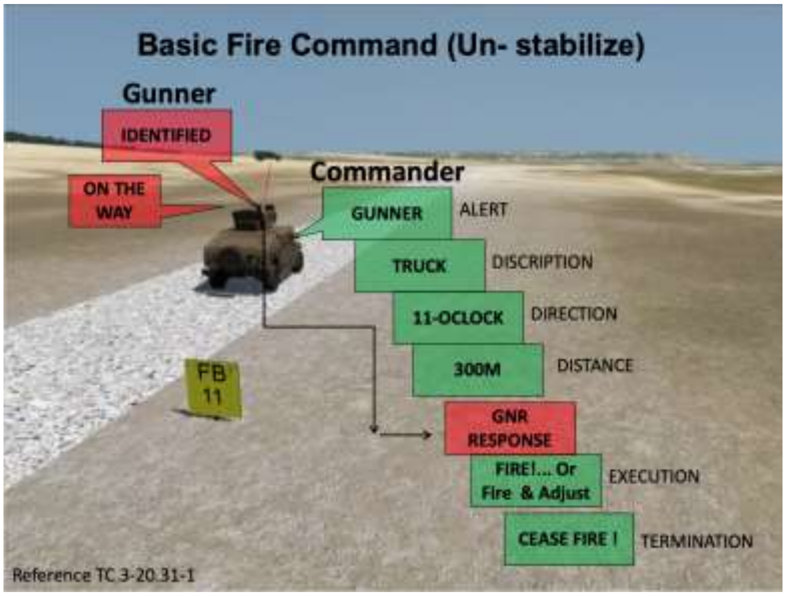


Figure 1. Example of basic fire command sequence for an unstabilized gunnery platform.

REPRESENTING TEAM COACHING MODELS THROUGH REINFORCEMENT LEARNING

RL is a family of machine learning techniques focused on creating agents that perform actions in an environment to optimize a numerical reward. A common framework for these problems is a Markov decision process (MDP), which provides a principled mathematical model for stochastic problems involving sequential decision making under uncertainty. MDPs can be represented as a set of states *S*, a set of actions *A*, a set of transition functions *T*, and a reward function *R*. Using one of several different optimization functions, the goal of RL is for an agent to learn a policy of which actions to take at what states to maximize the reward signal.

As illustrated in Figure 2 below, in the next sections, we apply this framework to the task of crew gunnery training and discuss how different aspects of the team training paradigm map to different facets of an MDP.

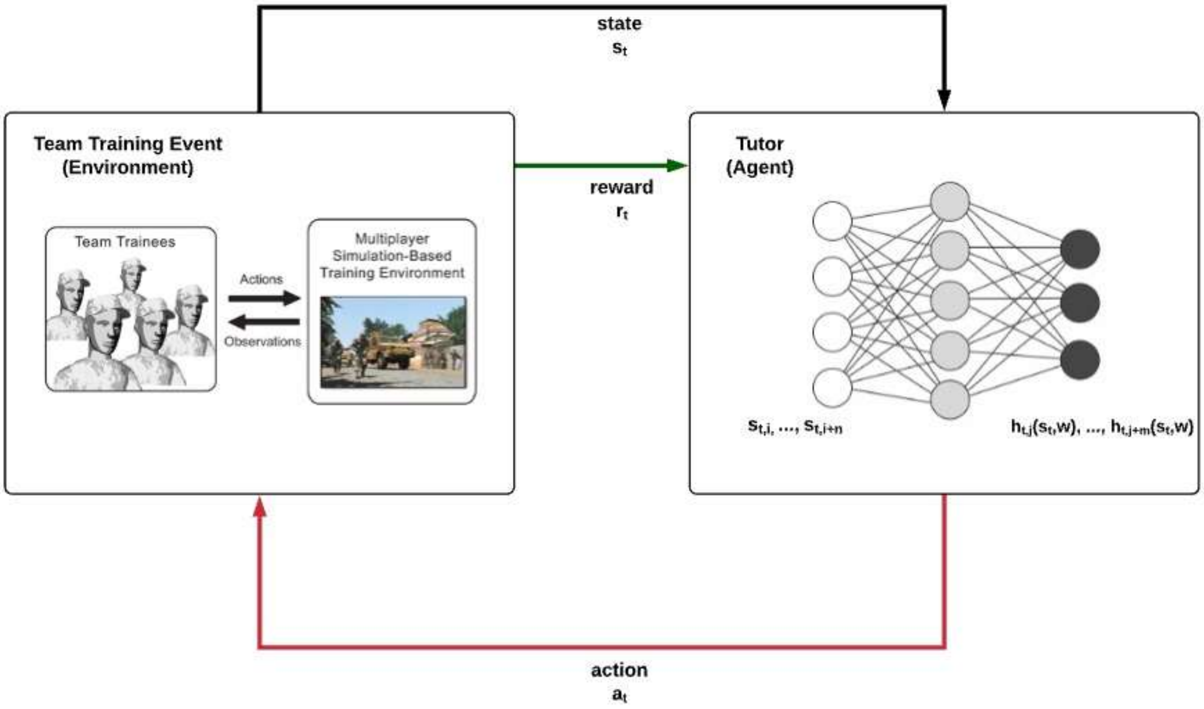

state
s_t
Team Training Event
(Environment)
Team Trainees
Actions
Observations
Multiplayer Simulation-Based Training Environment
reward
r_t
Tutor
(Agent)
$s_{t,i}, \ldots, s_{t,i+n}$
$h_{t,j}(s_t,w), \ldots, h_{t,j+m}(s_t,w)$
action
a_t

Figure 2. Integration of Team Training Environment with RL-based Tutorial Agent

State Representation

Depending on the application, *state* can have many different representations in an RL-based model. In a broad sense, *state* provides context to the model to help it better determine what action to take. For example, an agent being trained to play the card game Blackjack would likely have a *state* representation that includes information about their cards, the dealer's cards, and potentially other information like the remaining bankroll or cards from previous hands. From a training perspective, the *state* representation can be used to measure estimated student competencies based on prior exercises completed for instructional sequencing (Doroudi et al., 2019), model student problem solving trajectories (Rafferty et al., 2016), or track current progress, prior knowledge, and previous remediation history (Spain et al., 2021a; Spain et al., 2021b). *State* representation is a difficult challenge, as not including enough information can limit the effectiveness of the model, though too granular a representation can lead to sparsity issues where states are not encountered enough in training to learn the impact of different actions, though these issues can sometimes be mitigated through automated feature selection techniques (Mitchell et al., 2013; Shen & Chi, 2016).

In the context of the crew gunnery scenario described above, we have focused our attention on two main areas: *Incoming Characteristics*, and *Current Mission Characteristics*. For *Incoming Characteristics,* the goal is to capture relevant information about the prior knowledge and abilities of the trainees before entering the course. One approach is to assess this information using survey instruments administered before attempting the training activity. However, this approach is not suitable for this training domain. Another approach is to utilize incoming Soldiers' previous training ratings in related competencies for this task (i.e., crawl/walk/run). This competency level could be represented as one score for the entire team, or a larger representation encompassing the individual ratings of each team member for that role in the training task. Additionally, other non-task performance characteristics of the team and/or team members could be encoded and included in this representation, such as behavioral characteristics of the team (grit, personality, etc.) or a more general rating of teamwork ability. Such a rating could be valuable for determining tutorial actions, as the system may consider different interventions theorized to support team performance rather than task performance depending on the incoming levels of task and teamwork proficiencies.

Current Mission Characteristics represent information needed by the tutorial agent to reason about the team's progression and performance in the training activity. As described above, the team gunnery mission features multiple separate tasks/engagements, with each task containing several relevant sub-tasks as well. An effective state representation for this type of course must therefore be able to differentiate tutorial actions based on how far into the task the team is, as well as how they have performed on tasks up until that point. It may also benefit the system to include information about what types of tutorial actions (i.e., types of feedback, mission modifications) have already been administered in the current session.

In its current state, GIFT supports aspects of these representations, allowing for competency and training levels to be imported and utilized by the DKF. However, GIFT support for these features will need to be expanded to support different numerical representations of competencies. GIFT also supports extraction of various potential *Current Mission Characteristics* from log files, though it does not currently provide much flexibility for defining them in the course DKF.

Action Representation

Actions in a RL-based system can take many different forms, even when limited to educational and training tasks. Depending on the formulation of the problem, *actions* can include choosing between different instructional micro-tactics (Ausin et al., 2019; Chi et al., 2011), concept sequencing (Mandel et al., 2014), or feedback types (Spain et al., 2019).

For the crew gunnery task, we have identified three families of *actions*. The first is varying the type of feedback given. Here, we specify between different levels of feedback engagement (Elicit vs. tell). Feedback engagement refers to whether the message or coaching statement is meant to be a corrective statement whereby the coach or tutor tells the leaner what she or he should do (e.g., "You need to do X") or an elicit statement that asks the learner to answer a question (e.g., "Can you identify what your team did incorrectly at this point in the scenario? What caused this breakdown?"). This distinction is rooted in the human tutoring literature which shows tutors switch between asking questions and telling learners what he or she needs to do (Lepper et al., 1997). It is also frequently found in coaching practices wherein a coach may ask a player what they should do on a particular play and then follow this up with a tell statement affirming or correcting the player's statement. The difference between these two coaching styles is hypothesized to impact a trainee's level of cognitive engagement. A reflective statement that directs a trainee to constructively self-reflect or to engage in dialogue with other team members to critique their collective performance may foster deeper, more meaningful learning experiences than a corrective coaching statement that is directed to learners but does not require an overt response (Chi, 2009). Adaptively scaffolding cognitive engagement is a central challenge in the development of AISs for individuals (Fahid et al., 2021) and remains an open research question in the design of AISs for teams. Determining when to provide corrective feedback versus when to elicit information from trainees is a critical component of developing effective AISs that mimic human tutors and coaches.

The second family of *actions* is varying the timing of the feedback. The TeamCoach Framework for Adaptive Team Feedback in Synthetic Training Environments (Spain et al., 2021a; Spain et al., 2021b) identifies three levels of feedback timing: immediate, mid-action, and after-action. *Immediate* feedback refers to guidance given directly following a mistake. *Mid-action* feedback is a form of delayed feedback in which feedback and coaching are provided at some breakpoint or lull in the training scenario but before another scenario or task begins. *After-action review* is feedback that is delivered after the training event has concluded. These levels are based upon how feedback timing has been operationally defined and investigated in the simulation-based training literature. However, defining feedback timing rules that are more flexible than the traditional triad of immediate, delayed, after action, could allow for the development of timing policies that more accurately mimic human coaching practices. Depending on the goal of the drill or the scenario and the trainee's level of mastery, a coach may adopt and apply different feedback timing policies. A coach may not intervene until they observe the player commit the same mistake twice or the "*n*-th" instance of an error (e.g., "Hey -- you've forgotten to issue the Identify command twice now, and that's something that must be included in your fire command."). Using a data-driven approach to implement feedback timing policies would allow for more flexible scaffolding of feedback timing based on learner and team states. It would also allow for the development and investigation of feedback timing policies that are tailored for positive feedback statements versus corrective feedback statements. Positive feedback statements are reinforcement statements that are provided in direct response to desirable performance whereas corrective feedback statements are reinforcement statements that are provided in response to mistakes.

Finally, we plan to investigate scenario adaptations. Unlike feedback, which is delivered to the trainee during or after a task, scenario adaptations change the current or future task. For example, when observing crews were going through the training task at WSTC, it was common for the instructor to have the crew repeat an engagement if they failed it on a previous attempt, rather than move directly to the next engagement. Additional types of mission adaptations could be to vary the difficulty of upcoming engagements, such as having moving targets rather than stationary, or changing weather conditions in the virtual environment.

From a feedback perspective, GIFT offers great flexibility in authoring immediate feedback conditions, as well as supporting different feedback modalities that can be delivered through GIFT's Tutor User Interface (TUI). We are currently working with Dignitas to develop functionality supporting delayed feedback through summative assessments delivered in between engagements rather than immediately after an error occurs. GIFT also supports directing feedback to teams as a whole or to individuals. GIFT currently provides the capability to drive mission adaptations in VBS3, such as changing the time of day and weather. More complex adaptations can be triggered as well, but authoring the adaptations must be done through the VBS3 scripting interface.

Reward Representation

A well-defined reward function is key to any successful RL-based system, resulting in many different approaches to reward learning and reward engineering. From a training perspective, rewards can take several forms. Some examples include learning outcomes, as measured by a pre and post survey assessment (Wang et al., 2018), task efficiency (Ausin,2019), and student engagement (Sawyer et al., 2017). Additionally, different rewards can be combined by multi-objective RL systems to produce models responsive across multiple factors, such as engagement and task performance (Sawyer et al., 2017).

For team training tasks such as crew gunnery, it is important that the reward incorporates both task performance (taskwork), as well as team performance (teamwork). Taskwork skills are the technical skills that are needed of team members to execute a task or mission. Taskwork skills are role-specific and do not require interdependent interaction with other team members. Teamwork skills, on the other hand, are the interdependent components of performance required to effectively coordinate the performance of multiple individuals. They are the skills, including attitudes and behaviors, that are required to function effectively as a team. Training that aims to improve team performance should be designed to promote the development of team competencies such as communication, coordination, shared cognition, and team efficacy (Salas et al., 2008).

Defined assessment criteria are required for both types of performance throughout the training exercise so that both can be measured and fed into the policy training pipeline. To assess taskwork, it is important to consider both embedded and external assessments. For example, in the crew gunnery task, it is logistically unlikely that teams will take a pre- and post-survey assessment before completing the training activity. Therefore, for the system to produce a reward, it must be based on performance within the training mission. This can either be generated throughout the engagements or be generated after a "culminating" activity at the end of the training mission. These decisions must be made carefully, and with considerations to the other design decisions about tutorial actions and state representations, so as to avoid the model learning undesired behaviors. An example of this could be if the taskwork assessment is purely based on performance, then one could imagine the model favoring mission adaptations making the tasks as easy to complete as possible.

Likewise, if the goal of the reward is to promote the development of teamwork skills, then assessments have to be designed and mapped to teamwork competencies as opposed to performance outcomes. For our crew gunnery scenario, we have identified communication, coordination, and information exchange as important team-level competencies that should be remediated. In addition to these process-oriented factors, rewards can also be modeled to support the development of team-level attitudes, such as team efficacy, which reflects how confident a team is in their ability to achieve an upcoming goal with a high degree of success. Representing teamwork as a multidimensional construct could promote the use of multi-objective RL-driven coaching decisions that can be used to improve team processes and team beliefs.

Data Collection and Model Training

After defining the states, actions, and rewards, there are still design decisions to be made about how to collect data to best induce the data-driven pedagogical policies. One method for training RL policies is to train the policy based on data collected from detailed traces of users completing the training scenario with either a human instructor making feedback choices or some automated policy making feedback choices. This method is referred to as *offline RL. Offline RL* is easier to collect data for, as the data can be collected, aggregated, and processed before being fed through the policy training pipeline. It is important that the data collected cover as much of the state/action space as possible. This is usually done by implementing a "random" policy to guide feedback, though in training contexts this is often replaced by a "random yet reasonable" policy to avoid undesirable outcomes like a trainee receiving negative feedback on a correct action. Another option is to collect data using human instructors. This can be beneficial as the system will theoretically learn to mimic the behavior of "experts." In this case, it is important that the tutorial actions and states be defined in a way that captures what the human instructor is seeing and doing. This can be achieved by having the human instructor not observe the training directly, but rather view the same data stream the automated system would have, with the same set of potential actions as the system as well.

A core assumption of policies trained offline is that the reward signal will not change over time, or that the future trainees will respond in the same fashion as the set of trainees whose data the system was trained with. To accommodate for potential changes in user behavior, *Online RL* can be utilized. *Online RL,* while logistically more difficult to implement in many cases, has the attractive feature of being "self-improving," meaning the system adapts over time as more users use the system. In this case, the RL system must be built in a way as to handle the exploration exploitation trade-off. Exploration refers to trying new actions, rather than exploitation which refers to the system only performing whatever action it believes to be optimal at that given time. This trade-off can be handled through author-driven methods where the rate of exploration is explicitly defined through model parameters, or through automated methods where the system dynamically adjusts its confidence in various aspects of the policy and adjusts its exploration rate accordingly. *Online RL* systems can also be deployed in a way that they are initially bootstrapped with policies generated by an offline process. This can help the system more quickly perform effectively, while continuing to refine and revise its policies over time as more users complete the task.

While GIFT currently supports much of the infrastructure required for collecting training data at scale, there are some enhancements needed to enable it to be an effective platform for training and deploying data-driven systems. GIFT currently does not support stochastic authoring, meaning it is not possible to author courses with a "random" policy. Beyond random policy, it would be useful if GIFT courses could support different sampling algorithms, to ensure that the state/action space is explored as fully as possible. For human-driven tutoring, GIFT Game Master provides an excellent interface for providing instructors with a data-stream matching that of the model and providing a discrete set of tutorial actions through *Scenario Injects*. While we have authored policies through a JSON configuration file in a previous project, more flexibility in authorship would be necessary for more complex models. Additionally, for GIFT to support *Online RL,* GIFT courses would need to be able to update the policies as users progress through the system, as well as supporting mechanisms for managing the exploration vs. exploitation tradeoff. GIFT will also need to support improved tools for developing data pipelines capable of converting rich trace-log data from the training environments into state and reward representations required by the RL models.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

Delivering tailored coaching and feedback to team members in synthetic training environments is a critical challenge. Building on learning theory-based models of team coaching, we seek to operationalize these models in a RL framework capable of leveraging advances in machine learning and data-driven pedagogy. Mapping to an RL framework requires careful design decisions about how to most effectively represent state, tutorial actions, and rewards to ensure the model can support teams in both taskwork and teamwork. While GIFT supports some of these facets, there are also targeted improvements that can greatly benefit the development and evaluation of these systems. Future research will implement these models, and use them to investigate the effectiveness of different learner and team state representations, different types and timings of feedback, and different measures of both teamwork and taskwork effectiveness. Additional research goals will be to use the data collected with gunnery crews to investigate methods of incorporating automated assessments of these competencies into GIFT, as well as informing the design of simulated users to facilitate development of these data driven models in the future.

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. (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).

Chi, M., Jordan, P. W., Vanlehn, K., & Litman, D. J. (2009). To elicit or to tell: Does it matter? In *Proceedings of the 2009 conference on artificial intelligence in education*, (pp. 197–204). IOS Press Amsterdam.

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. (2019). Where's the Reward? *International Journal of Artificial Intelligence in Education*, *29*(4), 568–620.

Fahid, F. M., Rowe, J. P., Spain, R. D., Goldberg, B. S., Pokorny, R., & Lester, J. (2021). Adaptively Scaffolding Cognitive Engagement with Batch Constrained Deep Q-Networks. In *International conference on artificial intelligence in education* (pp. 113-124). Springer, Cham.

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

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).

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).

Rafferty, A. N., Jansen, R., & Griffiths, T. L. (2016). Using inverse planning for personalized feedback. In *Proceedings of the 9th international conference on educational data mining*, (pp.472–477). International Educational Data Mining Society.

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).

Salas, E., Cooke, N. J., & Rosen, M. A. (2008). On teams, teamwork, and team performance: Discoveries and developments. *Human Factors, 50*(3), 540-547.

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).

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.

Sottilare, R., Graesser, A., Hu, X. , & Sinatra A. (Eds.). (2018). *Design recommendations for intelligent tutoring systems: Volume 6 - team tutoring*. U.S. Army Research Laboratory, 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 seventh annual GIFT users symposium* (GIFTSym7) (pp. 111–120). US Army DEVCOM–Soldier Center.

Spain, R., Rowe, J., Goldberg, B., Pokorny, R., Mott, B., & Lester, J. (2021a). Automated coaching in synthetic training environments: Developing an adaptive team feedback framework. In *Proceedings of the ninth annual GIFT users symposium (GIFTSym9)* (pp. 187–199). US Army DEVCOM–Soldier Center.

Spain, R., Rowe, J., Smith, A., Goldberg, B., Pokorny, R., Mott, B., & Lester, J. (2021b). A reinforcement learning approach to adaptive remediation in online training. *The Journal of Defense Modeling and Simulation*, *19*(2), 173-193.

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

ABOUT THE AUTHORS

Dr. Andy Smith *is a Research Scientist in the Center for Educational Informatics at North Carolina State University. He received his M.S and PhD. in Computer Science from North Carolina State University, and his B.S. degrees in Computer Science and Electrical and Computer engineering from Duke University. Prior to graduate school Andy worked as an Underwater Robotics Engineer at SPAWAR SSC Pacific in San Diego, CA. His research is focused on the intersection of artificial intelligence and education, with emphasis on user modeling, game-based learning, and educational data mining.*

Dr. Randall Spain *is a Research Scientist in the Center for Educational Informatics at North Carolina State University. He holds a PhD in Human Factors Psychology from Old Dominion University. His research focuses on the design and evaluation of advanced training technologies on learning and performance.*

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 focuses on artificial intelligence in 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. Benjamin Goldberg *is a senior researcher in the Learning in Intelligent Tutoring Environments (LITE) Lab at the Combat Capabilities Development Command (DEVCOM) Soldier Center, Simulation and Training Technology Center (STTC) in Orlando, FL. He has been conducting research in modeling and simulation 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. 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).*

Moving Beyond Training Doctrine to Explainable Evaluations of Teamwork using Distributed Cognition

Caleb Vatral[1], Naveeduddin Mohammed[1], Gautam Biswas[1], and Benjamin Goldberg[2]
Institute for Software Integrated Systems – Vanderbilt University[1], US Army Combat Capabilities Development Command (DEVCOM) - Soldier Center[2]

INTRODUCTION

Simulation-based environments offer many affordances that streamline the learning process for complex workplace skills compared to traditional learning environments: simulations are easily repeatable and controllable; simulations offer physical and psychological safety for otherwise dangerous tasks; data can be easily collected from simulations to provide evidence-based assessment of learners. Thus, it is not surprising that simulation-based training (SBT) has been widely adopted for training complex cognitive, psychomotor, and teamwork skills for the workplace (Ravert, 2002). Within the Army, this SBT paradigm has been implemented through the development and use of Synthetic Training Environments (STEs), which offer mixed-reality solutions for experiential training of Soldiers on a variety of battle drills (Goldberg et al., 2021). One of the focuses of this STE training is development of effective teamwork skills in learners. Teamwork is a critical component of mission success but is also difficult to evaluate due to the complex cognitive and social components which makeup effective teamwork. One of the most popular frameworks used to analyze teamwork behaviors in simulation-based training is distributed cognition.

Distributed cognition extends analysis of cognition beyond individuals to teams and the environment in which these activities take place (Hutchins, 2000). Cognition in the context of distributed cognition is a dynamic process, including the physical space where cognition is taking place, any objects and tools in the space, the social network of people performing the cognition, and how these processes and elements evolve over time. Thus, distributed cognition provides a comprehensive framework for analyzing how teams of people work together on tasks, including interpretation and co-processing of information, as well as co-construction of knowledge and solutions, by combining the cognitive, metacognitive, and psychomotor skills of the team with the affordances of the environment. Within distributed cognition, the *Distributed Cognition for Teamwork* (DiCoT) model is one of the most effective qualitative analysis frameworks. DiCoT uses five high-level themes to frame its analysis: physical layout, information flow, artefacts and environment, social structures, and temporal evolution (Blandford & Furniss, 2005). Analyzing each of these themes and combining their insights produces a comprehensive qualitative picture of individual and team cognition.

However, while qualitative analysis does offer significant insights and evidence for instructional design and learner evaluation, it is limited due to its reliance on manual analysis by domain experts (Zachary et al., 2000). This is especially limiting for training domains which require development of rapid decision making and muscle memory, as training these skills generally requires a high volume of short, repeated practice exercises. This is often the case in the Army training domain. Because of this issue, evaluations of teamwork performance during training are typically quantitative and generated by analysis of the data collected from the training environment. However, historically these data-driven evaluations are primarily based on specific training doctrine (Chapman, 1991), rather than cognitive frameworks that capture the full scope of trainee thoughts and behaviors. Motivated by this gap between traditional doctrine-based performance metrics and qualitative analysis based on complex cognitive frameworks such as distributed cognition, in this paper we propose and develop methods to bridge the gap between qualitative cognitive analysis and quantitative doctrine-based analysis by providing graphical feedback mechanisms grounded in distributed cognition alongside the quantitative performance evaluations. By providing evidence-driven explanation of automated performance metrics, we can not only instill more confidence in the automated assessments, but also provide more detailed formative feedback to improve learner outcomes.

Our automated performance assessment and graphical feedback mechanisms are implemented in the Generalized Intelligent Framework for Tutoring (GIFT) as an *external assessment engine (EAE)* (Sottilare et al., 2012). The EAE applies Artificial Intelligence (AI) and machine learning methods to automatically calculate performance evaluation metrics, as well as selecting and generating graphical feedback for display to trainees and instructors to help explain how their performance was being evaluated. To demonstrate the integrated system, we present a case-study of squads of Soldiers training on *Enter and Clear a Room (ECR)*, a dismounted battle drill designed for training conditions found in modern urban warfare. Through the case-study, we demonstrate the architecture of the EAE in combination with GIFT, as well as providing examples of both the automated performance evaluations and graphical feedback mechanisms. We hope that by utilizing the feedback provided by our system as part of a comprehensive training program, trainees will have a better understanding of their behaviors and the implications on team performance.

CASE-STUDY: ENTER AND CLEAR A ROOM

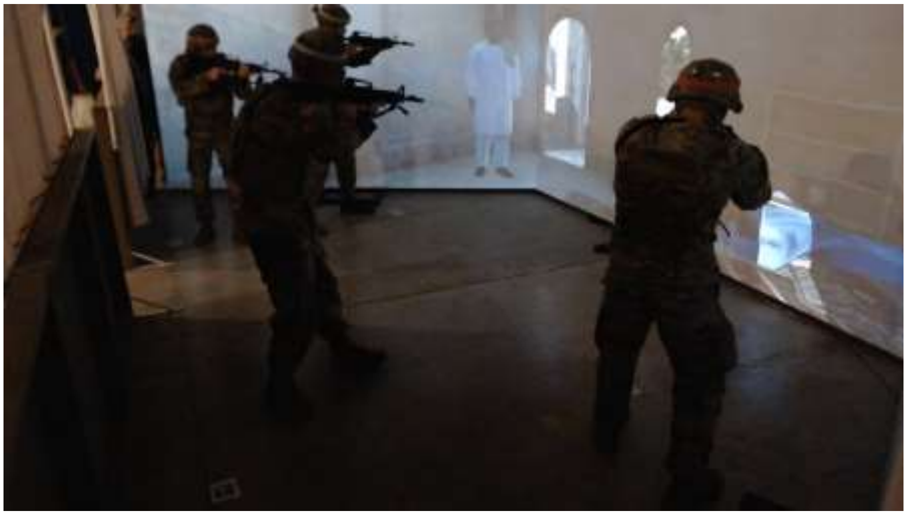


Figure 1. Example of the ECR drill being conducted in the SAM-T (Squad Advanced Marksmanship Trainer) training environment

To help frame the presentation of our EAE architecture and provide examples throughout the remainder of the paper, in this section we present the case-study used during development of our system. Two squads of three and four Soldiers, respectively, participated in the study at the Fort Campbell US Army installation, each performing training on the ECR dismounted battle. ECR is a dismounted battle drill designed to simulate operations often seen during modern urban warfare. In the drill, the squad enters a room with the goal of neutralizing all enemy combatants. The squad does not have knowledge of the layout of the room before entering, and each room may contain any combination of enemy combatants, civilian non-combatants, physical obstacles, and unique weapons. Once the operation commences, Soldiers successively and rapidly enter the room, following paths of least resistance along the walls and neutralizing any enemy combatants in their sectors of fire. Once all combatants have been cleared and all civilians are secured, the squad members successively search each entity and remove all weapons, as directed by the squad leader. Once this operation is complete, the team exits the room, vocalizing their exits to ensure no fratricide occurs. The ECR training in this case study used the *Squad Advanced Marksmanship Trainer* (SAM-T), which is a mixed-reality synthetic training environment designed for simulating live-fire weapons training drills

(Gant et al., 2019). ECR on the SAM-T system consists of three screens setup in a U-shaped arena, on which a Virtual Battle Simulator 3 (VBS3) simulation of the room is projected. Soldiers move around in the space created by the U-shaped arena to simulate moving around in the room and fire their weapons at the on-screen entities. Data was collected from SAM-T including weapon aim and firing, Soldier biometrics, simulation logs, and two video cameras. Each instance of the training drill took approximately 2 minutes and each squad performed between 20 and 30 drills. Between each drill, the Soldiers were given feedback and scaffolding on their performance by a trained instructor. Notes about the scenarios used for training, conversations among the squad members, and the feedback given to the Soldiers by instructors were recorded by the researchers.

THEORETICAL FRAMEWORK

In this section, we present the theoretical framework used for the design of the EAE including automated performance evaluation with the *hierarchical ABC (**A**ffective, **B**ehavioral, **C**ognitive) teamwork model (H-ABC)* (Bell et al, 2018; Vatral et al., 2022), the distributed cognition perspective on learner modeling and teamwork, and the integration of these two techniques into a cohesive learner behavior and performance analysis and feedback system.

Automated Performance Evaluation with H-ABC

Within simulation-based training, automated evaluation of learner performance is typically accomplished through analysis and fusion of multimodal data collected from the training environment (Biswas et al., 2020; Goldberg et al., 2021; Vatral et al., 2021). The multimodal data collected varies depending on the specific training application being utilized but is generally designed to capture traces of the learners' actions and behaviors while training in the system. In our case study, this includes both low-level sensor data – for example, biometrics and video – as well as high-level simulation action data – for example, weapon fire and VBS entity positions. On its own, this data does not directly provide information about trainee performance, but when combined with an interpretation model that includes domain-specific doctrine, this multimodal data can be used to generate comprehensive evaluations. In our work, we interpret the simulation data using the *H-ABC* model (Vatral et al., 2022).

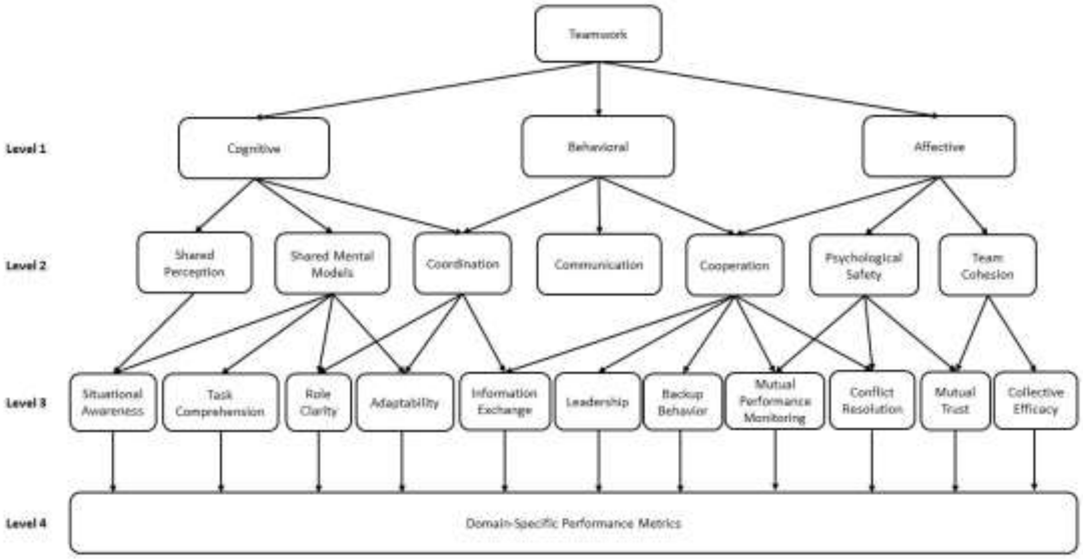


Figure 2. The H-ABC model for evaluation of teamwork behaviors, as developed in Vatral et al. (2022)

The H-ABC model is a hierarchical model of teamwork that is designed using techniques from cognitive task analysis (Vatral et al., 2022; Zachary et al., 2000). At the highest level, the model includes three concepts to describe abstract teamwork: affect, behavior, and cognition. At each level of the model below this, a series of teamwork constructs, such as adaptability, coordination, and trust are defined and linked to concepts to which they contribute in the level above them. For example, mutual trust and collective efficacy both contribute to team cohesion, which in turn contributes to evaluation of affect. At each deepening level of the model, the concepts included become more concrete and observable. The H-ABC model up to level 3 is shown in Figure 2. In order to measure these teamwork concepts at the lowest level, we define a series of domain-specific performance metrics that can be calculated from the collected multimodal data using a variety of AI and machine learning techniques. For the case-study presented here, we use five ECR-specific performance metrics that are defined by the training doctrine of ECR and developed based on review of the relevant literature and discussion with domain-expert instructors. These five metrics are all calculated using computer vision techniques applied to the video collected from SAM-T, and they are briefly summarized in Table 1. For further discussion of metric design and calculation, see Vatral et al. (2021). These calculated performance metrics can be provided directly back to trainees and instructors as evaluations of performance, but they can also be propagated back up the links in the H-ABC model to evaluate higher-level teamwork behaviors. These higher-level results can also be provided back to trainees and instructors and combined with evidence and scores from other training environments to generate a more comprehensive analysis of team performance; however, this combination with other evidence is currently reserved for future work. For further discussion of the H-ABC model, see Vatral et al. (2022).

Table 1. The five metrics used for automated performance evaluation in the ECR SAM-T case-study

Metric Name	Description	Calculation
Points of Domination (POD)	How well Soldiers reach and maintain their PODs	Normalized minimum Euclidian distance between Soldiers and their PODs
Move Along Wall	How well Soldiers keep along the walls of the room while entering	Percentage of video frames where Soldiers within a distance threshold of the wall
Entrance Vectors	Do the Soldiers enter the room in the opposite direction of the previous Soldier	Percentage of Soldiers for whom the angle of their entrance vector is opposite of the previous
Total Entry Time	How quickly does the team enter the room once commenced	Normalized difference between team’s entry time compared to the optimal time threshold
Entrance Hesitation	How quickly does each Soldier enter the room after the previous Soldier	Normalized difference in entry time between two successive Soldiers compared to the optimal time threshold

Distributed Cognition and the DiCoT Model

When generating assessment and feedback for learners and trainees, it is important to consider the cognitive framework that underpins the performance analysis. This is especially true when evaluating learners on their higher-level cognitive, metacognitive, and psychomotor concepts, as is required to understand and evaluate teamwork, instead of just their direct task performance. In the traditional view of cognition, the individual learner is the basis unit of analysis. Under this viewpoint, an individual is essentially a symbolic manipulator who collects information from the environment, processes that information, and then produces some set of output behaviors (Clark, 1997). However, this view of cognition does not consider the physical, social, and environmental considerations that affect cognition. Especially in cases where teamwork is of critical importance, such as our ECR case-study, not considering these factors is detrimental to the conclusions which can be drawn from the analysis. To remedy this problem, alternative cognitive systems, such as distributed cognition, have been developed.

Distributed cognition rejects the view of the individual as an isolated unit of cognition, and instead extends the bounds of the analysis to an entire cognitive *system*, including multiple people in social relationship, the artefacts used to aid in task execution, and the physical layout of the environment (Hutchins, 2000). The overall idea of distributed cognition is that the cognition of an individual is inseparable from the social and environmental framework in which they operate, so in order to understand the cognition of any individual, we must analyze the complete system. From this, it is easy to see why distributed cognition has become an increasingly popular framework for analyzing both teamwork and simulation-based training environments, especially mixed-reality training environments. By integrating the sociocultural components of teamwork and the physical and environments characteristics of simulation and mixed-reality, distributed cognition more completely matches and encompasses the components of an effective training system. This makes it an ideal cognitive system for analysis of STEs, such as our case-study.

Several structured methodologies have been developed to analyze the distributed cognition of a given system (e.g., Galliers et al., 2007; Stanton, 2014; Wright et al., 2000), but because of our focus on evaluation of teamwork, we adopt the DiCoT methodology for our analysis (Blandford & Furniss, 2005). DiCoT breaks down distributed cognition into five independent but overlapping themes. (1) *Information Flow* models how information moves between different sources in the environment and how each of those sources transforms the information. (2) *Physical Layout* models how the arrangement of physical objects and people in the environment affects cognition. (3) *Artefacts and Environment* models how the design of the environment, including the presence and use of tools, affects cognition. (4) *Social Roles* models how people interact within the environment and how those various interactions and roles affect cognition. (5) *Temporal Evolution* models how the system and its environment changes over time. By analyzing each of these themes in depth, the DiCoT methodology allows us to understand the distributed cognition which takes place in the training environments.

Integrated Feedback Framework

Each of the analysis methods presented in the previous two sections offers unique perspectives and advantages for learner feedback, but also unique disadvantages. The H-ABC model offers a way to evaluate performance automatically and numerically, allowing us to easily show progression of learner performance over time with little to no input from an instructor or simulation designer after the initial setup, but these performance metrics do not provide explanation to the learners of how they are calculated, why they are important, and what behaviors can be changed to resolve the issues. On the other hand, distributed cognition and DiCoT offers a more in-depth analysis of performance that is interpretable and explainable to learners, allowing the learners to more easily act on the feedback but requires extensive manual qualitative analysis by experts. In order to maximize the learning benefits of feedback while limiting the drawbacks, we propose an integration of both quantitative automated performance assessment and qualitative DiCoT-based analysis through the use of visual feedback mechanisms. A block diagram illustrating our integrated framework is shown in Figure 3. In the figure, the elements of the framework taken from each analysis technique described previously are highlighted in different colors: green represents the synthetic training environment and its associated data; blue represents the automated performance evaluation with the H-ABC model; yellow represents analysis with distributed cognition and DiCoT; and red represents the integration of these components into a unified learner feedback dashboard.

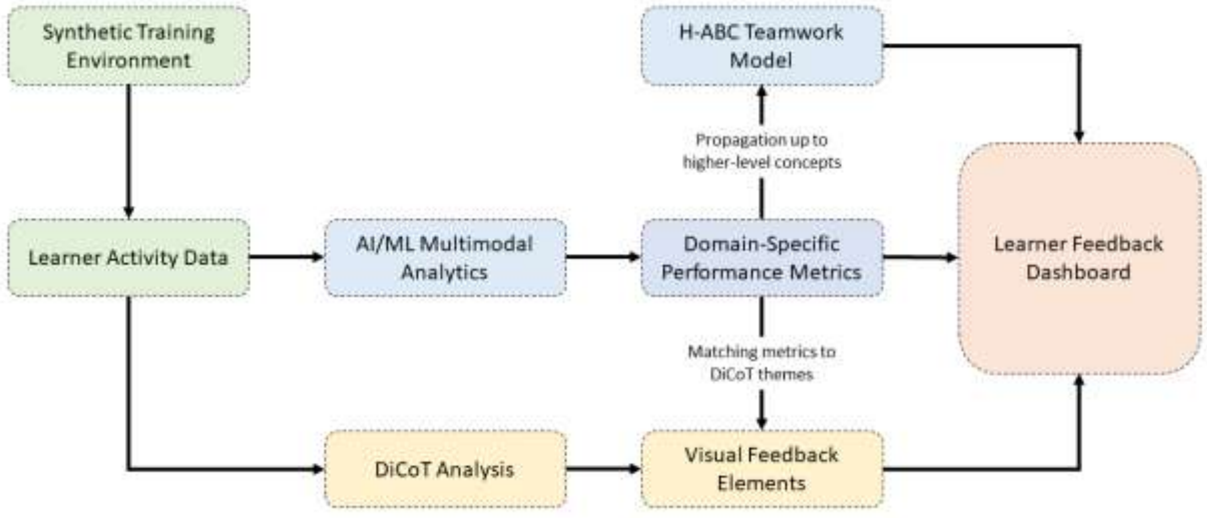

Synthetic Training Environment
H-ABC Teamwork Model
Propagation up to higher-level concepts
Learner Activity Data
AI/ML Multimodal Analytics
Domain-Specific Performance Metrics
Learner Feedback Dashboard
Matching metrics to DiCoT themes
DiCoT Analysis
Visual Feedback Elements

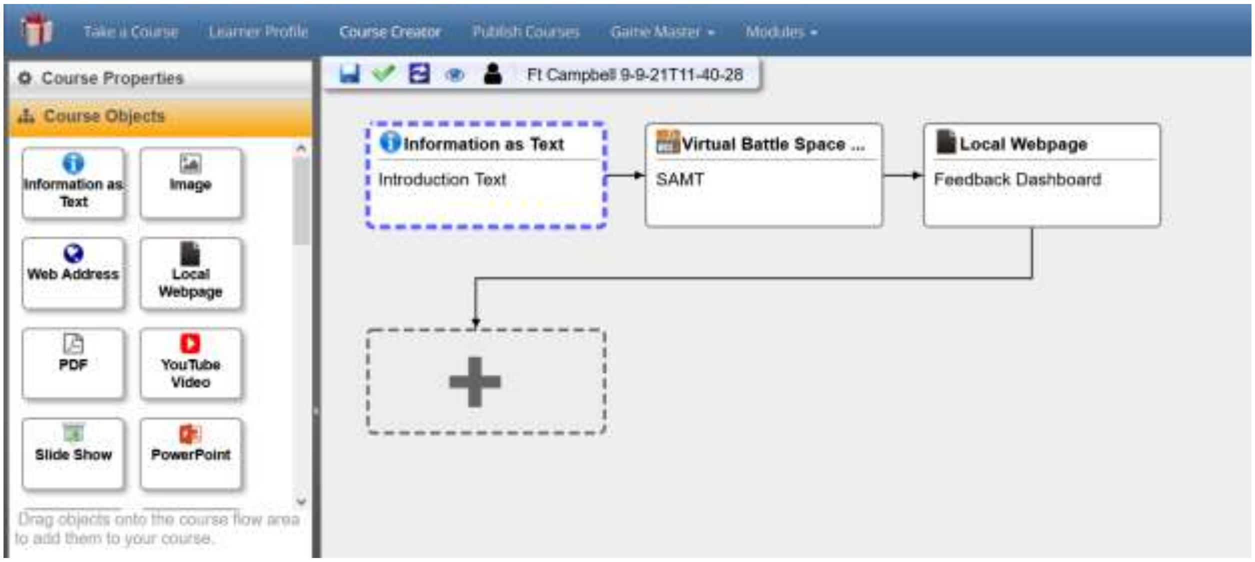
Figure 3. Block diagram of the complete integrated feedback framework

At the center of our integrated framework is the H-ABC model of teamwork, which remains largely unmodified. Just as in its original development, the three-level H-ABC model is extended by an additional layer representing the performance metrics for the domain being analyzed (ECR in our case). These performance metrics and their progression over time can be shown to instructors and trainees as the first form of feedback. However, alongside definition of the performance metrics for this domain, we define mappings of these performance metrics back onto the five themes of DiCoT. By analyzing the specifics of each performance metric's computation and the data involved, we can determine to which of the DiCoT themes a given performance metric belongs. For example, in our case study the *Move Along Walls* metric corresponds closely to the physical layout theme since it examines the physical positioning of the Soldiers as they move through the room, and the *Entrance Hesitation* metric corresponds closely to the temporal evolution theme since it examines the time delay between Soldiers entering the room. After mapping the performance metrics to their corresponding DiCoT themes, we design a visual feedback element, which highlights the characteristics of that theme in the context of the performance metric. For example, with *Move Along Walls* we want to highlight the physical space of the room and how each Soldier moved through this space, so our visual feedback mechanism might show a map of the room and the paths that each Soldier took through the space. To further explain the score of the associated performance metric, we draw on the map to indicate the optimal distance from the wall that Soldiers should be moving and highlight times where they strayed from this distance. For *Entrance Hesitation*, we want to highlight temporal progression, so our visual feedback mechanism might show a timeline of when each Soldier entered the room and the time delay from the previous team member. To further explain the score of the associated performance metric, we could add indications on the timeline of the optimal entry time for each Soldier and the maximum entry time which is considered acceptable.

While these are only two examples of visual feedback design, they represent the underlying principle behind the integrated framework. Performance metrics are designed based on training doctrine, but each also maps to an important component of the distributed cognition model. By showing visual feedback elements alongside the raw performance metrics and progression, the scores become more explainable and actionable to the trainees, so that during the next iteration of training, they can incorporate new behaviors and strategies inferred from the feedback and the discussion with other team members that the feedback will produce. Feedback which is actionable to learners is far more beneficial than simply assessment alone (Gegenfurtner et al., 2014). Thus, by grounding the doctrine-based performance metrics in explainable distributed cognition-based visual feedback, we can improve the feedback we are providing to trainees and hopefully improve learner outcomes.

IMPLEMENTATION IN GIFT

In this section, we will discuss the implementation of our integrated feedback framework in GIFT. First, we will show the architecture of our external assessment engine which receives the SAM-T data messages from GIFT to compute video-based performance metrics. Then, we show the development of the visual feedback elements into a dashboard designed to be displayed to trainees as a course element after their training session.



Drag objects onto the course flow area to add them to your course.

Figure 4. The basic GIFT course flow used for the integrated feedback dashboard

External Assessment Engine

To compute the performance assessments and generate the visual feedback elements, we implemented our feedback framework as a python server which communicates with GIFT as an external assessment engine. In the domain module of GIFT, we add an additional condition class which represents the assessments generated by our EAE. At startup of the course component for domain session log playback, GIFT will trigger our condition class on initialization and pass the relevant videos and SAM-T data for processing. The condition class passes this data to the python server via the XMLRPC interface, where the videos and data are analyzed, and the assessments are generated and queued for playback. Then, the EAE sends a ready signal back to GIFT and playback of the SAM-T domain session logs begins. During this playback, the condition class subscribes to *EST_COLLECTIVE_STOP* messages, which represent when a single run of the SAM-T ECR scenario completes. When it receives this collective stop message, the python server is queried for the assessments that correspond to this scenario execution and the corresponding assessments are updated. This process continues until the final collective stop message is received, at which time the playback is complete, and the learner can advance to the next course component – our feedback dashboard. For more information on the external assessment engine architecture used by our feedback framework, see Vatral et al. (2021).

Feedback Dashboard

In order to present the integrated feedback computed by our EAE back to the learner, we implemented an HTML dashboard to dynamically show the visual feedback elements. In the course flow in GIFT, we setup a local HTML page course component which occurs immediately after the SAM-T log playback. This basic course flow is shown in Figure 4. During the calculation of the performance metrics by the EAE described in the previous section, the EAE also dynamically updates the HTML page with the metrics and visual feedback elements relevant to the given playback. Thus, when the playback ends and the trainee advances to the next course element, the HTML dashboard has been populated with the feedback elements and the trainee is free to explore their performance in-depth using the dashboard.

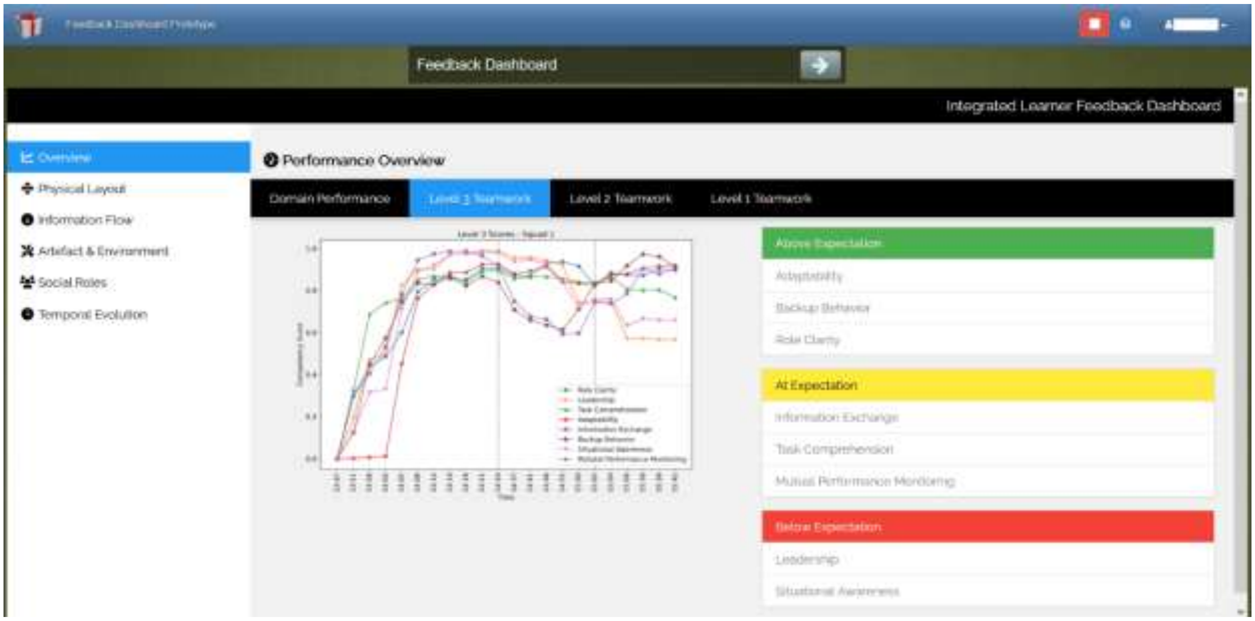


Figure 5. Example of the integrated feedback dashboard displayed as part of a GIFT course

The feedback dashboard is designed based on the five DiCoT themes and the associated visual feedback displayed for each theme. An example of the dashboard displayed within the GIFT course interface is shown in Figure 5. On the left-hand side of the dashboard, the user is presented with six tabs which each change the presentation to highlight specific information. The first tab presents a summary of the team's performance over the course of the training session, showing the competency ratings for multiple levels of the H-ABC model, as well as graphs of performance progression over time. The other five tabs each represent one of the DiCoT themes and presents the visual feedback elements relevant to the selected theme. For example, the *Physical Layout* page for our ECR case-study is shown in Figure 6. Here, we show a map-view of the room with each Soldier's movement paths overlayed. Learners can select between the different runs of ECR, and the map-view dynamically updates to display the relevant figures for that scenario. As additional metrics are added which also utilize the physical layout theme, additional tabs for their respective visual feedback elements can be added to the page. This is shown in the navigation bar for the page, which currently contains two metrics: move along walls and entrance vectors. By allowing the learners to explore this visual feedback, they can begin to understand their performance and how the performance metrics were calculated in more depth, and they can discuss changes that they can make during future training to improve.

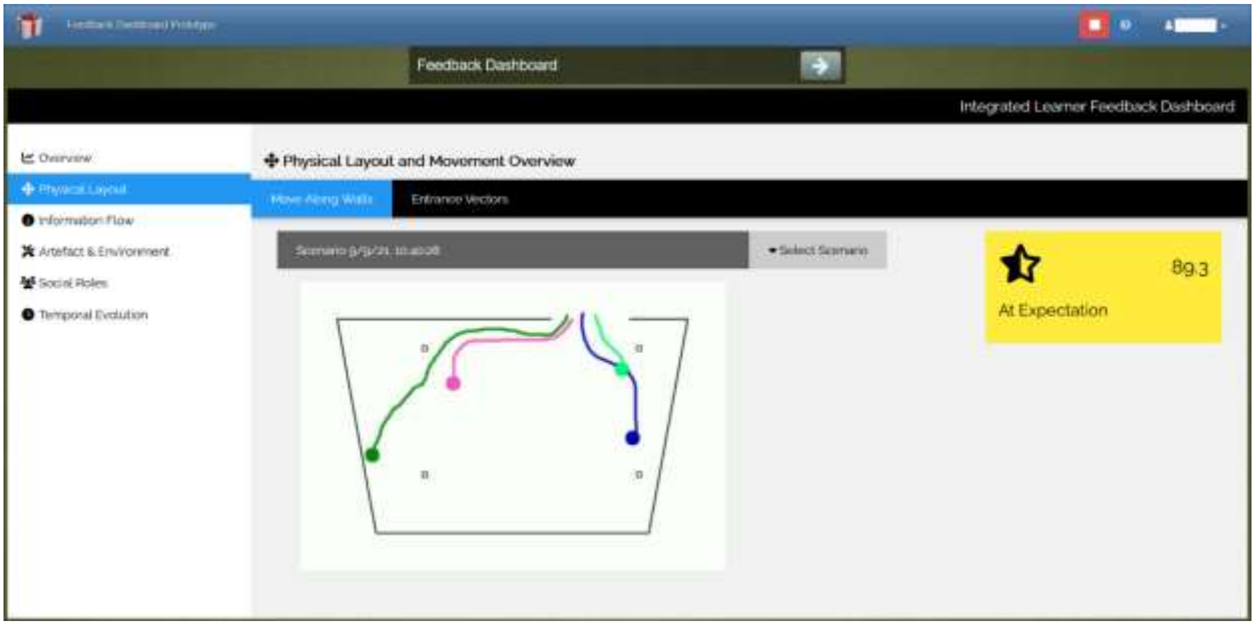


Figure 6. Example of the physical layout page of the feedback dashboard

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

In this paper, we developed a theoretical framework to combine automated quantitative performance assessment with visual feedback mechanisms derived from qualitative DiCoT analysis of distributed cognition. By combining both quantitative and qualitative assessment and feedback methodologies, the resulting feedback becomes more interpretable and actionable by trainees. In addition, we discussed a prototype implementation of a dashboard to display such integrated feedback in GIFT. The assessment and feedback generation is implemented as an external assessment engine in GIFT, which both updates competency states in the GIFT learner module, as well as generates visual feedback elements to be displayed on an HTML dashboard presented to the learner as a course component after training completes. The overall goal is to present the learner with not only assessment of their performance, but also understandable and actionable feedback that learners can use to modify their behaviors and strategies to improve performance on the next training iteration, leading to better learning outcomes overall.

While the initial results and development of the feedback dashboard are very promising, there are many areas for future research and development. First, the dashboard requires further development to fully integrate into the EAE for dynamic evaluation. Right now, the evaluated metrics are pre-programmed into the display and are simply populated during scenario playback. In the future, we will have elements of the dashboard dynamically change depending on what metrics and visual feedbacks have been designed, as well as which metrics and feedback are relevant to a given playback. In addition, we would like to extend the visual feedback element to become more interactive for the learner. Right now, the dashboard only displays pre-created images and charts, but in the future, these will be replaced by dynamic elements which allow the user to select and visualize specific areas of interest and further explore the relevant data with additional charts and videos. Next, we are working to integrate some of these same visual feedback elements into GIFT’s Game Master display, which would allow an instructor to see a summary of the feedback being provided to the team alongside their normal performance reports. By closing the loop between the learner feedback and the instructor’s Game Master display, we hope that the feedback we generate can be useful for guiding discussion during after-action reviews. Finally, in the future we will also run additional studies with these new feedback elements integrated. By allowing end users to experience these feedback elements in the field, we will be able to further validate the improved learning outcomes from the expanded feedback. In addition, the users of the system will be able to provide helpful design suggestions, based on what they would like to see, in order to improve the tool in the future. It is our hope that by providing detailed and understandable feedback to learners in these training environments, that the overall learning outcomes will see significant improvement.

ACKNOWLEDGEMENTS

This research and development work has been supported in part by funding from US Army DEVCOM Soldier Center Award #W912CG2220001. The statements and opinions expressed in this paper do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

REFERENCES

Bell, S. T., Brown, S. G., Colaneri, A., & Outland, N. (2018). Team composition and the ABCs of teamwork. *American Psychologist*, *73*(4), 349.

Biswas, G., Rajendran, R., Mohammed, N., Goldberg, B. S., Sottilare, R. A., Brawner, K., & Hoffman, M. (2020). Multilevel learner modeling in training environments for com- plex decision making. IEEE Transactions on Learning Technologies, 13 (1), 172-185. doi: 10.1109/TLT.2019.2923352

Blandford, A., & Furniss, D. (2005, July). DiCoT: a methodology for applying distributed cognition to the design of teamworking systems. In *International workshop on design, specification, and verification of interactive systems* (pp. 26-38). Springer, Berlin, Heidelberg.

Chapman, A. W. (1991). The Army's Training Revolution, 1973-1990: An Overview.

Clark, A. (1997). *Being there* (p. 222). Cambridge, MA: MIT Press.

Galliers, J., Wilson, S., & Fone, J. (2007). A method for determining information flow breakdown in clinical systems. *International journal of medical informatics*, *76*, S113-S121.

Gant, T., Speidel, J., Tatum, D., & Zuelke, E. (2019). Squad Advanced Marksmanship Trainer.

Gegenfurtner, A., Quesada-Pallarès, C., & Knogler, M. (2014). Digital simulation-based training: A meta-analysis. *British Journal of Educational Technology*, *45*(6), 1097-1114.

Goldberg, B., Owens, K., Hellman, K., Robson, R., Blake-Plock, S., Hoffman, M., Gupton, K. (2021). Forging Competency and Proficiency through the Synthetic Training Environment with an Experiential Learning for Readiness Strategy. In *Proceedings of the 2021 I/ITSEC*. Orlando, FL

Hutchins, E. (2000). Distributed cognition. *International Encyclopedia of the Social and Behavioral Sciences. Elsevier Science*, *138*.

Ravert, P. (2002). An integrative review of computer-based simulation in the education process. *CIN: Computers, Informatics, Nursing*, *20*(5), 203-208.

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

Stanton, N. A. (2014). Representing distributed cognition in complex systems: how a submarine returns to periscope depth. *Ergonomics*, *57*(3), 403-418.

Vatral, C., Mohammed, N., Biswas, G., & Goldberg, B. S. (2021). GIFT External Assessment Engine for Analyzing Individual and Team Performance for Dismounted Battle Drills. In *Proceedings of the Ninth Annual GIFT Users Symposium (GIFTsym9)* (p. 107). US Army DEVCOM–Soldier Center.

Vatral, C., Biswas, G., & Goldberg, B. S. (2022). Multimodal Learning Analytics Using Hierarchical Models for Analyzing Team Performance. In *Proceedings of the 15th International Conference on Computer Supported Collaborative Learning* (in press). International Society of the Learning Sciences.

Wright, P. C., Fields, R. E., & Harrison, M. D. (2000). Analyzing human-computer interaction as distributed cognition: the resources model. *Human-Computer Interaction*, *15*(1), 1-41.

Zachary, W. W., Ryder, J. M., & Hicinbothom, J. H. (2000). Building Cognitive Task Analyses and Models of a Decision-making Team in a Complex Real-Time Environment. In Chipman, Shalin, & Schraagen (Eds.), *Cognitive Task Analysis*. Erlbaum.

ABOUT THE AUTHORS

***Caleb Vatral** is a PhD student and research assistant at Vanderbilt University in the Department of Computer Science with a focus in intelligent systems. Working within the Institute for Software Integrated Systems, his research focuses on combining strong theoretical foundations in distributed cognition with multimodal data-driven approaches to support cognitive modeling and system design in human-centered simulation and simulation-based training. Prior to attending Vanderbilt, he received the B.S. degree in computer science and mathematics from Eastern Nazarene College.*

***Naveeduddin Mohammed** is a Senior Research Engineer with the Institute for Software Integrated Systems at Vanderbilt University. Naveed received the M.S. degree in Computer and Information Sciences from University of Colorado and the B.E. degree in Information Technology from Osmania University. He is a full stack developer, and his work focuses on designing, developing, and maintaining frameworks for open-ended computer-based learning environments and metacognitive tutors.*

***Dr. Gautam Biswas** is a Cornelius Vanderbilt Professor of Engineering and Professor of Computer Science and Computer Engineering at Vanderbilt University. He conducts research in Intelligent Systems with primary interests in monitoring, control, and fault adaptivity of complex cyber physical systems, as well as developing intelligent open-ended learning environments that adapt to students' learning performance and behaviors. He has developed innovative learning analytic techniques for studying students' learning behaviors in a variety of simulation and augmented reality-based training environments. He has over 600 refereed publications, and his research is supported by funding from the Army, NASA, and NSF.*

***Dr. Benjamin Goldberg** is a Senior Scientist at the U.S. Army DEVCOM Soldier Center, Simulation and Training Technology Center (STTC) in Orlando, FL. His research in Modeling & Simulation focuses on deliberate competency development, adaptive experiential learning in simulation-based environments, and how to leverage AI tools and methods to create personalized learning experiences. Currently, he is the lead scientist on a research program developing adaptive training solutions in support of the Synthetic Training Environment. Dr. Goldberg is co-creator of the award winning Generalized Intelligent Framework for Tutoring (GIFT) and holds a PhD from the University of Central Florida.*

Addressing Team Process with Automated Speech Act Assessments

Jeremiah T. Folsom-Kovarik[1], Antonio Roque[1], and Anne M. Sinatra[2]
Soar Technology, Inc.[1], US Army Combat Capabilities Development Command (DEVCOM) - Soldier Center[2]

INTRODUCTION

Team process behavior markers are observable actions that can be monitored to indicate the quality of interactions in a group of people who are working together. These markers can be useful in monitoring and guiding the performance of learners in a team.

In simulation-based training, observing and assessing team process behavior markers is often accomplished by instructors or small-unit leaders recording notes during or immediately after scenario execution. However, this consumes mental effort and introduces a training bottleneck since each expert can monitor only a few learners at once. Leaders may also benefit from support to provide effective feedback.

The solution is automating team process assessment to reduce the burden on human experts while producing feedback that is fine-grained, precise, consistent, and objective. Automating assessment of team communication requires computer support for accurate speech capture and processing, natural language processing, and comparing a semantic understanding of the speech to expectations in the context of current scenario events. This paper describes how these capabilities were added to a Generalized Intelligent Framework for Tutoring (GIFT) instantiation, and tested in a training scenario we developed, guided by actual training implementations. The next section provides theoretical background, followed by sections describing our specific approach and implementation details.

BACKGROUND

Team cognitive characteristics and observable behavior markers of team characteristics may be assessed using the U.S. Army's GIFT software, including within immersive simulations such as Virtual Battlespace (VBS). Furthermore, behavior markers may be associated with observations of team performance or team process. Team *performance* describes outcomes or intermediate goals that have defined standards and can be observed under given conditions. On the other hand, team *process* describes the individual and joint cognitive, affective, and psychomotor activities that produce the performance. While focusing solely on team performance can mislead learners into training negative habits or ignoring problems based on a lucky performance, focusing on team process will instead surface and address the underlying adaptive or maladaptive behaviors in a team. As such, team process markers in training are important to measure, assess, and improve via adaptive feedback.

Communication is a major component of team tasks. While there are many challenges with creating and implementing team tutoring, communication is one of the most important and difficult to assess precisely and objectively. Previous GIFT tutors have demonstrated team task tutoring in the form of a two-person surveillance task (Gilbert et al., 2018; Ostrander et al., 2020), a three-person surveillance or sniper task (Ouverson et al., 2021), and a squad-level search and rescue task (McCormack et al., 2019; McCormack et al., 2020). Additionally, earlier work for the project described in this paper developed squad-based VBS activities that include measures of team functional resilience and assessment of team roles (Folsom-Kovarik & Sinatra, 2020; Folsom-Kovarik et al., 2021). However, in most of these instances the content of spoken communication was either not assessed directly, or a human observer was required to make the assessment.

Additional efforts in GIFT have been working toward automated assessment of team-based communication during tutoring. Initial work used an approach of making recordings of learner audio, then creating transcripts based on it, and finally applying rules such as ratios of inclusive vs. exclusive language used (e.g., *We* vs. *I*) to examine ideas such as team cohesion (McCormack et al., 2020). Other recent work has been working on approaches that automate coding of spoken communication such that a human coder would not be required to listen to audio, read transcripts, and code the data as is traditionally done (Min et al., 2021; Spain et al., 2020; Spain et al., 2021).

One of the main challenges to assessing communication in team tutoring is being able to understand, transcribe, and assess the spoken material in real time. Technological challenges exist such as understanding in noisy environments, but there are also task-related challenges such as identifying which individual spoke and whether they communicated the right information to the right person in their team.

APPROACH

Given the background and related work described above, we decided to focus on the following aspects of the problem.

First, we have to decide what types of team cognitive processes to focus on. Among teams, examples of team cognitive processes that enable success in mission-oriented settings include using and updating shared mental models, transactive memory, and complementary or shared situational awareness. Effective team processes to use and update these can be trained and assessed. Shared mental models describe a common understanding about the team roles, responsibilities, goals, and activities to complete the mission at hand. Transactive memory refers to distributing knowledge across team members, knowing who has or needs certain information, and updating or retrieving information whenever needed (Wegner, 1987). Shared situational awareness refers to the team's perception, attention, and memory of changes in the current state of the world, the reasons or meaning of that state, and the projection of trends or changes into the future (Endsley, 1988; Salas et al., 1995). Complementary situational awareness specifies that team members have different awareness that nonetheless mesh across the team into a common operational picture containing all relevant information to perform the mission (Simaan et al., 2015).

Second, we have to identify the appropriate behavior markers to address team processes. Behavior markers that describe how teams communicate during mission execution can be observed and measured in order to assess and train effective team processes. These markers include discussing roles and responsibilities (shared mental models), updating progress toward goals (transactive memory), sharing timely and relevant task information (situational awareness), and communicating in a clear and concise manner. Clear and concise communication is a behavioral antecedent that enables team cognitive processes and good team performance.

To select the correct behavior markers, we are guided by Sottilare and colleagues (2018). We define information exchange markers which are equivalent to the "sharing timely and relevant task information (situational awareness)" marker described above, and to this we ascribe the following markers as described in Sottilare et al., 2018. First, "Occurrences of task relevant information being shared," such as one team member giving mission-related information to another team member. Second, "Occurrences of team members discussing roles and responsibilities," such as when the team divides up tasks, including commands and discussion related to that division. Third, "Occurrences of team members updating one another on progress towards goals," in other words, as the team proceeds, discussion about how much of the mission has been completed, and what remains to be done. We also defined Communication Delivery Markers which are an example of the "clear and concise communication" marker described above. We divide this up into one marker focusing on whether the communication is concise (such as by the length of the typical utterance) or whether the communication is clear (such as in terms of whether or not the other team members respond a readback or other affirmation that communication was understood or with a verbal request indicating that it was not understood).

IMPLEMENTATION DETAILS

Within a Virtual Battlespace (VBS) team training exercise, experimental extensions to automate the above team process assessments are being implemented in GIFT. A pipeline for speech and natural language processing was integrated with experimental GIFT conditions for team assessment. The system diagram shown in Figure 8 displays how current and planned components fit onto the existing infrastructure.

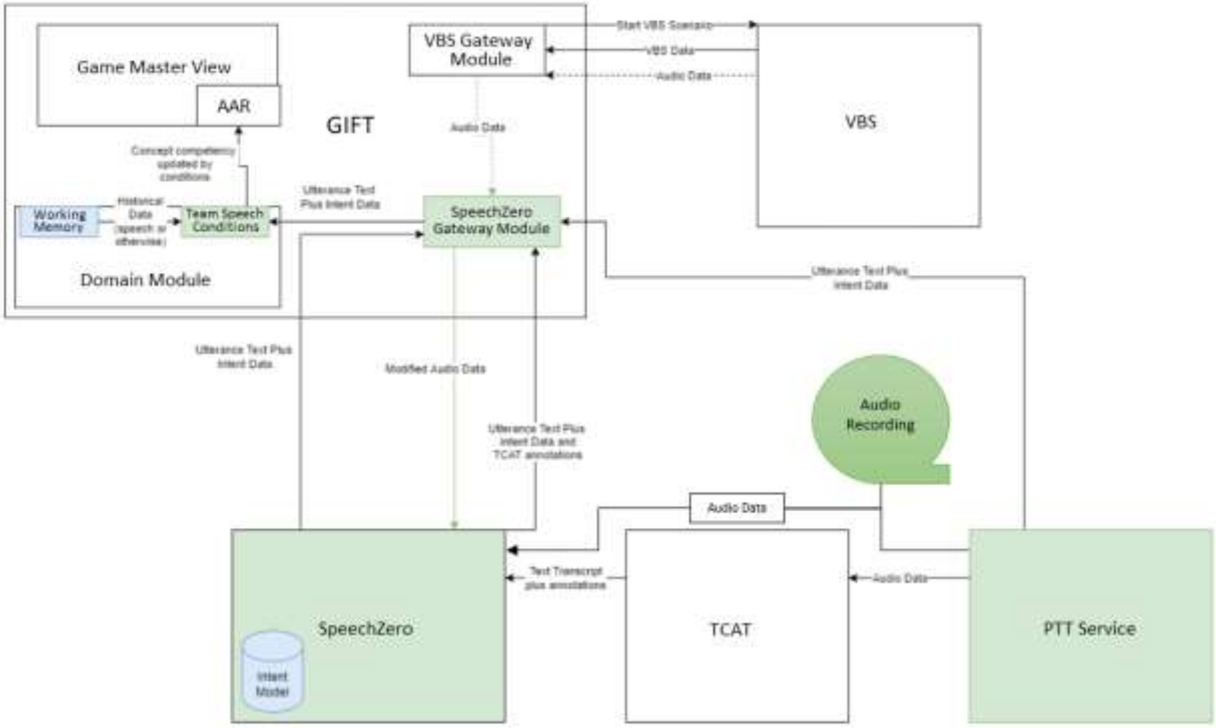

Game Master View
AAR
GIFT
VBS Gateway Module
Start VBS Scenario
VBS Data
Audio Data
VBS
Audio Data
Concept competency updated by conditions
Working Memory
Historical Data (speech or otherwise)
Team Speech Conditions
Utterance Text Plus Intent Data
SpeechZero Gateway Module
Domain Module
Utterance Text Plus Intent Data
Utterance Text Plus Intent Data
Modified Audio Data
Utterance Text Plus Intent Data and TCAT annotations
Audio Recording
Audio Data
Text Transcript plus annotations
Audio Data
SpeechZero
Intent Model
TCAT
PTT Service

Figure 8. System Diagram

First, the push-to-talk (PTT) service captures audio data and sends it to the SpeechZero system (Lebanoff et al., 2021), which performs automated speech recognition (ASR) and intent recognition. We used SpeechZero to manually author language resources reflecting the expected language for this domain, and performed tests to confirm that SpeechZero performed at acceptable rates for this domain. To perform these tests we submitted the audio data from audio recordings directly to SpeechZero through the pathway shown in Figure 8, and we then compared the resulting output to manually transcribed outputs representing ground truth.

The output of SpeechZero is sent along several channels. First, the Intent Data and the recognized text is sent directly to the SpeechZero Gateway Module, which is integrated into GIFT. Second, the data is sent through a secondary channel to the PTT Service and then on to the SpeechZero Gateway Module. Third, TCAT in the figure represents the proposed Team Communication Assessment Tool (Spain et al., 2020). If TCAT annotations are available, those can also be sent to the SpeechZero Gateway Module. The SpeechZero Gateway Module receives this information in a JSON format, so that the SpeechZero module can be replaced by a different ASR/Intent Recognition system if so desired.

The pathway to this point is separate from the VBS system itself, which acts on a different pathway. The VBS Gateway Module is what starts the VBS scenario, receives VBS data, and also receives audio data (such as speech over a simulated radio) which is processed without ASR and Intent Recognition. This input could potentially be routed by the SpeechZero Gateway Module to SpeechZero for processing.

In either case, the Utterance Text plus the Intent Data is then forwarded to the Team Speech Conditions module. Our design is to use this to perform one of several tasks. First, the most versatile GIFT Condition, evaluates the utterance intent for presence of all the “slots” and “values” that are required, i.e., whether the required type of information is given, and that the correct information content is also given. This could be used to detect the Information Exchange Marker related to determining whether task-relevant information has been shared, or that the team members are discussing roles and responsibilities, or that the team members are updating one another on progress toward goals. Second, a GIFT Condition counts the number of times that a particular intent is given. For example, it could be used to determine how often an utterance is not understood, resulting in a “Say again” response to help measure the Communication Delivery Marker associated with clarity. Third, a GIFT condition examines if a given intent is followed by another given intent: for example, the first may be discussing a particular role, and the second may be continuing the discussion about the role. Finally, a “readback” intent Condition checks whether the same information is repeated. These conditions have a Working Memory component available that can be used to maintain information about the types of information experienced so far, thus working as a dialogue history.

As the Team Speech Conditions in the GIFT Domain Module perform their work, it communicates with the AAR module of the Game Master View. This updates immediate and retrospective historical feedback, automating what was previously done manually. Figure 9 shows an example of this visual output. The indicator on the left shows the extent to which a team performed the task of verbally calling out details of a team member’s injuries; this is assessed manually by a trainer as being either substandard (one star), adequate (two stars), or above average (three stars). The indicator on the right shows the extent to which a team performed the task of communicating the presence of a sniper, on the same scale. However, in this case the assessment is performed automatically using the approach described above.

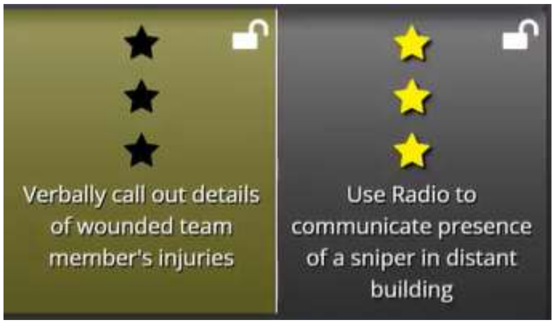

★★★
Verbally call out details of wounded team member's injuries
★★★
Use Radio to communicate presence of a sniper in distant building

Figure 9: Manual vs Automatic Assessments

DISCUSSION

To summarize, at the time of writing, the pipeline and data flows between the PTT capture and GIFT have been implemented in an experimental branch, while a design has been proposed to demonstrate the optimal integration of speech assessments within GIFT structures such as the domain knowledge file (DKF). In the near future, a full implementation on an experimental branch of GIFT will be demonstrated.

The experimental speech processing and natural language processing can be provided to GIFT using a Government off-the-shelf (GOTS) speech stack in military training, research, demonstration, and other Government-purpose use cases. For full compatibility with non-Government use cases, other third-party speech solutions can be substituted. The experimental speech stack provides accurate speech recognition and semantic intent labeling for speech through a domain-specific and task-specific language model. The speech semantic intent describes the meaning of an utterance and enables assessment at a higher level of abstraction compared to relying on specific keyword matches. A contribution of this work is to coordinate the domain specialization of a language model with the training scenario context as defined in the GIFT DKF. The language model can also be authored easily by non-technical personnel, who provide examples of speech which the speech solution then generalizes to recognize similar related speech. A new speech Application Program Interface (API) is proposed for GIFT which provides a standardized interface to send GIFT both the plain text and the semantic intent of captured speech. Therefore, third-party solutions for recognizing speech intent may be substituted and compared to the experimental software.

In order to accomplish team process assessment, experimental GIFT conditions are proposed which associate features of team speech with expectations for expert or inexpert team cognition. Examples of team speech features include the semantic intent, count, frequency, and timing of speech. Specific to assessing clear and concise communication, both computer recognition confidence and detection of team speech acts such as clarification requests contribute to individual and team assessments.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE USE

While several reusable speech features are being measured in the experimental GIFT conditions presented here, an important research question remains as to what thresholds or ranges for each measurement are associated with expert teams or inexpert teams. Thus, the expectations that let GIFT conditions produce assessments are currently set by researchers but could, in future work, be empirically grounded from domain-specific or domain-general observations that support evidence-based scientific understanding of team speech. Once additional speech processing functionality is incorporated into GIFT, authoring tools and interfaces will need to be designed so that non-computer programmers can easily modify condition classes and the assessments to meet their training needs. Moving towards automated assessment of team communication during adaptive training will provide additional opportunities for feedback to learners, and can assist instructors in understanding the performance of their learners in real-time.

REFERENCES

Endsley, M. R. (1988, April). The Functioning and Evaluation of Pilot Situation Awareness. Technical Report. NOR DOC 88-30.

Folsom-Kovarik, J. T., Sieh, J., & Sinatra, A. M. (2021, May). Reasoning about Team Roles and Responsibilities for Team Assessment. In Proceedings of the Ninth Annual GIFT Users Symposium (GIFTsym9) (p. 201). US Army DEVCOM–Soldier Center.

Folsom-Kovarik, J. T., & Sinatra, A. M. (2020, May). Automating assessment and feedback for teamwork to operationalize team functional resilience. In Proceedings of the 8th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym8) (p. 126). US Army Combat Capabilities Development Command–Soldier Center.

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

Lebanoff, L., Newton, C., Hung, V., Atkinson, B., Killilea, J., & Liu, F. (2021, April). Semantic Parsing of Brief and Multi-Intent Natural Language Utterances. In Proceedings of the Second Workshop on Domain Adaptation for NLP (pp. 255-262).

McCormack, R., Case, A., Howard, D., Logue, J., Kay, K., & Sinatra, A. M. (2020, May). Teamwork Training in GIFT: Updates on Measurement and Audio Analysis. In Proceedings of the 8th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym8) (p. 155). US Army Combat Capabilities Development Command–Soldier Center.

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.

Min, W., Spain, R., Saville, J. D., Mott, B., Brawner, K., Johnston, J., & Lester, J. (2021, June). Multidimensional team communication modeling for adaptive team training: A hybrid deep learning and graphical modeling framework. In International Conference on Artificial Intelligence in Education (pp. 293-305). Springer, Cham.

Ouverson, K. M., Ostrander, A. G., Walton, J., Kohl, A., Gilbert, S. B., Dorneich, M. C., ... & Sinatra, A. M. (2021). Analysis of Communication, Team Situational Awareness, and Feedback in a Three-Person Intelligent Team Tutoring System. Frontiers in Psychology, 12.

Ostrander, A., Bonner, D., Walton, J., Slavina, A., Ouverson, K., Kohl, A., ... & Winer, E. (2020). Evaluation of an intelligent team tutoring system for a collaborative two-person problem: Surveillance. Computers in human behavior, 104, 105873.

Salas, E., Prince, C., Baker, D. P., & Shrestha, L. (1995). Situation awareness in team performance: Implications for measurement and training. *Human Factors*, 37(1), 123-136.

Simaan, N., Taylor, R. H., & Choset, H. (2015). Intelligent surgical robots with situational awareness. Mechanical Engineering, 137(09), S3-S6.

Sottilare, R. A., Shawn Burke, C., 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.

Spain, R., Min, W., Saville, J., Brawner, K., Mott, B., & Lester, J. (2021, May). Automated Assessment of Teamwork Competencies using Evidence-Centered Design-Based Natural Language Processing Approach. In Proceedings of the Ninth Annual GIFT Users Symposium (GIFTsym9) (p. 140). US Army DEVCOM–Soldier Center.

Spain, R., Min, W., Saville, J., Mott, B., Brawner, K., Johnston, J., ... & Lester, J. (2020, May). Team Communication Analytics Using Automated Speech Recognition. In Proceedings of the 8th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym8) (p. 145). US Army Combat Capabilities Development Command–Soldier Center.

Wegner, D. M. (1987). Transactive memory: A contemporary analysis of the group mind. In *Theories of group behavior* (pp. 185-208). Springer, New York, NY.

ACKNOWLEDGEMENTS

This material is based upon work supported by the U.S. Army Combat Capabilities Development Command Soldier Center (DEVCOM 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 DEVCOM-SC STTC or the U.S. Government.

We are grateful to Nick Giranda for providing implementation details.

ABOUT THE AUTHORS

Dr. Jeremiah 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. Antonio Roque *is a Research Scientist with Soar Technology, Inc. His research focuses on natural language processing, user modeling, and natural language dialogue.*

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

Leveraging Advances in Natural Language Processing to Support Team Communication Analytics in GIFT

Randall D. Spain[1], Wookhee Min[1], Jason D. Saville[1], Andrew Emerson[2], Jay Pande[1], Keith Brawner[3], and James C. Lester[1]
North Carolina State University[1], Educational Testing Service[2], U.S. Army Combat Capabilities Development Command - Soldier Center[3]

INTRODUCTION

Developing tools that can facilitate team communication analytics is critical for supporting the Army's goal of providing squads with automated coaching and feedback during collective training exercises in synthetic training environments. Team communication analytics can provide insights regarding how team members collaborate, coordinate, and distribute information to support effective team performance (Marlow et al., 2018; Sottilare et al., 2018). Traditional methods for analyzing team communication patterns and behaviors can be resource intensive, requiring trained experts to transcribe, review, and code transcripts of team communication. This process of manually converting spoken communication into coded transcripts does not support real-time analysis of team communication that is required for adaptive instructional systems (AISs) for teams. Emerging deep learning-based natural language processing (NLP) techniques have shown promise for addressing these challenges and are being utilized in a number of domains to analyze team interaction data, including speech and text-based communication (Carpenter et al., 2020; Lucini et al., 2021; Pugh et al., 2021).

To meet the need for a robust team communication analysis capability, the U.S. Army Combat Capabilities Development Command - Soldier Center, Simulation and Training Technology Center, and North Carolina State University have been collaboratively designing, developing, and iteratively refining an NLP-based team communication analysis framework that aims to automatically analyze team communication data, parse it into classification schemes, and provide summary statistics of critical team communication features that can be used to examine and identify antecedents of team performance. The goal of this framework is to integrate NLP-based analysis capabilities into the Generalized Intelligent Framework for Tutoring (GIFT) to support real-time analysis of squad communication and to provide team communication analytics that can be used to drive adaptive coaching and scaffolding.

In this paper, we describe our work towards designing and developing this framework, called the Team Communication Analysis Toolkit (TCAT), and our plans for integrating it into GIFT. The set of core functionalities currently implemented in TCAT include: (1) automatic speech recognition using the MS Azure Speech-to-Text cloud service which takes as input an mp3 or wav-format audio file and produces a machine-generated transcript of squad members' spoken communication; (2) NLP-driven dialogue analysis that automatically classifies each utterance based on a predefined labeling scheme to indicate the intent of the speech (e.g., pass information, request information, command, acknowledge) and how the information was shared among team members (e.g., request from team leader, request from squad leader), (3) a data analysis interface that includes descriptive statistics summarizing dialogue labels and other variables associated with the team communication data, (4) a data management interface that allows users to edit, revise, and re-label team communication transcripts, and (5) a data visualization interface which shows communication patterns using a set of visualization techniques. TCAT also allows for importing and exporting Excel-formatted transcripts, as well as selecting or loading pre-trained NLP models that can be used to predict dialogue labels. We describe these functionalities in detail, as well as the features the research team has implemented following two rounds of focus groups conducted with team science researchers and adaptive instructional systems (AISs) developers. Then, we describe challenges and areas of future research for TCAT, including how to deal with a large degree of noise in the Automatic Speech Recognition (ASR) generated transcripts (e.g., high word error rates), generalizability issues with the trained NLP models when applied to different target training domains whose data distributions are different from the source domain, and the challenges of designing reliable predictive models, given a limited training dataset. The paper concludes with a discussion of promising future directions for TCAT and steps towards integrating the framework within GIFT to support the automated assessment of team communication during synthetic training events.

BACKGROUND

Investigating team members' spoken communication can provide significant insight into the processes that contribute effective team performance (Marlow et al., 2018). As previously noted, while traditional methods for analyzing team communication such as content analysis can provide detailed information about the types of information that are shared among team members and information exchange sequences, the process of converting spoken communication to transcripts and coded content has largely remained a labor-intensive, human-driven task which does not support the near real-time analysis of team communication that is required for AISs for teams.

Advances in deep learning-based NLP have shown significant promise for automating transcript generation and performing syntactic, semantic, and dialogue analyses based on transcribed data (Deng & Liu, 2018). Deep learning-based NLP techniques learn a multi-level, hierarchical set of features from lower-level textual data through multiple layers included in neural networks. This key advantage of deep learning reduces the need for feature engineering by human experts that is often expensive in terms of time and effort since the entire network is end-to-end trainable.

TCAT's Natural Language Pipeline

TCAT leverages advances in deep learning-driven NLP to provide a NLP pipeline and end-user interface to support end-to-end natural language analysis of team members' spoken dialogue (Figure 1). TCAT uses the Microsoft Azure Speech-to-Text cloud service to take raw speech audio and produce machine-generated transcripts of team members' spoken communication. To implement core NLP functionalities of TCAT, we developed an NLP pipeline, which sequentially performs a series of computational tasks required to understand and analyze team dialogue and deliver meaningful statistics for users to support team communication analytics. Additionally, TCAT includes a data management interface to support the investigation of team communication data, and a data visualization interface which shows communication patterns using a set of visualization techniques.

Automatic Speech Recognition (ASR)

The first component in the NLP pipeline supporting TCAT is ASR, which converts spoken team communication into text for the pipeline's dialogue act predictions. TCAT currently uses Microsoft Azure's Speech-to-Text cloud service to generate text transcripts on imported audio files of team members' communication exchanges. Utilizing TCAT's interface, users can select and import an mp3 or wav format audio file. TCAT will process this audio input file and generate an output file with speech recognition results that is saved in an Excel-formatted file. Speaker labels can also be added to the output of the ASR generated transcripts.

Our team's decision to use Microsoft Azure as the primary ASR service over other popular off-the shelf and open-source ASR engines was based on previous research showing that speech recognition software accuracy can vary in different environmental settings (e.g., Georgila, 2020), as well as results of our own testing which aimed to identify which contemporary ASR systems were most suited for analyzing multi-party speech data from collective training events with differing levels of environmental noise (Spain et al., 2020). Specifically, we analyzed the performance of three ASR systems: Google Cloud Speech-to-Text, Microsoft Azure Speech-to-Text, and Kaldi using transcripts from a live training mission and an after-action review (AAR) session. Results showed that the Microsoft Azure speech recognition system outperformed Google when transcribing speech from a live training event. In the evaluation, Kaldi was not effective in generating reliable transcripts for the live training session (Spain et al., 2020).

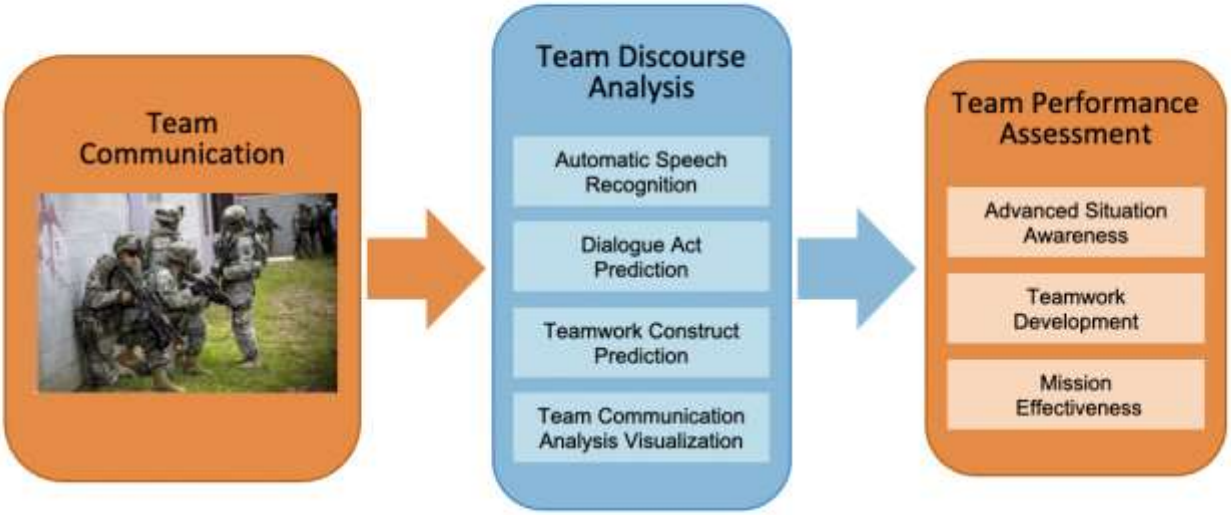

Team Communication
Team Discourse Analysis
Automatic Speech Recognition
Dialogue Act Prediction
Teamwork Construct Prediction
Team Communication Analysis Visualization
Team Performance Assessment
Advanced Situation Awareness
Teamwork Development
Mission Effectiveness

Figure 1. TCAT Architecture

Dialogue Act Prediction Model and Deep Learning Framework

The second component in the NLP pipeline supporting TCAT is the dialogue act prediction model, which analyzes and classifies team members' spoken words into common themes inherent in the spoken communication data. Recently, deep neural networks have demonstrated significant potential for analyzing team/group discourse (Carpenter et al., 2020; Park et al., 2021), leveraging deep learning's distributed representation learning capabilities (e.g., Peters et al., 2018) and high predictive performance in an end-to-end fashion (Young et al., 2018), as well as for enhanced flexibility in model design and training, such as multi-task learning and transfer learning (Torrey & Shavlik, 2010).

TCAT currently uses a hybrid framework that combines conditional random fields' structured prediction with deep neural networks' contextual language representation learning capabilities to classify team communication utterances into 1 of 9 speech acts, which reflect the basic purpose of the utterances (Min et al., 2021). The speech act labels include *acknowledgement, action request, action statement, attention, command, greeting, provide information, request information,* and *other*. These labels are based upon coding schemes established by team science researchers to examine team processes and teamwork behaviors in military settings.

TCAT's NLP-driven hybrid dialogue act prediction model was trained utilizing coded transcripts from Squad Overmatch's (SOvM) Mission 3 (M3) dataset. This dataset includes 6,181 utterances captured from six squads that each completed a 45-minute live training exercise that required them to conduct a zone reconnaissance in a local village, conduct key leader engagements, exploit intelligence, confirm location of a suspected arms cache, and exploit the site, if able. During the training exercise, squad members interacted with role players who needed assistance and coordinated actions while responding to a simulated explosion, exchanging gunfire with hostile faction leaders, and performing tactical combat casualty care tasks. Each transcribed utterance was coded using a framework of 27 speech act labels and 18 team dimension labels when applicable (David Traum, personal communication, June 4, 2020). The corpus also includes additional coding to represent the associated event in the training scenario, the speaker, the time stamp of the utterances, and who is involved in the communication exchange. Our research team applied a mapping scheme to reduce the number of speech acts from 27 to 9 labels to improve the predictive accuracy of the trained model.

Prediction accuracy evaluations showed the hybrid model (1) outperformed both multi-task and single-task variants of stacked bidirectional long short-term memory networks using the same distributed representations of the utterances, (2) outperformed a hybrid approach that uses non-contextual utterance representations for the dialogue classification tasks, and (3) was able to classify squad members' speech acts with 69% accuracy (using 9 different labels) and team dimension labels with 64% accuracy (using 19 different labels; Min et al., 2021). The model also showed positive domain transfer capabilities (67% accuracy) for a new squad communication dataset from the Squad Overmatch Project.

One of the goals of TCAT is to facilitate generalizable team communication discourse tagging models that can assess teamwork skills, such as communication, cooperation, and coordination, across new teams and tasks. To support a generalizable and open framework, TCAT allows users to import and utilize their own NLP-classifier models trained with other datasets. If a user would like to expand the set of speech acts used in the hybrid model or create a completely new classification model, this can be achieved by loading new models into TCAT using the model import feature. The prediction results are incorporated within TCAT's data frame, and the predicted set of dialogue tags can be visualized within TCAT's data visualization interface and/or saved into a .csv file.

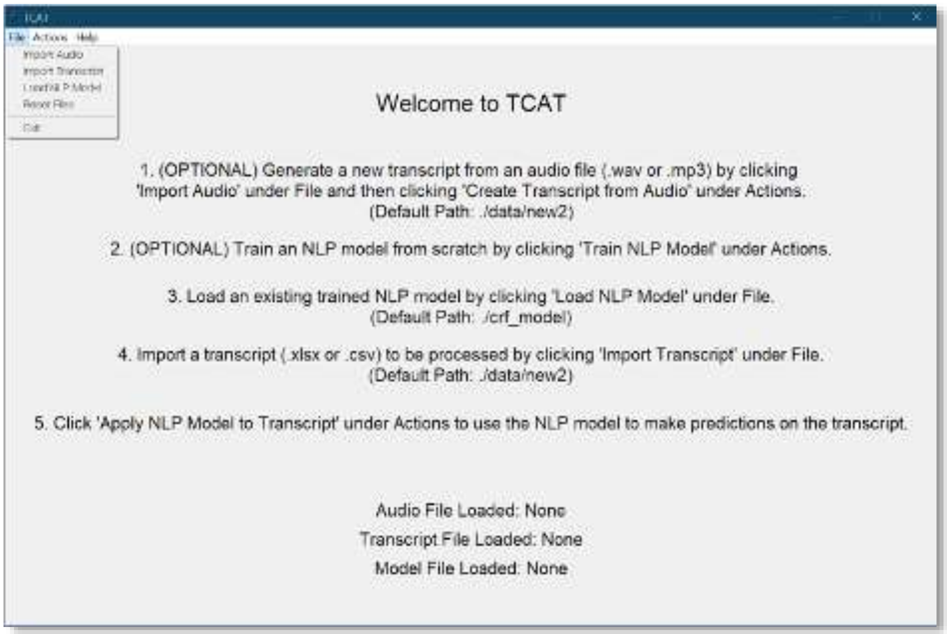
TCAT Interface and User Workflow

Main Menu

TCAT includes user-friendly menu options on its initial screen. The 'File' menu allows users to import audio files, transcript files, and NLP models, as well as de-selecting those files (i.e., 'Reset Files'). The 'Actions' menu allows users to process the audio file into a transcript and to apply a NLP model to the loaded transcript file (i.e., predicting speech acts of utterances in the transcripts). TCAT also allows users to import a team communication transcript file (.xlsx file) and export user-annotated data into a .csv file with the same column format as the input file. Figure 2 shows the current, menu-based interface for selecting and importing files with step-by-step instructions for how to import audio files, transcript files, and NLP models through the menu options. The interface also instructs the user on how to process these files for communication analysis.

Once users have a transcript and an NLP model they would like to utilize, they can: (1) Load an existing trained NLP model by clicking "Load NLP Model" under the File menu and selecting it; (2) Import a transcript to be processed by clicking "Import Transcript" under the File menu, (3) Apply the NLP model to the transcript to classify utterance predictions by selecting "Apply NLP Model to Transcript" under the Actions menu.

TCAT's interface indicates whether an audio file has been loaded, a transcript has been loaded, and whether an NLP model has been selected.



Welcome to TCAT

1. (OPTIONAL) Generate a new transcript from an audio file (.wav or .mp3) by clicking 'Import Audio' under File and then clicking 'Create Transcript from Audio' under Actions.
(Default Path: ./data/new2)

2. (OPTIONAL) Train an NLP model from scratch by clicking 'Train NLP Model' under Actions.

3. Load an existing trained NLP model by clicking 'Load NLP Model' under File.
(Default Path: ./crf_model)

4. Import a transcript (.xlsx or .csv) to be processed by clicking 'Import Transcript' under File.
(Default Path: ./data/new2)

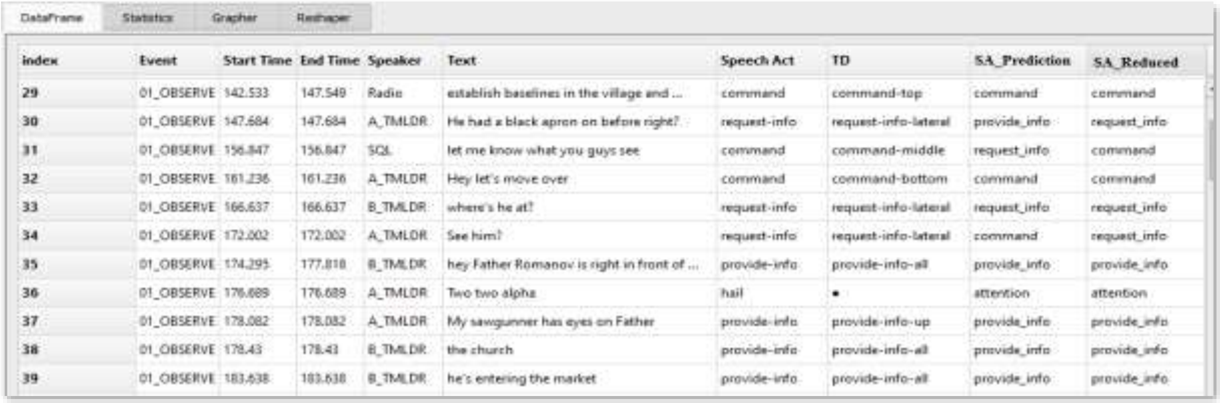
5. Click 'Apply NLP Model to Transcript' under Actions to use the NLP model to make predictions on the transcript.

Audio File Loaded: None
Transcript File Loaded: None
Model File Loaded: None

Figure 2. The Menu-based Design of the TCAT User Interface

Data Management Interface

TCAT's data management interface allows users to view ASR- or human-generated transcripts and manage team communication datasets within the toolkit. Users can make edits to the transcript within the toolkit itself, and these edits are captured and locally saved. The interface also displays the output of NLP model-generated predictions of dialogue acts. Figure 3 depicts the data management interface showing a sample transcript with a reduced set of 9 speech act labels ("SA_PREDICTION") predicted by a pre-trained model. If the transcript contains speech act labels produced by an expert human rater, then the "SA_REDUCED" column shows the expert-provided label for each utterance.

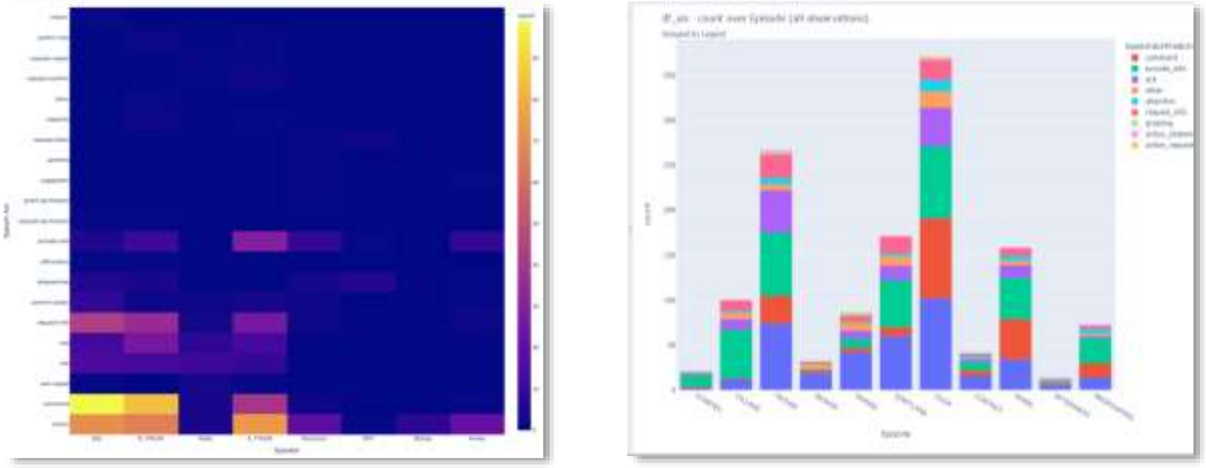


index	Event	Start Time	End Time	Speaker	Text	Speech Act	TD	SA_Prediction	SA_Reduced
29	01_OBSERVE	142.533	147.549	Radio	establish baselines in the village and ...	command	command-top	command	command
30	01_OBSERVE	147.684	147.684	A_TMLDR	He had a black apron on before right?	request-info	request-info-lateral	provide_info	request_info
31	01_OBSERVE	156.847	156.847	SQL	let me know what you guys see	command	command-middle	request_info	command
32	01_OBSERVE	161.236	161.236	A_TMLDR	Hey let's move over	command	command-bottom	command	command
33	01_OBSERVE	166.637	166.637	B_TMLDR	where's he at?	request-info	request-info-lateral	request_info	request_info
34	01_OBSERVE	172.002	172.002	A_TMLDR	See him?	request-info	request-info-lateral	command	request_info
35	01_OBSERVE	174.295	177.818	B_TMLDR	hey Father Romanov is right in front of ...	provide-info	provide-info-all	provide_info	provide_info
36	01_OBSERVE	176.689	176.689	A_TMLDR	Two two alpha	hail	•	attention	attention
37	01_OBSERVE	178.082	178.082	A_TMLDR	My sawgunner has eyes on Father	provide-info	provide-info-up	provide_info	provide_info
38	01_OBSERVE	178.43	178.43	B_TMLDR	the church	provide-info	provide-info-all	provide_info	provide_info
39	01_OBSERVE	183.638	183.638	B_TMLDR	he's entering the market	provide-info	provide-info-all	provide_info	provide_info

Figure 3. The TCAT Data Management Interface with Sample Transcript

Data Visualization Interface

TCAT includes an interactive discourse visualization interface that allows users to investigate team communication statistics (e.g., frequency of speech act, team development labels) that appear in the transcript. Users can view summary statistics that are directed from the dataset. TCAT also includes features that facilitate data visualization in the form of histograms, heatmaps, word clouds (e.g., frequent words that appear in the speech), and more. The data visualization interface also facilitates simple crosstab analysis which allows users to investigate frequencies of speech acts across speakers, time, and other labeled variables in team communication transcripts. For example, Figure 4a shows a heatmap in which the x-axis represents individual speakers, and the y-axis represents individual speech acts, with color indicating the frequency of each speech act per speaker (yellow: more frequent, purple: less frequent). Additionally, Figure 4b shows an example histogram examining the frequency of different speech act labels across each scenario event. Scenario events can be found on the x-axis, different colors indicate different speech acts, and the y-axis represents speech act frequency.



(a) (b)

Figure 4. Example data visualizations within the TCAT data visualization interface

LESSONS LEARNED AND DEVELOPMENT OPPORTUNITIES

Developing a generalizable end-to-end NLP pipeline that performs real-time natural language analysis on team members' spoken dialogue would significantly advance team assessment capabilities to support Army training. There is growing evidence that advances in NLP show significant promise for meeting this goal. During the course of developing and iteratively refining TCAT, we have encountered several expected and unexpected challenges. We discuss these challenges and opportunities for future research and development below.

ASR Performance

Although ASR technology is advancing steadily, a number of barriers to accurate transcription remain. When there are multiple simultaneous speakers, each speaker's utterances interfere with accurate recognition of other speakers' utterances (Yousefi & Hansen, 2020). ASR for TCAT needs to be able to effectively differentiate between speakers. Differences in the characteristics of speakers' voices, such as pitch and breathiness, can also impact ASR accuracy (Sokolov & Savchenko, 2021).

To combat these challenges, various ASR systems and the custom models they provide should be evaluated. Custom models can be used to help accurately recognize many domain-specific phrases, recognize terms with a nonstandard pronunciation, and achieve better accuracy when atypical speaking styles, accents, or background noises are present. Some ASR systems also allow for augmentation with specific phrases, and even allow the user to specify predefined classes of phrases to be recognized in the data. Additionally, many different pre-trained models are available for certain types of data, such as models for phone calls, video, or short utterances. The project team will also investigate approaches to properly handle partially accurate transcripts generated by ASR for team communication data captured in a noisy environment. Particularly, it will be important to explore pre-trained deep neural language models trained with a large corpus of noisy language data and evaluate their transfer learning capabilities for analyzing team communication data from live and synthetic training exercises.

Developing Accurate Prediction Models with Small Datasets

Another challenge we have faced is that machine-learned models, particularly deep neural networks, generally produce more robust results when they are trained with large data sets. The training data set we have used to develop and test TCAT's dialogue act prediction models come from the SOvM project which includes transcriptions of 6 squads performing two separate live training missions. While this dataset presents a very realistic set of team communication exchanges and behaviors, its size is very limited. Examining whether pre-training TCAT's dialogue classification models using other large natural language datasets and fine-tuning them using squad-level dialogue data improves dialogue act prediction accuracy is a promising direction for future research. Using this transfer learning approach, deep neural networks can initially learn a common set of linguistic features leveraging various sources of natural language data, and further update the learned features through an optimization process using the given communication dataset.

Using machine-learned models to predict team performance outcomes with sparse datasets presents a similar set of challenges, especially given that team performance assessment requires a multi-method, multi-dimensional strategy to ensure reliable measurement. We have investigated how data-driven approaches, that are not as data intensive as machine learning based methods can be used to predict team performance from linguistic features extracted from team communication transcripts. Our preliminary results suggest that analyzing linguistic features may produce accurate predictions of certain team performance outcomes, such as ratings of information exchange and advanced situation awareness (Spain et al., 2021).

As we continue to develop and refine TCAT's prediction models, our team plans to explore how top-down (i.e., domain expert-created rules) and bottom-up analytic (i.e., data-driven) approaches can supplement each other by infusing human-generated evidence-of-performance rules in the NLP-based approach. While evidence "rule" generation is a labor-intensive process, there is opportunity for domain experts such as observer coaches/trainers to develop rules alongside performance criteria before training events are conducted. When combined with the data-driven approach, evidence rules can be highly predictive of team performance and readily transferable across different training missions, especially in situations in which ideal performance can be readily pre-defined, such as for live training events.

Improving the Generalizability of NLP-based Dialogue Prediction Models

While predicting team performance outcomes from spoken team dialogue is critical for supporting adaptive team training in AISs, creating accurate squad communication NLP models that can generalize to different training domains and tasks presents its own unique set of challenges. Our hybrid, multidimensional team communication analysis framework, which utilizes a probabilistic graphical model with a deep learning-based contextual language representation model, has demonstrated encouraging near-domain transfer capabilities when we examined how well the trained model performed on a different training mission from the Squad Overmatch project (Min et al., 2021). Building on findings from predictive models' near-domain transfer capabilities, it will be important to investigate domain adaptation techniques to effectively handle different data distributions. In addition, it will be important to investigate intermediate representations (e.g., latent encoding generated by autoencoders) that can deal with a set of input features that could be generated in target domains, which are different from features for the source domains, to support far transfer of the NLP models, including model transfer to other proximal training domains such as squad-level battle drills.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

As our project moves forward, we will continue to work with the GIFT user community to determine how TCAT can be refined to support team communication analysis and team performance assessment in GIFT. There are several promising directions for future research. One direction is determining how TCAT can be integrated with GIFT to perform end-to-end natural language analysis on team members' spoken communication. In this capacity, TCAT could serve as an external assessment engine to support real-time assessment of course concepts in GIFT and provide summative assessments services.

A second direction for future research is expanding TCAT's team discourse analytic capabilities to perform real-time team discourse analysis. TCAT currently performs ASR transcript generation and dialogue act prediction offline. There are a number of technical challenges that need to be addressed in creating a real-time communication analysis pipeline, including:

- Optimizing parameters in ASR services and deep neural network-based language models, to accurately process team speech data in real-time,

- Investigating data synchronization methods and determining how best to segment utterances for downstream NLP processing tasks, such as dialogue act prediction, and

- Comparing network architectures and examining network bandwidth for optimally handling the spoken data, Artificial Intelligence (AI) driven NLP pipeline, and feedback delivery for each team member.

In our future work, we plan to explore TCAT's real-time dialogue prediction feature by investigating capabilities of cloud-based ASR services for real-time audio processing and implementing a streamlined NLP approach that will perform dialogue act prediction in predefined time-based segments (e.g., 3-minute windows).

A third direction for future research is to expand TCAT's dialogue classification model. TCAT currently includes speech act dialogue predictions only. Expanding TCAT to include a team dimension dialogue prediction model is a natural next step and our previous work in this area shows promising results (Min et al. 2021). A fourth direction, which has been discussed in the previous section, is to explore additional methods for improving TCAT's ASR accuracy. A fifth recommendation is to continue to refine TCAT's NLP architecture to improve dialogue classification performance, which can be bolstered with large, labeled datasets for prediction model training. This line of research can be greatly enhanced by gaining access to large, labeled datasets that can be used to train prediction models. Finally, it will be important to refine TCAT's data visualization and user interface to ensure it contains usable features to support team communication analytics.

ACKNOWLEDGMENTS

The research described herein has been sponsored by DEVCOM Soldier Center / 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

Carpenter, D., Geden, M., Rowe, J., Azevedo, R., & Lester, J. (2020, July). Automated analysis of middle school students' written reflections during game-based learning. In I. Bittencourt, M. Cukurova, K. Muldner, R. Luckin, & E. Millán (Eds.). *Artificial intelligence in education. AIED 2020. Lecture notes in computer science*, *12163* (pp. 67–78). Springer, Cham.

Deng, L., & Liu, Y. (2018). A joint introduction to natural language processing and to deep learning. In L. Deng & Y. Liu (Eds.). *Deep learning in natural language processing* (pp. 1–22). Springer, Singapore.

Georgila, K., Leuski, A., Yanov, V. & Traum, D. (2020). Evaluation of off-the-shelf speech recognizers across diverse dialogue domains. In *Proceedings of the 12th language resources and evaluation conference (pp. 6469–6476)*. European Language Resources Association, France.

Lucini, F. R., Krewulak, K. D., Fiest, K. M., Bagshaw, S. M., Zuege, D. J., Lee, J., & Stelfox, H. T. (2021). Natural language processing to measure the frequency and mode of communication between healthcare professionals and family members of critically ill patients. *Journal of the American Medical Informatics Association, 28*(3), 541–548.

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.

Min, W., Spain, R., Saville, J. D., Mott, B., Brawner, K., Johnston, J., & Lester, J. (2021, June). Multidimensional team communication modeling for adaptive team training: A hybrid deep learning and graphical modeling framework. In I. Roll, D. McNamara, S. Sosnovsky, R. Luckin, & V. Dimitrova (Eds.). *Artificial intelligence in education. AIED 2021. Lecture notes in computer science, 12748* (pp. 293–305). Springer, Cham.

Park, K., Sohn, H., Mott, B., Min, W., Saleh, A., Glazewski, K., Hmelo-Silver, C., and Lester J. (2021). Detecting disruptive talk in student chat-based discussion within collaborative game-based learning environments. *Proceedings of the Eleventh International Learning Analytics and Knowledge Conference*, 405–415.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L. (2018). Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2227–2237.

Pugh, S. L., Subburaj, S. K., Rao, A. R., Stewart, A. E., Andrews-Todd, J., & D'Mello, S. K. (2021). Say what? Automatic modeling of collaborative problem-solving skills from student speech in the wild. In I. Hsiao, S. Sahebi, F. Bouchet, & J. Vie (Eds.). *Proceedings of the 14th international conference on educational data mining (EDM21)* (pp. 55–67).

Sokolov, A., & Savchenko, A. V. (2021). Gender domain adaptation for automatic speech recognition. In *Proceedings of the 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics* (pp. 413–418).

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.

Spain, R., Min, W., Saville, J., Brawner, K., Mott, B., & Lester, J. (2021, May). Automated assessment of teamwork competencies using an evidence-centered design-based natural language processing approach. In B. S.

Goldberg (Ed.). *Proceedings of the ninth annual GIFT users symposium (GIFTSym9)* (pp. 140–149). US Army Combat Capabilities Development Command–Soldier Center.

Spain, R., Min, W., Saville, J., Mott, B., Brawner, K., Johnston, J., Goodwin, G., & Lester, J. (2020, May). Team communication analytics using automated speech recognition. In B.S. Goldberg (Ed.), *Proceedings of the eighth annual generalized intelligent framework for tutoring (GIFT) users symposium (GIFTSym8)* (pp. 145–154). US Army Combat Capabilities Development Command–Soldier Center.

Torrey, L., & Shavlik, J. (2010). Transfer learning. In E. Olivas, J. Guerrero, M. Martinez-Sober, J. Magdalena-Benedito, & A. Serrano López (Eds.). *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* (pp. 242–264). IGI Global.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine, 13*(3), 55–75.

Yousefi, M., & Hansen, J. H. L. (2020) Block-based high performance CNN architectures for frame-level overlapping speech detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29*, 28–40.

ABOUT THE AUTHORS

Dr. Randall Spain *is a Research Scientist in the Center for Educational Informatics at North Carolina State University. He holds a PhD in Human Factors Psychology from Old Dominion University. His research focuses on the design and evaluation of advanced training technologies on learning and performance.*

Dr. Wookhee Min *is a Research Scientist in the Center for Educational Informatics at North Carolina State University. He holds a PhD in Computer Science from North Carolina State University. 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.*

Jason Saville *is a PhD candidate of industrial and organizational psychology at North Carolina State University and Graduate Research Assistant in the Center for Educational Informatics. He received his B.A. degree in Psychology from North Carolina State in 2018 and 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. Andrew Emerson *is a Machine Learning Engineer at Educational Testing Service where he works on the Multimodal AI team. He received his Ph.D. and M.S. degrees in Computer Science from North Carolina State University, and his B.S. in Mathematics and Computer Science from Furman University. His research interests span predictive student modeling, multimodal learning analytics, and educational applications of machine learning.*

Jay Pande *is a PhD student in Computer Science at North Carolina State University. He received a BS in Computer Science from Duke University in 2020. His research interests lie in the use of speech data to improve educational outcomes for users of computed-based learning environments.*

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 (DEVCOM-SC-STTC) and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT). He holds a master's and PhD degree in Computer Engineering with a focus on Intelligent Systems and Machine Learning from the University of Central Florida.*

Dr. James Lester *is a 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. 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 Assessment Methods for Training Command Staff Decision Making

Randy Jensen[1], Evan Finnigan[1], Grace Teo[2], and Gregory Goodwin[3]
Stottler Henke Associates, Inc.[1], Quantum Improvements Consulting, LLC[2],
U.S. Army Combat Capabilities Development Command (DEVCOM) - Soldier Center[3]

INTRODUCTION

This paper shares an updated view of assessment and feedback methods under development for team training in an applied Army command-level decision making domain. The overarching objective is to work toward generalizable practices for assessing team performance and team processes in command-level training domains where team effectiveness centers on the dynamics of teamwork and associated cognitive skills. The initial example training application focus for this effort is on wargaming, a critical stage in the Army's military decision making process (MDMP). Wargaming is a collective activity where command staff perform course of action (COA) analysis by stepping through major events and considering actions, reactions, and counteractions, with representatives of multiple warfighting functional areas contributing perspectives critical to the synchronization of plans. A prototype is currently under development for this application, designed as a distributed team trainer. This is intended to create an environment where a team of participants can join from remote locations and step through a facilitated wargaming exercise with opportunities to exhibit teamwork behaviors in the process of collaboratively walking through a COA scenario. A previous GIFTSym presentation (Jensen et al., 2021) described the preliminary structural design of the distributed trainer and interactions across modules and GIFT (Generalized Intelligent Framework for Tutoring, Sottilare et al., 2012). The focus for this paper is on the design of team assessment methods for command staff wargaming in the prototype. Teo et al. (2021) presented a model for teamwork constructs applicable to wargaming performance assessment, which provides the theoretical underpinning for team dimensions to be utilized for assessment in the prototype application. The model is adapted from common constructs appearing in the literature (Johnston et al., 1998; Kozlowski et al., 2015; Marlow et al., 2018; Salas et al., 2005; Sottilare et al., 2018), and tailored to command-level teamwork in wargaming. Teamwork dimensions in the model include Leadership, Team Cognition, Information Exchange, Communication Quality, Supporting Behaviors, and Team Orientation.

This paper walks through practical design considerations for constructing assessment rules that are organized around the team dimensions in the model, and that can be applied in wargaming exercises under development in the current effort. A fundamental goal for this effort in a technology sense is to construct automated assessment capabilities that can facilitate instructors as much as possible in identifying markers of teamwork exhibited in a wargaming exercise. And yet this goal is also moderated by the need to include mechanisms for human observers and/or team participants themselves to contribute assessment inputs when feasible, to augment those generated automatically. Since there are channels of team interaction that the system is unable to parse (such as communications outside of the training environment), it is important to provide a means for assessments based on human understanding, when available. In complex team decision making domains, assessment results are ideally echoed to the training audience in a manner that stimulates a guided process of self-correction within the team. Thus, markers intended for presentation in debriefing come from two sources: automated assessments and manual inputs by human observers or participants. Both are tagged with relevant team dimensions, and information about noted positive or negative teamwork behaviors. Tagging facilitates post-processing to distill the aggregated set of markers for practical use in team reflection. This paper gives examples of assessment rules in the wargaming application, and preliminary design thoughts for how assessment logic will be encoded in the GIFT Domain Module.

Although this is work in progress, the intention is to share design and development approaches that may inform related efforts.

TEAM TRAINING APPLICATION

The team training application discussed throughout this paper is under development in an effort called Reusable Automated Assessment and Feedback for Teams (RAAFT), which is being conducted for the U.S. Army Combat Capabilities Development Command Soldier Center. With the focus of the RAAFT distributed training prototype on MDMP and wargaming, the development team observed instructor-led training at the Command and General Staff College (CGSC) at Fort Leavenworth, where students have the opportunity to practice wargaming in fully-developed scenarios. The curriculum starts with classroom training to introduce concepts before ultimately moving to culminating exercises with cohorts of students performing in command staff roles throughout the MDMP process. The initial prototype is designed for preliminary training, to work on the dynamics of teamwork in wargaming scenarios, while operating under a number of constraints to simplify exercises. As a browser-based application, it supports distributed training with participants in remote locations, but this is not a requirement and the trainees can also be co-located. Some elements of the training experience that are simplified for this instructional context include:

- Participants. The initial training application is structured for 6 students assigned to key warfighting functions in each exercise. In contrast, a full command staff performing COA analysis represents a wider range of warfighting functional areas, and typically involves greater numbers of participants.

- Observer / instructor roles. In addition to student participants, the training environment allocates roles for observers / instructors, providing the opportunity to monitor exercise events, and tools to add markers for teamwork (complementary to automated assessment mechanisms which also generate teamwork markers).

- Structured decision making. The COA analysis process is approximated in the training application with a structure involving predefined, scenario-oriented prompts. In real-world wargaming, MDMP practices provide a certain amount of structure for evaluating each phase of the COA, with open-ended analysis of the events involved, enemy reactions, possible counteractions, intelligence requirements, associated time and distance calculations, projected adjudication of engagements, and so on. However, in order to make it less open-ended to work effectively in a browser-based setting, the training environment uses structured prompts to guide participants through a constrained sequence of decisions. These prompts aim to be analogous to freeform wargaming decisions, with similar opportunities for teamwork.

- Overt teamwork actions. Input mechanisms are integrated into the trainer design to make it easy for participants to convey information or coordinate with other team members during an exercise. Examples include widgets to send pre-formatted messages, to facilitate not only the act of passing information, but also the training system’s ability to track what’s being conveyed to whom. Another example is a set of widgets used when reviewing decisions, to mark when team members agree with decisions or wish to discuss further. The initial training application does not attempt to process the content of freeform chat messages, or any of the natural communications that may happen between co-located participants.

- Decision engagement. Partly because of the distributed setting, one design concern was to keep participants engaged. So the exercise flow is designed to include all participants in all prompts, at least for collecting initial responses. In real-world wargaming processes conducted in person, discussion of a particular topic or decision may be limited to the representatives of warfighting functional areas directly involved in the decision. For example, for a decision requiring a choice of suppression methods against an enemy target where options may include either artillery or close air support, the fire support and aviation leads may be the main contributors. However, all prompts in the trainer elicit responses from all participants. This approach was chosen not only to maintain engagement throughout the exercise, but also for a secondary benefit of yielding insights about shared thinking among the team with each prompt, as participants are asked to give opinions about decisions that do not directly involve their own functional areas.

- Team-driven exercise control. In order to mimic real-world command staff wargaming and also support distributed training, the environment is designed so that the team of participants manages their own progress through an exercise. The leader, in this case the Chief of Staff (COS), is given special controls to advance through "turns" and prompts in the exercise. This is intended to resemble the COS leadership role in real-world wargaming. Exercise controls are designed to be simple (e.g., one-click tools to advance) to minimize special familiarization for the COS in order to carry out an exercise. Also most cases of system-generated feedback are delivered via the COS, who has an opportunity to review before relaying to other staff team members.

These design constraints in the training environment provide background for the discussion of team assessment methods. In constructing automated assessment rules, first it is helpful to delineate constructs relating to team performance and team processes, to help organize behavioral indicators to be monitored by the system (Grand et al., 2013). **Team performance** relates to outcomes in terms of team tasks or status, whereas **team processes** relate to the dynamics of interactions or cognitive states within team members. In a wargaming context, team performance examples include planning decisions in the COA scenario (e.g., a choice of a route or position, unit tasking, timing decision, method of attack, etc.). Team processes can include any of the dynamics that lead to these outcomes (e.g., information sharing or communication in the process of deciding upon a route, etc.). Assessment rules for the wargaming trainer are designed to identify behavioral indicators, either in specific actions or sequences, that match conditions for a teamwork marker of interest. For example, among the exercise environment's mechanisms for overt teamwork actions, the design includes interface elements for team members to express agreement or a need to discuss further, as the team works toward collective decisions during wargaming. These inputs can be analyzed in sequence to recognize patterns of supporting behavior even when lacking knowledge of other modes of communication (e.g., unparsed chat messages). Similarly, actions in the exercise environment serve as a data source for other team process indicators of information sharing, leadership, and knowledge of roles. Measures like the *frequency* of messaging between certain roles associated with certain elements of COA decision making provide insight into team processes, even if freeform messages are not parsed for semantic content.

Team performance assessment rules are designed to look more directly at the content of decisions, which is relevant for dimensions such as team cognition. For example, although COA analysis frequently involves the consideration of competing options without a singular best answer, in some cases there may be decisions that reflect a lack of understanding or a lack of synchronization among the team. In such cases, if an explicitly suboptimal decision is reached, this can be an indicator of poor shared awareness of the situation or objectives. Assessment rules should capture behaviors at the granular individual level in order to support traceability to markers referencing dimensions at the team level. This is especially helpful for individual trainees to be able to see how their own behaviors affect performance outcomes.

Example Training Vignette

In this section we step through a condensed example sequence of prompts and team inputs in a wargaming exercise. For illustration, it omits some team interactions that would occur in a team training event with 6 participants and one or more human observers. The sequence provides context for selected examples of assessment applied to both team processes and team performance.

The participating roles include the Chief of Staff (COS), with five staff leads for Intel, Maneuver (ground maneuver and plans), FSCOORD (fire support coordinator for indirect fires), Aviation, and Logistics (also called sustainment). The exercise is structured around the wargaming, or analysis, of a single course of action (COA), which proceeds through a series of exercise turns corresponding to major events, phases or decision points in the battle. The pattern for each exercise turn involves a series of prompts to first consider the next planned action, the enemy reaction, and the Blue counteraction. Each turn also involves at least one prompt for a primary decision about how the action will be carried out or synchronized across warfighting functional areas. Additional prompts may also be given to probe dimensions of teamwork relevant to the context of the turn at hand. All prompts are designed to conform to one of two types:

- ***Discussion*** prompts assess teamwork dimensions in context
  - Directed to all participants, for the purpose of prompting discussion and gathering inputs, with no explicit step for team review of answers. Participants do not see others' answers.

- ***Decision*** prompts involve COA decisions
  - Often multi-part prompts, for example with a decision and associated rationale.
  - Initially directed to all participants, with a designated lead role for the decision. After all participants respond, there is an opportunity for team review of the lead's answer.
  - After team review, and any revisions, the lead's answer is recorded as a COA decision.

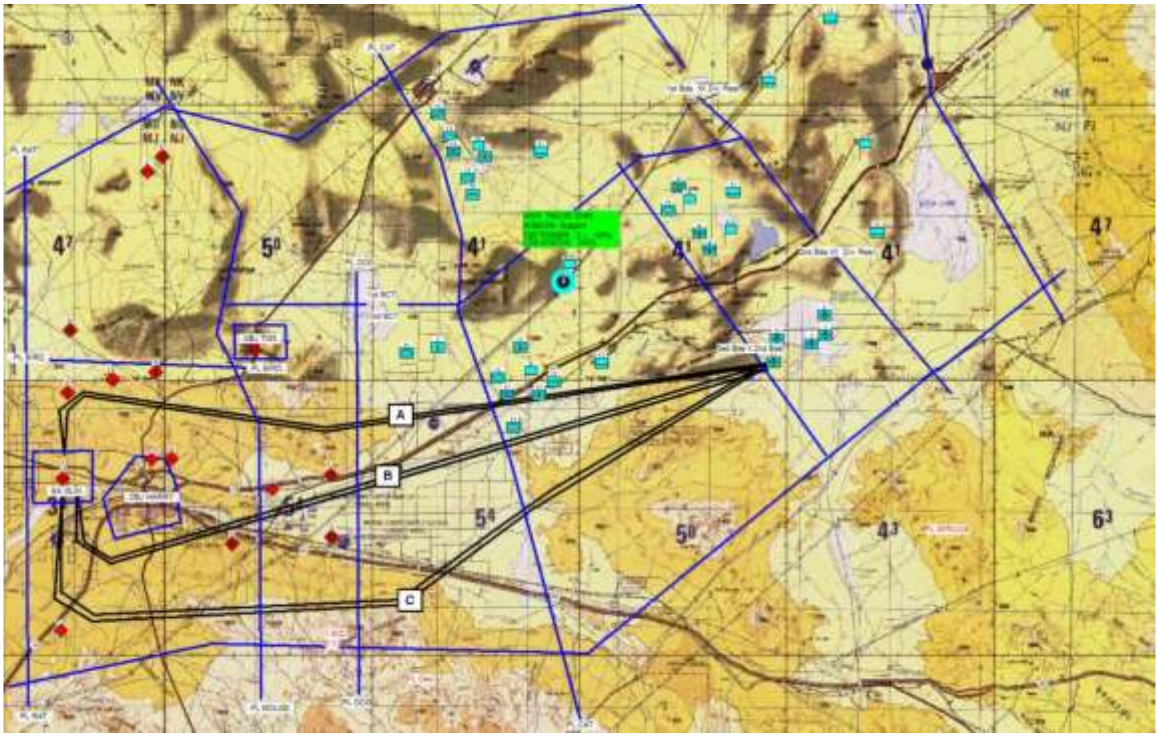


Figure 1. Tactical Map View for Choice of Helicopter Attack Route

Figure 1 above shows the tactical map presented at the start of a wargaming turn involving a helicopter attack. In this turn, the command staff reviews the helicopter attack plan, with prompts about action, reaction, and counteraction, followed by a COA decision to choose among three helicopter attack routes (A, B, or C as shown on map overlays). This is one of the ways that the distributed trainer is a simplification of real-world wargaming, which would not involve predefined choices for attack routes. Although terrain or other elements may ultimately limit the range of options for certain decisions, a real-world training audience would not have options explicitly provided, as they are here. At the start of this wargaming turn, the map is pre-configured to display the overlays with the three routes. Figure 2 steps through a condensed sequence of prompts and team inputs for this turn.

Turn: Plan helicopter attack			
1	Discussion Prompt	[Discuss Action – no decision lead] *What is the purpose of the attack helicopter mission in this COA?*	
	a)	[All]	[Response] *Destroy the Enemy reserve.*
2	Discussion Prompt	[Discuss Reaction – no decision lead] *What enemy defenses are a threat to the helicopter mission?*	
	a)	Intel	[Response] *air defenses at OBJ HARRY, ground units along route*
	b)	FSCOORD	[Response] *air defenses at OBJ HARRY*
	c)	COS	[Communication] *@Intel Please share your thoughts*
	d)	Intel	[Communication] *All routes are threatened by air defenses at OBJ HARRY*
	e)	[Others]	[Responses omitted]
3	Discussion Prompt	[Discuss Counteraction – no decision lead] *What action should we take if the attack helicopters cannot complete their mission?*	
	a)	FSCOORD	[Response] *Use available on-call CAS to attack the enemy reserve*
	b)	Aviation	[Response] *Task the Aviation Brigade to conduct a second attack*
	c)	Logistics	[Response] *Take no further action*
	d)	[Others]	[Responses omitted]
4	Decision Prompt	[COA decision – lead role is Aviation] *Select the best helicopter attack route and give rationale.*	
	a)	FSCOORD	[Communication] *Active fire line planned for RED 208 Arty Bn north of EA GUN*
	b)	FSCOORD	[Response] *Route: B* *Rationale*: *low fratricide risk, minimal conflict with other targets, short distance*
	c)	Aviation	[Response] *Route: C* *Rationale*: *low fratricide risk, minimal conflict with other targets*
	d)	[Others]	[Responses omitted]
	e)	COS	[Exercise Control - trigger review of Aviation inputs]
	f)	Maneuver	[Review] *Agree*
	g)	FSCOORD	[Review] *Discuss*
	h)	FSCOORD	[Communication] *@Aviation B and C are good, but B is shorter*
	i)	Aviation	[Communication] *B overflies too many enemy positions*
	j)	[All]	[Review] *Agree*
	k)	COS	[Exercise Control - trigger logging of Aviation inputs]

Figure 2. Example Prompts and Team Inputs for Choice of Helicopter Attack Route

In the example sequence above, there are three discussion prompts (for discussion of action, reaction, and counteraction), followed by a decision prompt. Sample participant inputs are shown with each prompt, labeled either by individual role (COS, Intel, FSCOORD, etc.) or as a group ([All]) when all have the same input. For this example, there are four kinds of input:

- [Response] inputs are the answers directly responding to a prompt. These are submitted using user interface mechanisms that vary with the prompt, such as multiple choice, checklists, and dropdown menus. For example, 4(b) and 4(c) involve a combination of choosing a route by multiple choice, and specifying rationale from checklists with general purpose objectives.

- [Communication] inputs are messages in the chat panel, created either as freeform text or using a dialog window for pre-formatted messages from templates.
- [Review] inputs capture indications from team members when reviewing a decision. These are simply either an Agree or Discuss value.
- [Exercise Control] inputs are actions taken by the COS to advance exercise states.

Examples of teamwork assessments in the following section refer to the prompts and team inputs in the sequence above.

ASSESSMENT METHODS

The objective of team assessment methods in the wargaming staff trainer is to produce markers of good and bad teamwork from an exercise, for the purpose of team review and self-correction in an after action review (AAR) discussion. For the examples of assessment methods in this section, we focus primarily on the conditions and logic that produce teamwork markers.

Assessment Examples

Referring to the sequence of prompts and team inputs in Figure 2 above, the following examples give a sampling of cases where assessments are made, the conditions they look for, and the team dimensions they relate to.

Response Correctness

The concept of the correctness of individual responses to prompts during the exercises is intended to play only a small role in assessments of teamwork. In MDMP in general, but more specifically in wargaming, but also many other forms of tactical decision-making, there are no singular correct answers when it comes to tactical choices. But 1(a) is an example where a very simple prompt is expected to produce a straightforward answer ("*Destroy the Enemy reserve*."). As the staff team begins the discussion of the helicopter attack, this prompt is a simple spot-check about the purpose of the attack, which is roughly considered a situational awareness level one question. In this case, there really is only one correct answer, and it would be unusual if team responses are not all correct. So while many prompts may provide possible answers that are all considered acceptable, in this case the scenario markup includes information about which answers are considered unacceptable, and what team dimension is involved. The actual assessment mechanism is also very simple in this case, and it only produces a marker when unacceptable answers are received. In 1(a) no marker is created. Had the responses included unacceptable answers, then a marker would be generated and tagged for Team Cognition, relating to team performance (as opposed to process). However, because of the relative low priority on markers related to correctness, this would most likely not be highlighted for team discussion in AAR. Other cases where correctness comes into play can include compound answers to prompts, such as the rationale associated with a route (e.g., 4(b)), which can be evaluated for the validity of the association between the two.

Response Consistency

In 3(a), 3(b), and 3(c), three different staff members give three different responses to the same prompt, a "what-if" question about possible counteractions, as part of the initial structured discussion of the helicopter attack (to consider action, reaction, counteraction). In contrast to prompts involving concepts of correctness, none of these responses are necessarily incorrect, even if some might be considered better than others. However, the main problem with the fact that team members have three different ideas of the best counteraction in this case is that this is likely to lead to problems during execution. In fact, a significant part of the purpose of wargaming is to synchronize plans. In both real-world wargaming as well as the approximation in this exercise, discussion between staff members is encouraged. When each prompt is directed to the training audience, they are welcome to discuss answers before submitting. So the differences in answers also suggest that there has not been enough communication (team process), in addition to the lack of an integrated mission plan (team performance). Assessment rules to generate teamwork markers relating to consistency are tagged with markup about the level of consistency needed, and the team dimensions involved – in this case Information Exchange and Team Cognition.

Leader Support

2(c) is a simple example of leader support during an exercise ("*@Intel Please share your thoughts*"), and often there will be many such examples. The team dimension of Leadership relates to the degree of guidance, direction, and coordination from those in team leader roles. In this case, the COS understands that the Intel officer is the primary source for information about likely enemy capabilities, actions, and tactics. The prompt itself asks about enemy defenses relating to the helicopter attack, with responses available in a checklist. So the COS asks Intel to share thoughts to help the staff, not only for the immediate purpose of responding to the prompt, but also to reach shared awareness about action, reaction, and counteraction. Commonly the Intel officer volunteers this kind of information anyway, as a matter of routine for any questions about projecting enemy actions. In this case, the COS uses a pre-formatted message to Intel, which allows the system to recognize the content of the message and its relevance to the current prompt, and generate a positive teamwork marker for Leadership. This is a simple example, but it should be noted that for more complex examples or also cases where the pre-formatted message dialog is not used, a human observer may also produce the same or similar marker. Generally, the actions of the leader to support the team relate to measures of team processes, and are somewhat independent of the outcomes reflected in the answers collected for the prompts themselves (except for the fact that good team processes ideally produce better team performance outcomes).

Providing Relevant Information

In the previous example with the COS eliciting information from Intel, the information shared by the Intel role (2(d) – "*All routes are threatened by air defenses at OBJ HARRY"*) is useful and relevant. So this also relates to the team dimension of Information Sharing. However, it is not an optimal example. Intel's answers to the prompt included both the enemy air defense threats and also the concern about enemy ground units along any helicopter attack route to be considered. But the Intel officer did not share comments about the latter concern involving ground units. Another example of Information Sharing is in 4(a) ("*Active fire line planned for RED 208 Arty Bn north of EA GUN*"). Once the prompt initiates consideration of the helicopter attack routes, the FSCOORD proactively shares information to inform or remind all staff members about a planned active gun-target line. This involves indirect fires against the enemy at EA GUN at the time of the helicopter attack, so routes passing through this active fire line should be avoided (in this case, avoid route A, in Figure 1). The pre-formatted message dialog is constructed to make this kind of information sharing easy for participants, where role-specific information is available to simply select and send. So for example, the message in 4(a) is only available to the FSCOORD to share. Not all such information is relevant at all times, so the assessment logic looks for cases of messages sent, along with markup about relevance. In this case, a positive teamwork marker is generated, which is related to team processes and involves the team dimensions of Information Sharing and Supporting Behavior. If the FSCOORD typed a similar message, or verbalized similar comments in a co-located or VTC setting, then the system has no knowledge of this kind of natural teamwork, but a similar marker can be created by a human observer or participant.

Participation in Decisions

Another assessment measure related to team processes in wargaming involves the participation of team members in the decision prompts. As discussed above, the decision prompts involve a team review sequence. These prompts are initially directed to all participants, but there is a designated lead role for the decision. In the example exercise turn above, involving the helicopter attack route decision, the Aviation staff lead is the designated lead for the prompt involving the actual choice of the route (prompt #4). After all participants respond, the next step is for the team to review the lead's answer and discuss any possible changes before then recording an answer as a decision for that turn in the COA. For each decision prompt, scenario markup includes information about which roles should be a part of the final decision, based on domain knowledge related to the decision itself. The exercise environment implements a simple formalization for process-oriented data capture in decision prompts, with user interface tools to express agreement or a desire to discuss a decision further.

In 4(g) above, the FSCOORD wishes to discuss the Aviation lead's choice of route C. In subsequent communications, the FSCOORD suggests that route B is shorter (while still avoiding the gun target line mentioned earlier), but the Aviation lead also points out the route B forces the helicopters to overfly too many enemy ground units. Ultimately this leads the FSCOORD (and all other team members) to agree with the choice. There are several opportunities for assessment rules to look for good and bad teamwork markers based on these inputs. If the team proceeds with a decision without receiving any input (*Agree* or *Discuss*) from key roles for the prompt, then this may be a shortcoming in Leadership and/or Supporting Behavior. If a discussion sequence occurs while reviewing the designated lead's input, and the decision changes as a result, this can be an excellent example of Supporting Behavior which can be identified even without parsing the contents of text messages in the chat window. If a team member seeks to discuss the lead input, but the team proceeds with the decision without further review, then this may potentially be an indicator to track for a cumulative effect on Team Orientation if it happens repeatedly (i.e., if team members are being disregarded). Each of these rules is designed to be implemented with relatively simple logic to process the discrete inputs from team members and produce teamwork markers.

GIFT Interoperability

Initial prototyping of the wargaming trainer started with a simple infrastructure where GIFT is used to manage participant profiles and logins with assigned roles within the staff team. From the GIFT lobby the exercise is launched and participants interact with the exercise environment in browsers, with exercise flow and data managed by the RAAFT server. In the previous paper describing the system design (Jensen et al., 2021), we outlined a plan to use the GIFT survey mechanism to deliver prompts throughout the exercise. This included a notional message passing infrastructure between the RAAFT server used to deliver training content to each training participant's browser and GIFT survey utilities, through the Gateway Module to the Tutor Module. However, as details for the nature of prompts evolved beyond the initial design, this made it more complex to preserve the mechanics for switching control from the server-driven exercise interface to GIFT surveys, and also to present the multi-part prompts needed in exercises. Especially with the need to coordinate team member responses within the exercise flow, the updated design plan shifts the implementation of prompts to be entirely within RAAFT client-server components.

By the same token, the RAAFT server also executes the logic for analyzing team inputs and identifying conditions to produce teamwork markers. Team member inputs in the browser client exercise environment are processed and characterized on the RAAFT server to generate teamwork markers, while markers coming from human inputs can also be created manually. Both are recorded on the RAAFT server exercise log, and both are also treated as state transition events in terms of the design for exercise data to be pushed to the GIFT Domain Module. Initially, the primary instructional strategy triggered by these transitions on the GIFT side is to record the teamwork markers for AAR. However, it remains an area of further investigation to determine if there are benefits for generalization, reuse, or authorability if more of the assessment logic can be performed in the GIFT Domain Module. Instead of limiting the outgoing data to the teamwork markers already identified at the RAAFT server, an alternative option to be considered is to create more data throughput with more discrete predicates for user input events in the trainer environment, to be processed by condition class logic in GIFT Domain Knowledge Files to generate markers.

Similarly, we plan to also investigate further about the application of instructional strategies during exercises, and the possible benefits of a more modular function with GIFT. Even operating under the constraints of the browser-based version of a wargaming exercise designed for the initial prototype, there are certain circumstances where immediate feedback or an adjustment in the direction of decision-making is necessary. For example, in order to reduce branching during COA analysis, scenarios are constructed to follow a planned sequence of COA events. If the exercise team makes an early decision that would require a significantly different path in subsequent phases of the battle with the COA under consideration, then some feedback to the training audience or redirection may be needed, beyond the simple assessment function of generating and logging a teamwork marker. In such cases, under the current design, logic based on scenario markup produces a message shared privately first with the team leader (COS), who can then relay to the rest of the team. Using the earlier example with the helicopter attack, if route A were to create serious continuity problems for the remainder of the exercise, then if the team chooses route A, the COS is given a message to relay explaining the problems with route A and the need to settle on a different route. This relay convention also helps preserve natural team relationships with the COS. Under the initial design, this logic for potentially controlling the scenario is implemented on the RAAFT server, but there may be conditions where it is more generalizable to move the logic to instructional strategies in the GIFT Domain Module, using more data throughput with predicates for exercise events.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE USE

Many factors make automated and adaptive training methods complex for domains involving team decision-making. Often in team domains, the constructs for competencies and how they are used as the structure for assessment and feedback are different than in individual training use cases. This paper describes work in progress and examples of some of the assessment mechanisms for training command staff teams in wargaming exercises, with several unique factors that impact how assessments are initially designed and used. One characteristic of the initial training application is that it is not designed around a persistent competency model for specific teams assembled for an exercise. That is, the purpose of assessment and the review of teamwork markers generated in an exercise is limited to the scope of the exercise itself. Starting with this initial narrow purpose, the natural next step for future development is to expand the use case to situations where the same team composition engages in a series of exercises, where the goal is to see improvement among the team as a result of repeating the cycle of conducting exercises and team review.

Another unique factor is that the initial intention is not to produce granular scoring on competencies associated with specific tasks. In a broad sense, the competencies are linked to six dimensions of teamwork organized in a model as mentioned earlier. When teamwork markers are identified during an exercise as a result of automated assessments and/or input from human observers or participants, they are tagged with relevant team dimensions in the model. But the purpose of the markers is more to spur discussion and team self-correction, and less to be used as tallies summed for scores associated with the team dimensions as discrete competencies. One reason for this is that instructors do not have an immediate need for system-generated scores; within the training curriculum the purpose of the exercise is more as practice and less as a scored event. Another reason is the fact that teamwork markers are produced from a combination of automated and human sources, which may skew formulas that would attempt to aggregate results. However, this is another area planned for further investigation, to explore post-processing rules that can analyze the total picture of teamwork markers produced from an exercise not only to yield well-founded high-level scoring outcomes, but also to facilitate the team AAR. It is expected that exercises will produce more teamwork markers than the training audience would want to directly review in an AAR. In order to facilitate team self-review discussion in AAR, the purpose of post-processing rules is to highlight trends and the most significant or meaningful specific markers from the exercise. AAR feedback is designed to allow the training audience to further explore and browse any of the teamwork markers from the exercise, but a distilled AAR highlighting key topics for team discussion is critical. Future research will continue to explore these areas, not only for the specific wargaming training application but also for more general methods that can be applied to other team decision-making domains.

ACKNOWLEDGMENTS

The research reported in this document was performed in connection with contract number W912CG-20-C-0014 with the U.S. Army. The views presented in this paper are those of the authors and should not be interpreted as presenting official positions, either expressed or implied, of the U.S. Government.

REFERENCES

Grand, J.A., Pearce, M., Rench, T.A., Chao, G.T., Fernandez, R., & Kozlowski, S.W. (2013). Going DEEP: guidelines for building simulation-based team assessments. BMJ Qual Saf. 2013 May;22(5):436-48. doi: 10.1136/bmjqs-2012-000957. Epub 2013 Jan 25. PMID: 23355693.

Jensen, R., Presnell, B., DeFalco, J., and Goodwin, G. (2021). Designing a Distributed Trainer Using GIFT for Team Tutoring in Command Level Decision Making and Coordination. *In Proceedings of the 9th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym9).* 176-185.

Johnston, J. H., Cannon-bowers, J. A., & Salas, E. (1998). Tactical Decision Making Under Stress (TADMUS): Mapping a program of research to a real world incident—The USS Vincennes.

Kozlowski, S. W., Grand, J. A., Baard, S. K., & Pearce, M. (2015). Teams, teamwork, and team effectiveness: Implications for human systems integration.

Marlow, S. L., Lacerenza, C. N., Paoletti, J., Burke, 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. https://doi.org/10.1016/j.obhdp.2017.08.001

Salas, E., Sims, D. E., & Burke, C. S. (2005). Is there a “big five” in teamwork? Small Group Research, 36(5), 555–599.

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

Sottilare, R. A., Shawn Burke, C., 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. https://doi.org/10.1007/s40593-017-0146-z

Teo, G., Jensen, R., Johnston, J., DeFalco, J., & Goodwin, G. (2021). Measures for Assessing Command Staff Team Performance in Wargaming Training. *Proceedings of the Interservice / Industry Training, Simulation, and Education Conference (I/ITSEC 2021).*

ABOUT THE AUTHORS

Randy Jensen *has been a project manager for over 20 years at Stottler Henke Associates, Inc., in San Mateo, California, where he has led projects to develop intelligent tutoring systems for domains involving complex decision-making. Examples include trainers for air attack planning, unmanned vehicle command and control, small unit tactics, combined arms team exercises, information systems troubleshooting, satellite scheduling, and a current effort for division level command group wargaming. Randy has a B.S. with honors from Stanford University.*

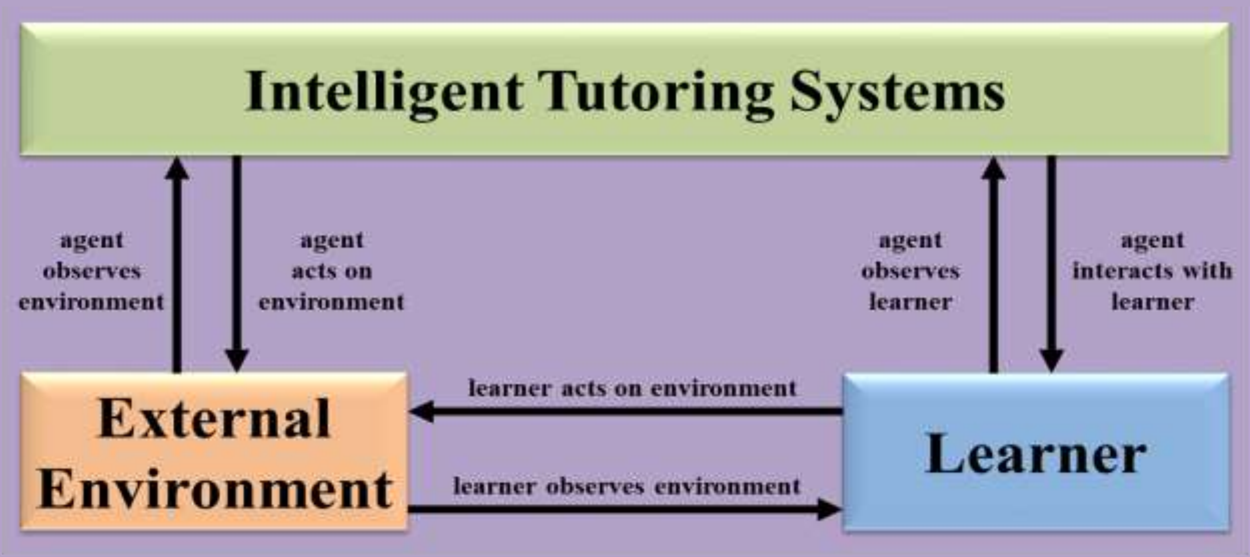
Evan Finnigan *is a software engineer at Stottler Henke Associates Inc., where his work includes developing intelligent tutoring software and building interactive web-based user interfaces. Previously, Evan was a student researcher in a human robot interaction lab where he built web-based interfaces to control an assistive robot. Evan has a B.S and M.S. from the University of California at Berkeley.*

Dr. Grace Teo *is a Senior Research Psychologist at Quantum Improvements Consulting. Grace's research involves understanding and improving human performance under various conditions and in different contexts such as working with different technologies, and in teams. Other research interests include assessments, decision making processes and measures, vigilance performance, human-robot teaming, automation, and individual differences.*

Dr. Gregory Goodwin *is a senior research scientist with the U. S. Army Combat Capability Development Command Soldier Center at the Simulation and Training Technology Center (DEVCOM-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.*

Proceedings of the Tenth 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.


Intelligent Tutoring Systems
agent observes environment
agent acts on environment
agent observes learner
agent interacts with learner
External Environment
Learner
learner acts on environment
learner observes 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. Anne M. Sinatra is a Research Psychologist*** *at the U.S. Army Combat Capability Development Command – Solider Center and has been a part of the Generalized Intelligent Framework for Tutoring (GIFT) team since 2012.*

Part of the Adaptive Tutoring Series

9 780997 725827