Designing a Distributed Trainer Using GIFT for Team Tutoring in Command Level Decision Making and Coordination

Randy Jensen[1], Bart Presnell[1], Jeanine DeFalco[2], Gregory Goodwin[2]
Stottler Henke Associates, Inc.[1], U.S. Army Combat Capabilities Development Center (CCDC)[2]

INTRODUCTION

The increasing use of distributed collaboration environments for collective activities has changed the landscape for how teams must perform, and consequently how they train. Although the demand for technologies supporting remote collaboration has dramatically accelerated due to recent pandemic conditions, the good news is that the trajectory of prior research has produced models, methods and tools for development to address these growing needs. This paper discusses design factors in building a distributed team tutor to provide adaptive training in both teamwork and taskwork, in the context of scenario-based exercises requiring synchronous collaborative team performance. These design considerations are explored using an example application under development, which focuses on training team decision-making and coordination for Army command staff collaborating on the analysis and development of military courses of action (COAs). Although there are noteworthy team performance modeling considerations specific to this domain, the focus of this paper is primarily on the structural design of the distributed team trainer and how it integrates with existing tools. While this paper reports on early stage work and plans for building functionality with the Generalized Intelligent Framework for Tutoring (GIFT), the purpose is to share design concepts with the GIFT community. Ultimately this design may share common features with training needs for other domains.

One way to breakdown the functional elements in a distributed team trainer is to group them into three categories: (i) session and profile management, (ii) the operational, decision-making and communication environment, and (iii) instructional modeling, assessment, and feedback. Session and profile management refers to session status tracking, profile records for individual participants, and team and sub-team composition. The operational environment is where material is presented and exercises are performed, so it includes simulation interfaces specialized to the domain. In our example application this refers to a planned collaborative exercise interface with tactical maps, overlay tools, unit hierarchies, and other tools specific to the war gaming process. The environment also includes collaborative communication tools like chat rooms or other mechanisms to support interactions between remote participants, which may be either general purpose or specialized to the exercises. Instructional modeling, assessment, and feedback are all essential for providing a learning experience where teams receive direct tailored feedback on their performance working together. GIFT offers a reusable framework for building intelligent tutors, which provides models and practices to help with this category of training system elements (Sottilare et al., 2012). Several recent projects have implemented approaches configuring the GIFT Domain Module for team training, either by aggregating individual performance factors for their indications at the team level (Gilbert et al., 2018), or by evaluating performance of collective teams (McCormack et al., 2019). This paper discusses the design of a distributed team trainer integrated with GIFT intelligent tutoring infrastructure, to be deployed in a browser-based setting.

TEAM TRAINING DOMAIN

The application used as an example throughout this paper is a distributed team trainer under development in an effort for the U.S. Army Combat Capabilities Development Command. The effort is called Reusable Automated Assessment and Feedback for Teams (RAAFT), and the objective is to develop automated assessment mechanisms for team training with the goal of reusability in several forms – across scenarios, across platforms, and potentially across domains. Many constructs of teamwork inherently apply to different operational and training settings, so there is the prospect to develop consistent reusable methods grounded in instructional science. For the RAAFT effort the approach to reusability is to start with prototype development in a specific team training domain and then generalize assessment mechanisms from the initial application for reuse.

The team domain selected for the initial prototype under development is command staff war gaming conducted at the division level. War gaming is a clearly delineated step in the Army's Military Decision Making Process (MDMP), involving a deliberate group analysis of one or more courses of action. And yet, although MDMP processes are defined, there is also an art to effective war gaming, which specifically includes factors related to the human dynamics of teamwork. The literature in dimensions of teamwork has provided a number of effective general purpose breakdowns, often with common themes (Johnston et al., 1998; Kozlowski et al., 2015; Marlow et al., 2018; Salas et al., 2005; Sottilare et al., 2018). The model of teamwork to be used for the trainer includes six dimensions relevant to war gaming: Leadership, Supporting Behaviors, Information Exchange, Communication Quality, Team Cognition, Team Orientation. These six dimensions are the underpinning for team assessment in the distributed trainer, which is designed to use both declarative and observational measures. The initial prototype will primarily use declarative measures, which involve directive questions posed to all participants at predefined steps in the war gaming process. There are several categories of questions, to probe different elements of team-level war gaming effectiveness, such as shared mental models, understanding of roles, and team orientation. The observational measures to be developed subsequently will monitor the decisions, actions, and behaviors of participants during war gaming, which can be treated as teamwork indicators for automated assessment purposes. Given this order for planned system development, this paper focuses primarily on the first part – design of the initial declarative assessment measures.

Planned Training Experience

The use case envisions a training setting where all participants are in different physical locations, so primary interactions are in the browser-based environment, potentially augmented with videoconferencing or other live communications channels. However, co-located exercises are also allowed; it is not required that participants be remote. So co-located teams may also use verbal communications or other means. From a training perspective, the primary requirement in this regard is that official decisions and inputs are expressed in the environment, to make data available for assessment.

To illustrate, the following is an example vignette for the kinds of interactions envisioned in an exercise scenario as designed for the trainer. Participants include a Leader and five other staff members representing different warfighting functional areas such as Intelligence, Maneuver, Aviation, Fire Support, and Logistics. There is a status board showing the list of participants with symbology to indicate their status on current activities, such as any decision inputs in progress. Participants all see a shared tactical map view, and all have context-sensitive toolboxes based on their roles and the COA events currently being considered. The Leader has unique tools for controlling the flow through events in the COA. There is also a shared interaction panel for chat messages, tutor messages, and records of decisions from the war gaming process. In the example sequence below, midstream in the analysis of a COA, the team is considering a step that involves a planned helicopter attack.

Leader	Directs Aviation lead to select helicopter attack route
All	See 3 possible routes on shared map view
Army Aviation	Selects a route, selects a rationale from a dropdown, submits
All	See inputs from Aviation
Leader	Prompts Fire Support for consent
Fire Support	Selects "Agree" [options: Suggest / Discuss / Agree], selects a rationale from a dropdown
Leader	Records decisions with COA
Tutor	[Question posed to all] "What intelligence requirements directly support the helicopter attack?"
All	[Free to discuss the question before answering] Select and submit answer(s) from checklist

In this vignette, the Tutor question posed to all is a declarative assessment relating to shared mental models and an understanding of team roles. Throughout the exercise, the questions directed to participants are tailored so that the net combined effect is that they cover a cross-section of teamwork dimensions, while each question is also contextualized to scenario events. The observational assessments in this sequence would relate to team process concerns, such as information exchange between different roles (Aviation and Fire Support) before deciding on a route, the potential for supporting behavior in selecting the route, and the role of the Leader in facilitating this teamwork. The sequence above depicts a nominally effective flow of events, but the intention is to allow for possible teamwork errors. For example, an observational assessment would detect a situation where the Leader concludes the team review of the helicopter attack event, without Fire Support having given input on the route selected by Aviation.

The declarative assessment mechanism is a focus for initial development, and is planned to make use of GIFT survey assessment functionality. The declarative assessment questions are adapted from a set of assessments described by Cianciolo and Sanders (2006) in a conceptual framework for determining war gaming effectiveness. Questions fall into the following categories, specifically oriented toward teamwork in the war gaming process.

Knowledge of own role, and roles of others	Questions about what staff roles need to be involved in information sharing and collaboration for specific war gaming tasks
Tacit knowledge for war gaming	Questions about the relationships between war gaming decisions and inter-related needs of different warfighting functional areas
Team-related motivation	Questions probing perceptions of the utility of staff performance and individual contributions to team outcomes
Adaptivity of team thought	Contextual "what if" questions about the battle situation, enemy courses of action, and contingencies
Shared battlefield visualization	Situational awareness questions to probe for shared mental models
Integrated mission plan	Questions to gauge understanding of rationale for decisions made by specific staff members for parts of the COA

The categories are not necessarily exclusive, so some questions may be formulated to relate to more than one category. Some categories entail questions that are more context-sensitive than others. For example, questions relating to the integrated mission plan are intended to measure a factor of Team Cognition, by looking at the degree to which all team members understand the rationale for staff decision inputs in the war gaming process. Using the example in the earlier vignette, a question might be used to verify that there is understanding throughout the team for the Army Aviation lead's rationale for picking a certain route for the helicopter attack. In this case, both the question and the expected answer are contextualized to the decision input from Aviation. In more general terms, questions of this nature are configured based on the inputs from different staff members for different COA events. So one challenge in this regard is to deliver directive questions that are contextualized to the exercise flow by design and representation. Another challenge is to design the reasoning that takes the aggregate collection of answers from participants, and draw conclusions about teamwork, in a model that uses GIFT structures for team member roles and hierarchy. These goals are discussed further with the structural design below.

STRUCTURAL DESIGN

Since this paper discusses early stage work, the focus is primarily on design factors and plans for supporting the envisioned functionality in the distributed trainer and its integration with GIFT. Especially in a distributed architecture, there are questions about which "side of the fence" each element of the trainer functionality will be implemented on. That is, how features may be supported by existing utilities in GIFT, or supported with adaptations or extensions, or implemented as capabilities in the trainer environment external to GIFT. The overall design of the distributed trainer is broken down by the three functional areas discussed earlier: (i) session and profile management, (ii) the operational, decision-making and communication environment, and (iii) instructional modeling, assessment, and feedback. These functional distinctions are made for convenience in describing elements in a server-based architecture for synchronous exercises involving teams of remote participants. However, while these boundaries relate to distributed training for the sake of discussion, they are not intended to carry any special weight as a unique alternative organizational scheme compared to other designs. At the highest level, the training system design assumes there is a RAAFT server, a GIFT server, and browser-based clients for participants. Much of the complexity lies in defining the data flow between these components. This section lays out design thoughts for building this interoperability.

Session and Profile Management

GIFT provides existing utilities that help with session and profile management at the server, which can be used for outer loop coordination of distributed team members. Since the focus for initial implementation is on the training functionality within team exercises, the main purpose of outer loop functionality in this architecture is to establish the exercise coordination necessitated in the distributed setting. The initial design leaves out outer loop questions relating to the choice of courses or exercises. Inner loop session and profile management functionality is shared by parallel RAAFT and GIFT modules.

The planned design is that users first go through the GIFT Lobby as the precursor to an exercise, where they are given the opportunity to create a new session or join an existing session. By policy, the person designated the leader is responsible for creating a session. Once users are part of a session, they are prompted to select an unoccupied role.

Roles are defined using the team hierarchy organization in GIFT. For the war gaming application, the hierarchy of roles is flat. That is, the team composition is a simple collection of roles without nested sub-teams. So the exercises are defined with one Team level, and TeamMember nodes for each of the participant roles, in this case corresponding to the different warfighting functional areas.

The creator of the session is responsible for starting an exercise, which leads to two triggers. First, the RAAFT server is informed of a new session, and the playerIds and roles associated with each participant. Also the client user interface for each participant is started and informed of the session it is joining. After the client starts up, it connects to the RAAFT server supplying the playerId and session id. The RAAFT server relays updates to the connected clients, which display participant status and also show any expected connections that are still missing.

Inner loop session management related to the tracking of user state information within an exercise is shared between modules on the RAAFT and GIFT servers. Both systems maintain session information. GIFT maintains user session information to use with the Tutor Module and assessment. RAAFT maintains a parallel session profile with detailed records of what actions and decisions have been made in the exercise. These records can be used to restart an incomplete session or replay a scenario. This detailed session information is maintained in a JSON database, which is consistent with the planned method for communication data flow between the RAAFT server and client. Reusing the same structure for session management and persistence saves significant effort from using a table based approach.

There is a related area of interest involving the question of how to handle a persistent team model, as a team counterpart to an individual learner model. Although the current effort is initially focused mainly on inner loop, within-exercise team training, there will likely be a need for future work to define practices for preserving models of team competencies and performance, as teams are directed through a course involving multiple exercises, or even multiple courses. Teamwork factors such as Team Orientation may be difficult to represent and persist at the team level, when considering the possibility of changes in team composition (who's participating in each event) or variations in experience levels among different participants. Ideas for building support for a Team Model as part of GIFT have begun to be explored by Gilbert et al (2018) and others.

Operational, Decision-Making and Communication Environment

The main elements of the environment for distributed team exercises are the front-end client-side user interface and the back-end RAAFT server which manages data flow both to the front-end and to the GIFT server.

Client User Interface Framework

The design of the client side interface where participants interact with the trainer mainly involves the selection of a user interface framework that works readily for the needs of a distributed trainer. We discuss two frameworks – Unity and Angular – evaluated on four criteria: applicability to domain, ease of development, GIFT integration, and flexibility. Some of the benefits of each are enumerated, along with the initial selection of Angular as the framework to pursue based on its overall applicability to the domain, as well as its flexibility in being able to add additional collaborative tools and visualizations.

Unity is a widely used game development platform, originally intended for simplifying the process of developing 3D games. The application has expanded to support 2D games, as well as a large variety of applications for different industries. Unity has an editor for creating screens, and scripting is performed in C#. Unity applications are authored in the editor, and then can be exported to run on a variety of platforms, including as a WebGL application. Unity was considered for two reasons. First, it is a well established tool that carries significant resources and advantages for quickly developing a sophisticated application. Second, Unity WebGL has already been embedded within GIFT.

The second package we considered was Angular. This is one of the standard frameworks for building a web application. Angular unlike Unity was not intended for games, but is instead intended for general purpose applications. Angular is a JavaScript based framework, which allows it to be connected to a number of JavaScript based packages to support a variety of displays and visualizations. Angular was considered for two reasons. First, like Unity, Angular provides a number of tools and resources for rapid development. Second, Angular provides access to a number useful display packages such as map displays and timeline displays. We compared the two frameworks based on four criteria: applicability to domain, ease of development, GIFT integration, and flexibility. For applicability to domain, we judged Angular better. The proposed application is more focused on graphical displays, and standard interface controls such as buttons and menus, rather than fast moving objects, which lends itself more to Angular than Unity. For ease of development, we felt both were comparable in terms of development support. For integration, we gave a slight edge to Unity, as there is already support for Unity WebGL being embedded within GIFT. However, much of this framework could be reworked slightly to support an Angular client. We consider Angular to offer more flexibility. Both platforms have significant flexibility, however Angular seems to offer more possible extensions or services relevant to this application.

The aim is that the in-exercise experience is managed by Angular, which has support for all elements of the user interface design, including the shared tactical map view, role-specific toolboxes, the status board showing participants, and the shared interaction panel. The initial plan is to use the existing GIFT Tutoring UI for tutor interventions and directive survey questions. This may be more tightly integrated with the Angular UI over time, if this leads to a more cohesive experience.

There are two other types of information that will be updated to the user regularly and make sense to consider placing in the same UI element. First, the design treats the team leader as a special role in each war gaming exercise, not only as a participant but also as an exercise controller. Thus leaders are given process cues to help in their role facilitating team execution of the war gaming process. These cues are different from assessment feedback, and other participants have no firsthand view of the system cues provided to leaders. We intend to explore from both a user perspective and an implementation perspective whether to deliver process cues to leaders in the same user interface element as tutor interactions and feedback. In a similar vein, another element of the initial user interface design is a visual timeline to depict the flow of the war gaming process and decisions made along the way.

RAAFT Server

The RAAFT server is built on Node.js to simplify connecting between the front-end client and the back-end, as both systems can be built with Typescript. The RAAFT server and the GIFT Cloud server are connected via the GIFT Gateway Module and a RAAFT Interop Plugin. The connection between the two servers is based on REST protocols.

When an exercise starts, the GIFT server manages the lobby allowing participants to join the session and select their roles. The Interop Plugin informs the RAAFT server to start a new exercise session and the players-role assignments, which triggers the client sessions. Once the Angular client sessions start, they only communicate with the RAAFT server. The primary communication is a REST interface that passes player actions, and then receives updates from the server for state changes. An auxiliary connection based on websockets is needed to support real-time chat communication. The RAAFT Server will communicate actions available to each player. Updating controls based on player suggestions and decisions, as well as updates based on the state of the war-gaming process.

The Angular client could be configured to communicate with GIFT via JavaScript methods from the GIFT Tutor code, but the current design does not use this functionality for two reasons. First, the aim is to allow the client to function outside of GIFT if needed. Second, the RAAFT server is intended to communicate teamwide actions and states to GIFT, rather than having individual communications from each client. However, tradeoffs for this design decision will continue to be explored.

Assessment Methods

For the functional area relating to instructional modeling, assessment, and feedback, our current focus is on assessment methods, specifically for measures of teamwork in distributed war gaming exercises. The initial assessment measures to be implemented will be those associated with the declarative questions that are injected during the war gaming process. The current design for these assessments is discussed in more detail below, in particular to address the question of how some questions will be contextualized to the exercise. The observational assessments will be more complex to implement, as they involve measures that monitor team members' behaviors. For example, complexity is added by the possibility of supporting behaviors, where observable actions can inherently be carried out by different individuals rather than a particular expected person, and still be considered effective teamwork. Although the design details for the observational assessments are not addressed in this paper, there is promising work that may assist with this added complexity. Folsom-Kovarik & Sinatra (2020) describe an approach extending GIFT to associate both Roles and Responsibilities with TeamMembers, to allow for more complex team relationships. For both the declarative and observational assessments, the RAAFT design anticipates that the mappings between team roles and logic to be defined in the GIFT Domain Module are straightforward, partly because of the flat hierarchy of roles for war gaming. This may make it possible to limit the number of Domain Knowledge Files (DKFs) that need to be created, so that there is one for each participant, and one for the team as a whole, but not the combinatoric expansion of DKFs for every n-wise subset of participants. For a team of 6, this means 7 DKFs rather than 63.

Declarative Assessments

One area of complexity in designing declarative assessments involves the generation and evaluation of questions that are contextualized to the decision inputs from participants during the exercise. Referring back to the helicopter route example, consider a question posed to all staff members to determine if they all understand the Aviation lead's rationale in selecting a particular route. The correct answer depends on the inputs previously given by the Aviation lead, and cannot be scripted in advance.

The initial version of the RAAFT server is designed to maintain a data structure with scenario information that includes decision elements involved in the analysis of each COA. Information about the decisions to be made, the roles involved, and the possible rationale factors are all defined in scenario data. So the contextualization is mostly a matter of referencing these items. This scripted format allows participants to gain practice in communication and teamwork with a facilitated war gaming process. This structure does require manual authoring initially, but it also allows us to provide GIFT with updates that are about generic teamwork actions and states that are separated from the domain specifics of the scenario.

The questions to be used as declarative assessments are designed with a representation to capture several features for effective use during an exercise:

- Metadata to identify how questions fit to the context of COA scenario events / conditions
- Categorization in terms of teamwork dimensions, so that the net effect of questions posed throughout the exercise is to have adequate coverage of the range of dimensions
- Representation of the nature of expected answers (e.g., multiple choice, checklist, ranking, etc.), to configure how the questions are presented and scored

GIFT provides different survey instruments, which may apply for different teamwork measures. Multiple choice survey instruments are suited for many questions, relating to adaptivity of thought, shared mental models, and integrated mission plans. Checklists apply for questions relating to the understanding of team roles and Information Exchange, for example to identify roles needing information in certain steps of the COA. Questions expecting answers in the form of a ranking can be used for assessments in the team-related motivation category. These relate to teamwork factors like Leadership or Team Orientation, where the goal is to measure perceptions of the importance or effectiveness of certain team processes. In some cases the goal is to synchronize questions with information to be displayed in the client. For instance, situational awareness questions may query participants about key terrain associated with events in the COA. The simplest approach is to label terrain and provide multiple choices for answers referring to the labeled terrain features by name. A future approach might allow participants to answer by directly clicking on the map. Some questions are assessed in terms of correctness, while others related to team processes are assessed more in terms of consistency within the team. For example, if the Aviation lead chooses a certain route for certain reasons, then a question aiming to assess a shared team understanding of the rationale is looking less for an absolute justification for the route selection, and more for consistency reflecting team members' understanding of the underlying intention.

When a declarative question will be injected, the RAAFT server selects a question template from a library of survey instruments, each containing its own representation of information needed. Based on the current exercise state, configuration information is sent as a survey request through the Gateway Module, to the GIFT Tutor Module and presented. Participant inputs are then captured and processed for what they indicate about individual and team measures.

There are two additional design needs to be considered for the declarative assessments and the use of GIFT surveys. First, in the cases where assessment is a function of consistency within a team, there is a need that scoring should not be completed until all members of the team complete the survey. This is an open area to explore whether this can be accommodated with the standard GIFT survey, which is intended to either just collect information on a single user, or is scored on the basis of correct or incorrect answers. The second feature to consider involves runtime synchronization of the tutor UI during team-wide surveys. GIFT has the concept of teams, and has functionality to coordinate the launch of an external training application. However, we are exploring the best way to design support for synchronizing the pausing of the external training application while all team members take the survey, and then maintaining the paused state until all members of the team complete the survey.

One approach to these needs is to handle both within the RAAFT server and the Interop Plugin. Team-wide scoring involves three steps. First, the surveys themselves are marked as non-scored surveys, to stop GIFT from trying to score the results, so that results are just stored in the User Management System. Second, the Interop Plugin is configured to receive a message carrying the TUTOR_SURVEY_QUESTION_RESPONSE. This provides a means to monitor as players respond to each question. The responses can then be forwarded to the RAAFT server which collates responses from the team as a whole. Once all members of the team have completed a question, the RAAFT server calculates the appropriate metrics for teamwork scores, which are sent to GIFT as a state update to the Domain Module. The synchronization of pauses during surveys is handled in a similar manner. GIFT sends messages to the RAAFT server, which are relayed to Angular clients, to pause and unpause each client as they start and complete a survey. The messages to pause the client are sent immediately, but the unpause messages are reserved until all team members in a session are prepared to continue (or some other state is reached, such as a timeout or leader override). Alternative approaches could involve changes within the standard GIFT modules to support this functionality. Although the approach on the RAAFT side may be more manageable as an initial implementation, there may be value in the future to expanding GIFT survey functionality in several ways to support more team related situations like these.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

The discussion in this paper shares the RAAFT project design thoughts for using GIFT to develop a distributed team trainer for war gaming, while acknowledging that in its early stage of development there are few specific lessons-learned to offer to the community. Future work will lead to more concrete findings resulting from implementation. The intention is to highlight areas where there appears to be existing support within GIFT for needed functionality or a need for new extensions, and also to welcome feedback. The following are some elements and observations from the current design work.

- The distributed training architecture involves a GIFT server, a RAAFT server to manage the training experience, and client user interfaces to be developed with the Angular framework. The servers communicate using REST protocols, and the RAAFT server communicates directly with the clients. Any user-specific client information going to or from GIFT is relayed through the RAAFT server. Session and profile management functions are shared across GIFT and the RAAFT server. Team member roles are defined in GIFT, and the GIFT Lobby is used to coordinate preparation for an exercise.
- There are two kinds of teamwork assessment in planned exercises. Declarative assessments involve questions posed to all participants at specific times during war gaming, using the GIFT surveys. Observational assessments monitor war gaming process behaviors to identify instances of good or bad teamwork. Both are organized for traceability to generalized teamwork dimensions.
- The representation for the declarative assessment questions to be generated includes parametric information for teamwork categories, contextualization to war gaming processes, and the nature of expected answers. These parameters and especially exercise context are prepared at the RAAFT server to create survey requests sent to GIFT.
- Question to be explored: how can the GIFT Tutor Module survey mechanisms support synchronized inputs from a team? For example, when team responses are to be assessed for consistency with each other, is there a best practice for suspending client-side actions and suspending scoring until all responses are collected?
- Question to be explored: for system process cues to leaders facilitating exercise control, is it most effective to use the same mechanisms as tutor feedback, or a different unique leader interface?

The assessment design is intended to have applications for other team trainers. For example, a wide range of team training applications both for distributed or co-located settings may have needs for declarative assessment mechanisms that use GIFT survey functionality but add contextualization to exercise events. Future work defining the observational assessment methods for monitoring team actions in the exercises is also intended to ultimately apply to other team trainers, as the focus is on assessment at the team decision level, abstracted from the platform.

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

Cianciolo, A. T., & Sanders, W. R. (2006). *Wargaming Effectiveness: Its Conceptualization and Assessment* (Technical Report No. 1178; p. 128). U.S. Army Research Institute for the Behavioral and Social Sciences.

Folsom-Kovarik, J. T., Sinatra, A. M. (2020). Automating Assessment and Feedback for Teamwork to Operationalize Team Functional Resilience. In *Proceedings of the 8th Annual GIFT Users Symposium* (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., Holub J., MacAllister A., & Winer, E. (2018). Creating a team tutor using GIFT. *International Journal of Artificial Intelligence in Education*, *28*(2), 286-313.

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

McCormack, R., Kilcullen, T., Sinatra, A. M., Case, A., & Howard, D. (2019). 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.

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

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

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 war gaming. Randy has a B.S. with honors from Stanford University.*

Bart Presnell *is a software developer at Stottler Henke Associates Inc. in San Mateo, California. He has over 15 years experience developing game-based training systems, including projects to train teamwork in medical scenarios, healthy eating and exercise, small unit tactics, and decision making in air combat rescues. Previously Bart worked in video game development at Electronic Arts. Bart has a B.S from Brown University and M.S from the Georgia Institute of Technology.*

Jeanine A. DeFalco, PhD, *is Research Psychologist (Adaptive Training) with the Army Futures Command, CCDC-STTC, Orlando. Current research includes developing pedagogical models for the Generalized Intelligent Framework for Tutoring (GIFT) to accelerate expert problem-solving in critical care medical education, and supporting ethical decision making in soldier tactical and leadership training. She is currently the chair for IEEE P2247.4, Recommended Practice for Ethically Aligned Design of Artificial Intelligence (AI) in Adaptive Instructional Systems. Dr. DeFalco received her PhD in Psychology from Columbia University (2017), specializing in Human Development/Cognitive Studies in Education with a concentration in Intelligent Technologies.*

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 (CCDC-SC-STTC), in Orlando, Florida. For the last decade, he has worked for the Army researching ways to improve training methods and technologies. He holds a PhD in Psychology from Binghamton University and an M.A. in Psychology from Wake Forest University.*