

Analysis of Team Tutoring Training Data

**Anastacia MacAllister, Adam Kohl, Stephen Gilbert,
Eliot Winer**

Iowa State University

Ames, IA

**anastac@iastate.edu, adamkohl@iastate.edu,
gilbert@iastate.edu, ewiner@iastate.edu**

**Michael Dorneich, Desmond Bonner,
Anna Slavina**

Iowa State University

Ames, IA

**dorneich@iastate.edu, dbonner@iastate.edu,
aslavina@iastate.edu**

ABSTRACT

In 2015, the Army identified intelligent tutoring as a key tool for preparing soldiers in an ever-changing world. Intelligent tutoring uses targeted feedback to train soldiers in critical skills. Not only can tutoring train individuals, it can also train teams. As missions become more complex, success requires teamwork. Interactions between team members directly impacts outcomes regardless of individual performance. Currently, there exists some literature looking at the challenges of intelligent team tutoring, however, many of these scenarios have well defined roles that lend themselves well to constructing behavior rules. While these structured roles are easy to construct tutors for, they do not reflect real world scenarios. For the potential of intelligent team tutoring to be fully realized, more realistic tasks need to be studied. This work outlines the analysis strategies developed to decipher task performance from team tutoring data. A simulation-based military resonance task was developed that required communication and coordination between two trainees. The task was designed using VBS2 as the simulation engine and intelligent tutoring was implemented using the Generalized Intelligent Tutoring Framework (GIFT), adapted for teams. Quantitative data collected included entity positions, tutor feedback instances, and subtask performance. As is typical with team tutoring, the amount of data was very large (compared to individual tutoring) and noisy. This paper presents the multiple data parsing strategies and visualizations developed to fully understand the team interactions which took place. These strategies allowed targeted improvement of a team's deficiencies.

ABOUT THE AUTHORS

Anastacia MacAllister is a PhD student in Mechanical Engineering and Human-Computer Interaction at Iowa State University's Virtual Reality Applications Center. She is working on developing Augmented Reality work instructions for complex assembly and intelligent team tutoring systems.

Adam Kohl is a MS student in Mechanical Engineering and Human-Computer Interaction at Iowa State University's Virtual Reality Applications Center. He is working on developing pattern recognition techniques to enhance n-dimensional data visualization methods.

Stephen Gilbert, Ph.D., is an associate director of the Virtual Reality Applications Center and assistant professor of Industrial and Manufacturing Systems Engineering at Iowa State University. His research interests focus on technology to advance cognition, including interface design, intelligent tutoring systems, and cognitive engineering. He is a member of IEEE and ACM and works closely with industry and federal agencies on research contracts. He is currently PI on a project supporting the U.S. Army Research Laboratory STTC in future training technologies for teams.

Eliot Winer, Ph.D., is an associate director of the Virtual Reality Applications Center and professor of Mechanical Engineering and Electrical and Computer Engineering at Iowa State University. He is currently co-leading an effort to develop a next-generation mixed-reality virtual and constructive training environment for the U.S. Army. Dr. Winer has over 15 years of experience working in virtual reality and 3D computer graphics technologies on sponsored projects for the Department of Defense, Air Force Office of Scientific Research, Department of the Army, National Science Foundation, Department of Agriculture, Boeing, and John Deere.

Michael Dorneich, Ph.D., is associate professor of Industrial and Manufacturing Systems Engineering and a faculty affiliate of the human computer interaction (HCI) graduate program at Iowa State University. Dr. Dorneich's research interests focus on creating joint human-machine systems that enable people to be effective in the complex and often stressful environments found in aviation, robotic, learning, and space applications. Dr. Dorneich has over 19 years' experience developing adaptive systems which can provide assistance tailored to the user's current cognitive state, situation, and environment.

Desmond Bonner, is a graduate student in the Human-Computer Interaction and Industrial Engineering program at Iowa State University's Virtual Reality Applications Center. His background is in Graphic Design. His work focuses on using games for learning to increase interest in Science, Technology, Engineering, and Mathematics (STEM) fields as well as leadership within small teams.

Anna Slavina, is a graduate student in Psychology, Education, and Human-Computer Interaction at Iowa State University's Virtual Reality Applications Center. Her background is in psychology. Her research interests include the effects of technology on memory, intelligent tutoring systems, and visually induced motion sickness.

Analysis of Team Tutoring Training Data

**Anastacia MacAllister, Adam Kohl, Stephen Gilbert,
Eliot Winer**

Iowa State University

Ames, IA

**anastac@iastate.edu, adamkohl@iastate.edu,
gilbert@iastate.edu, ewiner@iastate.edu**

**Michael Dorneich, Desmond Bonner,
Anna Slavina**

Iowa State University

Ames, IA

**dorneich@iastate.edu, dbonner@iastate.edu,
aslavina@iastate.edu**

INTRODUCTION

Today's military faces unique threats that are constantly evolving from insurgents in the deserts of Afghanistan to territorial disputes in the Pacific to state sponsored interference in Eastern Europe (Anderson, 2014). Understanding how to respond to provocations and deescalate a situation is important to prevent international conflict and loss of life. Training warfighters to react appropriately in these tense situations requires training tools that can quickly teach a wide variety of skills as new threats emerge (Hou & Fidopiastis, 2016). However, the appearance of wide ranging new threats at an ever-increasing pace makes live action physical training costly and potentially dangerous for novices. To mitigate this problem, Intelligent Tutoring Systems (ITS), coupled with simulation-based training, have been increasingly explored to teach invaluable skills to warfighters (R. A. Sottolare, 2013). These training systems can offer flexibility and adaptability that might be too costly or time consuming in real life (Shubeck, Craig, & Hu, 2016). Not only can these ITS facilitate skill transfer, they can also provide an opportunity to capture massive amounts of detailed data on individual behaviors or team interactions. This data provides the opportunity to analyze and quantify performance, helping pinpoint problem areas in skill acquisition. This type of analysis, while very challenging, provides an incredible opportunity for targeted soldier or unit training (R. A. Sottolare, Holden, Brawner, & Golberg, 2011).

While ITSs do show considerable promise at helping train warfighters in a timely manner, much of the work to date focuses on individual tutoring. Many tasks and missions are extremely complex requiring teams of individuals, thus requiring tutoring on team skills. As these tutoring systems are scaled to address this, creating a scenario and analyzing the resultant data becomes an enormous challenge (Krüger, Merceron, & Wolf, 2010). Specifically, managing the immense amount of data generated by these scenarios and turning raw data into actionable strategies takes considerable skill (Wallner & Kriglstein, 2013). When conducting analysis, especially on a scenario for the first time, care needs to be taken to deal with noise and data artifacts that could obscure results (Bowman, Elmquist, & Jankun-Kelly, 2012; Gibson & Jakl, 2007; Medler, B. & Magerko, 2011; R. Sottolare, Goldberg, Brawner, & Holden, 2012). While computerized scoring works well for something as simple as a marksmanship trainer, when something more complex like team skills are studied a computerized scoring metric becomes harder to develop. For example, consider an individual ITS trainer. This system will have a simulation-based scenario and typically measure several performance metrics (i.e., individual skills). Simply adding a second individual significantly increases the amount of data generated. Now, the system has double the performance metrics to measure (i.e., a set of each individual being trained), but it also has new metrics to consider for the interactions between the teammates. This complexity grows more with the addition of each new team member. It is not uncommon for a team task trainer, with three to four members, to have 10X the data of a comparable single participant trainer.

To ensure accurate characterization of behaviors and training success, data analysis of ITSs needs to be robust. Developing purely automated systems to score and analyze tutoring scenarios is extremely difficult. The more complex a task becomes the harder it is to quantify variables that instructors use to assess training performance. To mitigate the potential for error, data analysis strategies that blend automated and human in the loop analysis to clearly visualize behaviors pertinent to the skills taught are required. This paper discusses the development of data analysis strategies used for tutoring a military resonance team task. First, the development of the team resonance task and the tutor that provided performance feedback on the task to learners is described. The paper then presents how scenario data was collected from the game engine Virtual Battle Space 2 (VBS2), the learners, and the virtual tutor. From here, the different data mining analysis strategies developed to make sense of the data are outlined. In the end, the work

provides insights into strategies that can be employed to collect, analyze, and visualize trainee behaviors that are important to task success.

BACKGROUND

Intelligent Tutoring Systems

Currently there are a limited number of papers that explore military applications for ITSs, but work in this area is increasing as the military recognizes their potential (R. A. Sottolare, 2013). Stottler and Vinkavich developed an early example of a military ITS (R. H. Stottler & Vinkavich, 2000). They explored using a computerized system, rather than an instructor, to teach and provide feedback for tactical action officers learning how to defend a warship. They found the tutor was helpful, allowing students to become more confident in their decision-making skills. However, while the tutor was helpful it was basic and highly specific to that scenario. Stottler et al. aimed to address this problem by creating an ITS that could be adapted to different scenarios across domains (D. Stottler, Fu, Jackson, & Afb, 2006). Unfortunately, the efforts were impeded by the challenges associated with interfacing between different components in a training simulation and dealing with the large amount of data generated. Richards constructed an intelligent tutoring system for teaching pilots about new helicopter cockpits (Richards, 2002). Their work found the tutoring systems helpful at facilitating learning, but ran into issues creating tutoring content for all the possible trainee actions. Goldberg et al. takes the first steps towards building an intelligent tutor for marksmanship training (Goldberg, Amburn, Brawner, & Westphal, 2014). However, instead of trying to construct tutor responses based on hardcoded rules, a challenging endeavor, they collect data from expert marksman to help establish tutor behavior. The goal of the project is to establish a baseline of high performance for the tutor and then provide feedback to novices when a deviation from this baseline happens. While intelligent tutoring does show promise at helping warfighters learn valuable skills, deploying this technology is still going to be challenging. Sottolare describes the issues still preventing the wide adoption of ITS for military use (Robert A. Sottolare, 2015). One of the major issues they address is the limited work in ITS exploring more complex scenarios that have ill-defined rules for tutoring or teams of trainees. While these scenarios are difficult, because of large number of actions possible, they are important for helping ITS realize their full potential, especially in the case of teams.

Data Visualization for Education and Training

While providing performance feedback to a trainee during a scenario is important, it is also imperative that students and instructors can visualize quantified progress over time. Performance metrics allow trainees to gauge their skill level and allow instructors to understand their potential deficiencies. To understand a trainee's skill level, not only does a simulation need to be developed, but also a method for data capture and visualization. Towards the goal of visualization, work by Dobashi aims to quantify and visualize class performance though collecting student engagement with an online Moodle class (Dobashi, 2016). In the work, they track student engagement with an online class though mouse clicks, page views and quiz scores. They construct a time series graph of interaction events overlaid with quiz scores. This visualization allows instructors to pinpoint problem areas for a student in each lecture or subject area. Through using this data instructors are better equipped to tailor instruction to areas where students need the most help. Work by Hu, Lo, and Shih aimed to identify at risk students using data mining techniques (Hu, Lo, & Shih, 2014). They developed a system to track student's online course engagement and performance to help identify those in need of extra instructor attention. The hope was to correct potential behaviors that could lead to dropping out and apply corrections early. The tool they developed, visually, shows a student's learning status compared with others. Abbud and De Miguel look at workload over a time period for air traffic controllers (Abbud & De Miguel, 2014). They use flight data such as aircraft position, flight plans, altitude, heading, speed, heading instructions and flight distance to simulate an aircraft control tower. The workload for a flight controller is charted over time based on the flights present and orders required to maintain a safe position. This workload is charted and overlaid with instructions given at a specific time for each aircraft. This visual allows the workload of a flight operator to be viewed quickly and easily. The visual shows how slight changes in aircraft arrival and departure times can impact a controller's workload. This is an example of how a time-based visualization can be used to make inferences about a complex process.

METHODOLOGY

The methodology section is broken down into three sections. The first discusses the development of the training scenario task, the second talks about data sources and how data was collected, and finally the analysis strategies employed to make sense of the data are presented.

Scenario Development

Testing different data analysis strategies for team tutoring required a training scenario. For this research, a team surveillance task was developed that paralleled accepted military practices. The scenario focused on building strong team communication skills. Because communication is a critical team skill, and communications are relatively easy to measure if implemented using structured language or another formal method, measured variables focused on team observation and communication. The surveillance task was created using Virtual Battlespace 2 (VBS2) as the game engine. It used a model similar to the Event-Based Approach to Training (EBAT) (Fowlkes, Dwyer, Oser, & Salas, 1998; Schaafstal, Johnston, & Oser, 2001) in which simulated events are created for teams to practice specific skills.

The surveillance task requires a team of two, in which each member stood atop a building and was responsible for surveillance of a 180-degree zone (see Figure 1 and Figure 2). At the start of the scenario, enemies emerge from behind walls, quickly move from place to place, and sometimes cross zone boundaries at the single green pole or double green poles. Each team member used the mouse to scan for enemies in his or her zone, looking back and forth as the entire zone could not be viewed at once. Because of the extent of the building rooftop, each member could see only slightly into the other member's zone.

To be more specific, using correct military terminology: the team consisted of two team members. The tasks were as follows:

- **Transfer:** Each team member had to alert the other member whenever an enemy was about to cross into the other member's zone. This had to be done verbally as well as by pressing a specific key on a computer keyboard.
- **Acknowledge:** Each team member had to acknowledge that a communication from the other team member had been received. This had to be done by keypress and accompanied with a verbal acknowledgment.
- **Identify:** After an enemy had been transferred from one team member to the other, the teammate receiving the transfer had to identify the enemy as soon as it entered his or her zone.

For example, if the two team members were Alice (assigned to Zone 1) and Bob (assigned to Zone 2), then as an enemy in Zone 1 approached the zone crossing point with one green pole, Alice would press the 1 key to indicate person crossing at the 1-pole zone and say "Person crossing at the 1." Bob would then press the E key to acknowledge the enemy, say something like, "Got it!" and then press the spacebar when he sees and identifies the enemy who just crossed. The closer the enemy is to the boundary at the moment of transfer, the better the performance (a premature transfer might lead the other team member to look to the zone too early and waste time watching for the enemy to arrive). If an enemy crossed at the two-pole zone, the member pressed the 2 key. These four keys, 1, 2, space, and E, were the only keys used, and were chosen so that the team member could type them all with his or her left hand while using the right hand on the mouse to scan back and forth in the zone. The language spoken by team members was chosen by them. In early implementations of this task, the members were required to use specific phrases designated

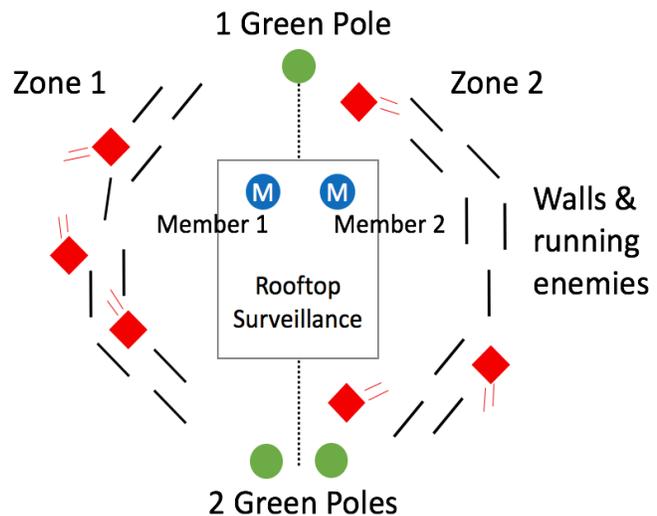


Figure 1. Aerial view of Surveillance Scenario

by the research personnel, but this requirement was sufficiently onerous to prevent some teams from reaching consistent performance levels, even after multiple trials. Additional details on the creation and military basis of this task are described in detail in Bonner, et al. (2015), Bonner, Gilbert, Dorneich, et al. (2016), and Bonner, Gilbert, Slavina, et al. (2016) (Bonner et al., 2015; Bonner, Gilbert, Dorneich, et al., 2016; Bonner, Gilbert, Slavina, et al., 2016).

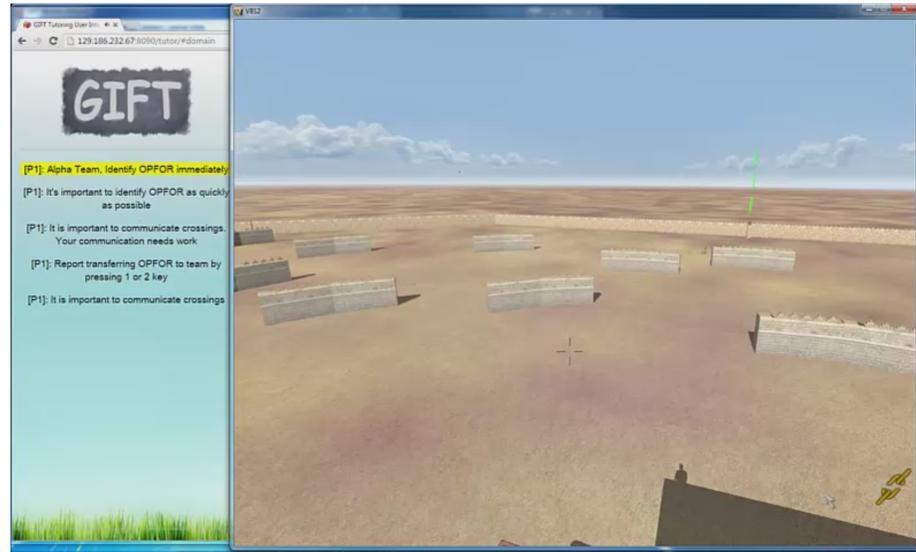


Figure 2. Surveillance Scenario Tutor Screenshot

While the task context was relatively simple, the surveillance task became a powerful research testbed because of the ability to explore numerous team tutor research questions using the same task such as how cognitive load affects ability to perceive feedback, whether knowing one's team member affects performance, how consecutive trials form a learning curve, and how that learning curve is affected by different forms of feedback. With this setup many different dimensions of feedback can be explored. For example, immediate delivery of feedback has been shown to be effective during tasks that imposed a higher cognitive load (Kulik & Kulik, 1988), but feedback at too high a frequency could be overwhelming and result in the feedback being ignored or missed (Fu, Derue, Karam, & Hollenbeck, 2011). The surveillance task could be used to explore how much feedback is too much, and the answer may depend on the team members themselves. The task load on players can be adjusted by changing the number of enemies that appear, subsequently cognitively tasking the players quite heavily. The task is difficult primarily because a member must continuously scan the entire 180-degree zone while watching for enemies appearing (visual search), visually tracking enemies leaving the zone, and listening for transfer alerts. The cognitive complexity arising from this relatively simple scenario was surprising to many participants, who perceived the task as quite difficult, if not completely overwhelming at first.

Data Sources and Data Collection

The VBS2 game engine and GIFT intelligent tutoring system have a well-rounded customizable interface to design training scenarios, but generate a tremendous amount of data as a by-product of the user's interaction with the simulation. The flowchart in Figure 3 brings together a visual representation of the steps encompassing the training simulation and data transformations needed to properly analyze individual and overall team performance. In terms of computation, the team tutoring scenario discussed in the previous section has two major categories, runtime application and post-processing.

The run-time application encompasses the user's interaction with the simulation powered by the VBS2 game engine and GIFT tutoring system. First, the user arrives for a study or training session and proceeds to interact with recon simulation via keystrokes. While the user is training, VBS2 relays raw OPFOR position data and all keystrokes entered by the participant. Next, GIFT processes the user's input and provides targeted feedback to the participant's computer monitor to help improve task deficiencies. Finally, GIFT writes out massive log files regarding system and user information as the training simulation takes place. It's important to note scenario data will not be lost as long as VBS2 and GIFT does not crash. After the training scenario has been completed and the data logged, it's time to start processing the information for analysis. It's the instructor's job to import all of GIFT's simulation log files into the Event Report Tool (ERT) to bridge the gap between the runtime application and post-processing stage. The ERT is an asset of the GIFT tutoring system and designed to convert the massive scenario log files into organized and readable csv files containing pertinent player interactions and performances. After the instructor loads the GIFT simulation log files from each of the study's trials, they then select the information important to the study's analysis for the ERT tool

to filter out of the GIFT simulation log files. There are a number of detailed steps required to analyze the simulation data, but it's necessary to achieve files containing actionable information from the study. Once the ERT script had been run, a participant's csv log file is generated for each of the trials they've performed. Next, the participant csv log files must be organized in corresponding team folders to give structure for further analysis. Furthermore, all team data folders are then imported and run through a custom data analysis and visualization engine created by the researchers. Finally, metrics and chart excel files are generated by a custom parser. These provide the instructor with an understanding of the player's performance history and any issues that require attention.

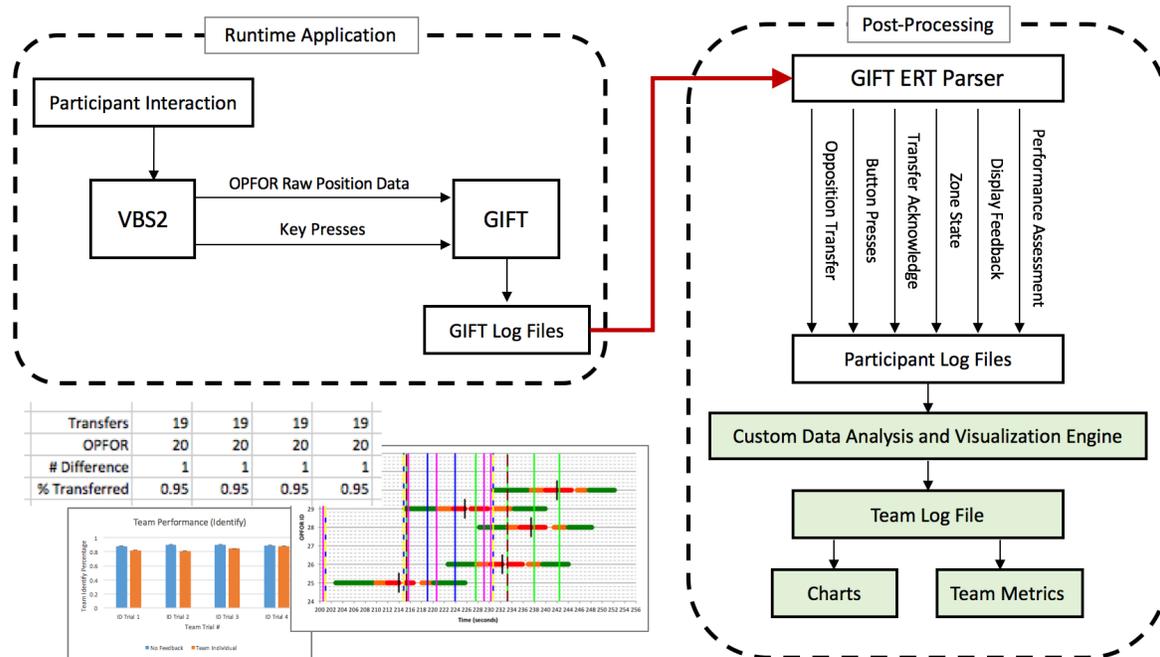


Figure 3. Data Collection and Analysis Process Flowchart

At first, post-processing was not a seamless and repeatable process because the ERT did not produce clean interpretable data for analysis. Even though the ERT provides the instructor access to information about each player's actions, the formatted csv files do not provide an understandable representation of the data. This is due to each of the csv files produced by the ERT containing non-essential information that hides relevant player data in long text message strings. These message strings make extracting a user's event data for statistical analysis challenging. Without having a custom script parse out the relevant data points within each message trends from the study were not discernable from the collected data. These message strings, as outlines in the scenario description above, contained information regarding opposition transfers, button presses, acknowledging transfers, zone states, tutor displayed feedback, and performance assessments. The opposition transfer messages, specifically, contained time and zone location information about when a user hits the transfer key. Button press messages contained the press time and what action key was pressed. Zone state messages contained a list of OPFOR zone locations at a given time. Tutor displayed feedback messages detailed the content participants received on his/her monitor generated from the GIFT tutoring engine during the runtime application. Finally, the performance assessments enclosed the performance ratings participant's scenario tasks.

The study conducted by the researchers consisted of two player teams who ran through the training simulation over the course of four trials. Once all the files for each of the team's trials were converted using the ERT, they needed to be organized in a manner the Custom Data Analysis Visualization Engine could extract information from. A neat and organized file system was crucial to maintaining the correct participant log file for each trial period generated by the instructor running the ERT script.

Data Analysis Strategies

Creating a custom parser requires a significant understanding of the problem and types of metrics pertinent to the study's desired outcome. The researchers wanted to analyze how well the participants performed at the transfer, identify, and acknowledge tasks individually as well as a team. The study ran by the researchers consisted of two player teams who ran through the training simulation over the course of four trials. Therefore, the data needed to be manipulated in a way to not only show how well one performed, but also if feedback given from GIFT improved task performance throughout each trial. As

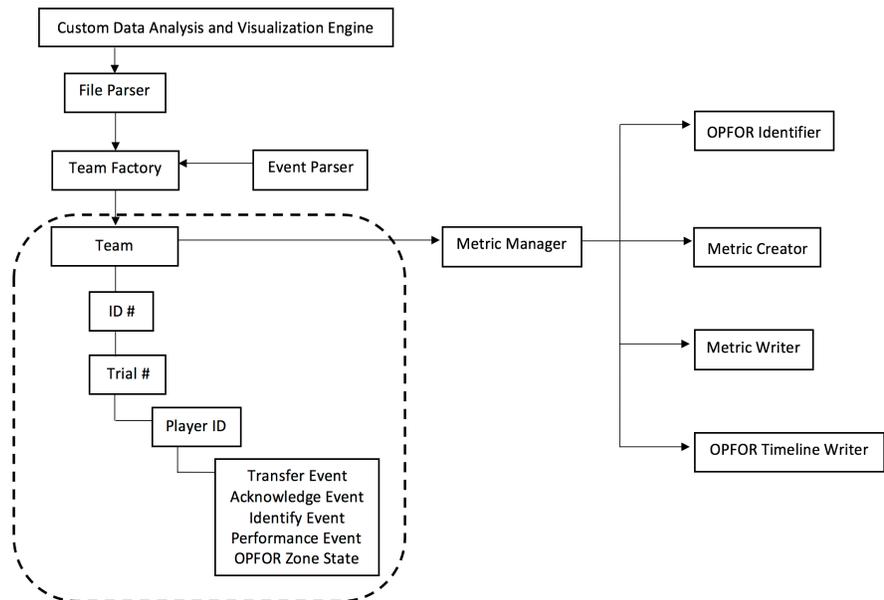


Figure 4. Custom Data Analysis and Visualization Engine Software Architecture

mentioned before, the participant ERT output files were challenging to analyze. To overcome this, a python-based custom data analysis visualization engine was developed. The architecture is shown in Figure 4. The heart of the engine was created in a way to structure the data for each team encompassing all events needed for metric creation.

First, the participant log files are read into the python-based program and parsed by event type. Next, a team was assigned lists corresponding to each event type and populated all lists using the Team Factory class. The team factory generated teams containing an identifying number, and lists distinguishable event types for each trial. It's important to note each event contained their respected values and time information when stored into each team class, but were all linked together according to each team's task. To do this, the code was efficiently structured with the sole purpose to parse through player events to correlate each event's relevant time and location information so that it could be easily used in the future to create reliable corresponding metrics. Once the participant data was properly parsed and stored, the custom parser was tasked with creating excel spreadsheets containing metrics and visualizations.

The Metric Manager was the key to creating the excel spreadsheets and charts. The Metric Manager oversaw the creation of metrics defined by the investigator. More specifically, the Metric Manager relayed crucial information contained in each team from the Team Factory to create excel spreadsheets and charts. Now, the software engine had the ability to automatically replicate excel spreadsheets and graphs for each team located in the input directory, which eliminated the time the researchers had previously spent isolating team events by hand. As a result, time could now be spent analyzing situational behaviors to improve scenarios and bring about generalized study conclusions. In order to analyze data from the scenario, each OPFOR's travel path along with player performance metrics needed to be defined and calculated. The OPFOR Identifier class was responsible for determining the time every OPFOR entered and exited each zone as they traveled from one zone to the other, while the Metric Creator class in turn created timeline measurement and macro metrics for each team's players. At this point in the custom parser process all the data is grouped by team automatically, which saves countless hours of the researchers' time. The Metric Manager then relayed timeline measurements and macro measurement to the Metric Writer, which was responsible for creating excel spreadsheet for each time regarding all the instructor defined metrics. Finally, the Metric Manager relayed OPFOR and player measurement information to the OPFOR Timeline Writer class to generate charts detailing all the events occurring while an OPFOR traveled from one zone to the other. By parsing through and organizing the log files produced by the ERT, the results from the study were placed in a structured format that could then be used to create visualizations to understand team performance. Without the creation of the custom parser structured data format, the individual and team behaviors recorded during the study in the GIFT logs would be extremely challenging to interpret.

After the data was parsed and structured, at first, the researchers sought out high level macro measures of team and individual recon task performance. Macro measures strictly analyzed a player's overall transfer, acknowledge, and identify task performance based on key press numbers. The transfer macro compared the number of OPFORs each player had been tasked with to the amount of transfers each player had made. Separate from the transfer macro was the acknowledge macro. It detailed the amount of times a player acknowledged their partner's transfers versus the amount of transfers their partner had made. Finally, the identify macro compared the amount times a player identified an OPFOR in his/her zone versus the amount of OPFORs that transferred into their zone. Ideally, the difference between task events should be zero signaling the user performed each of the tasks perfectly. Unfortunately, when the data was analyzed a high variance in macro metrics indicated that participants were having issues correctly completing the task. While the macro summary statistics provided general insight in to scenario performance, the metrics were not able to provide insight into what may have caused performance challenges. Because of this, macro metrics can misrepresent the results of the study especially if high cognitive load resulted in pressing incorrect keys on the keyboard to indicate a Transfer, Acknowledge, or Identify. As a result of the issues found with macro measures, more detailed time/event based metrics were developed to provide more information about performance.

The researchers first attempted to seek a detailed analysis regarding how a single participant interacted with the simulation by organizing the task events, triggered by each player, in ascending time steps. The reason for creating the timeline was to first see if a player's transfer presses were matching their partner's acknowledge events. If the pairing of the two events matched it would signal the two players were properly understanding their task assignments and communicating well with one another. Each player had four vital columns making up their task performance: Transferred, Acknowledge, Identified, and Feedback. The spreadsheet gave insight to how the participant's performances increased or decreased over the course of the training simulation, but failed to feature why a participant decided to make certain keystrokes. Each key the participant pressed had been influenced by an OPFOR appearing and moving across the scene, therefore the researchers created a data column containing the time an OPFOR first appeared in the scene. The data column only denoted when the OPFOR started travelling towards the other zone, not the location of the OPFOR at each key stroke. The discrete time stamps signaling an OPFOR appearance failed to illustrate the travel path an OPFOR had taken while moving towards the opposite side of the scene. An OPFOR's location at the time of a participant's button press was needed to understand a player's decision to trigger a transfer, acknowledge, and identify event.

Utilizing the charts depicted in Figure 5 and Figure 6 the researchers could understand that the OPFOR's generated path leads to changing zone exit and entrance times making it hard to automate a definite metric of when the user should have ideally pressed a transfer, acknowledge, or identify key. Instead, the ideal flow of events starts with the player transferring an OPFOR, the partner acknowledging the player has transferred the OPFOR, and finally the partner identifying the OPFOR in his/her zone. For example, if player one pressed the transfer key this should have been aligned next to when player 2 would have acknowledged the transfer and identified the OPFOR in their zone. An overall event-based timeline is a great tool to understand the event trends throughout an entire simulation but can be tough to visually interpret an event's intended purpose. To give insight in to why a participant may have triggered a transfer, acknowledge, or identify event the researchers created a timeline chart that isolates the OPFOR's as they travel through the simulation's scene. The timeline chart puts into perspective the range of times where cognitive load may have varied dramatically, a crucial aspect macro measures cannot show. To perceive the participants' training, the researchers decided to create the graphical representation in Figure 5. The timeline illustrates each player's button press with a colored vertical line overlaid onto green, orange, and red horizontal lines that denote the zone location of each OPFOR location during its time of travel.

In a broad sense, the OPFOR start behind a blockade and enter in then out of pre-defined zones while travelling from one player's side of the scenario to the other. The pre-defined zones cannot be seen by the participants. Instead, the zones are colored coded to aid in evaluating a player's performance. A green zone is the ideal time for a participant to trigger a transfer and acknowledge event because it's located the farthest from the border crossing. As the OPFOR get closer to the border boundary the zones change from green to yellow and finally red. The changing colors correspond to depreciating performance because of the limited time a participant gives their partner to successfully complete the task. First, the OPFOR enters the green zone where a player should press the transfer key to signal to their partner an OPFOR is going to be crossing to their partner's side where the OPFOR must be acknowledged and identified. As the OPFOR continues traveling it proceeds into the orange zone then the red zone before coming to the midpoint of the scenario. Overlaid on top the OPFOR zone movement are vertical lines for each of the player's key presses. The transfer events were then labeled pink, acknowledge events labeled blue, and identify events labeled green. If a player

were to press the transfer key while the OPFOR was in the orange or red zone, the transfer event would be considered late. Since the user input is only logged based off a time stamp there cannot be a definite certainty a key press is associated with a certain OPFOR movement. This makes analyzing which key press belongs to which OPFOR difficult.

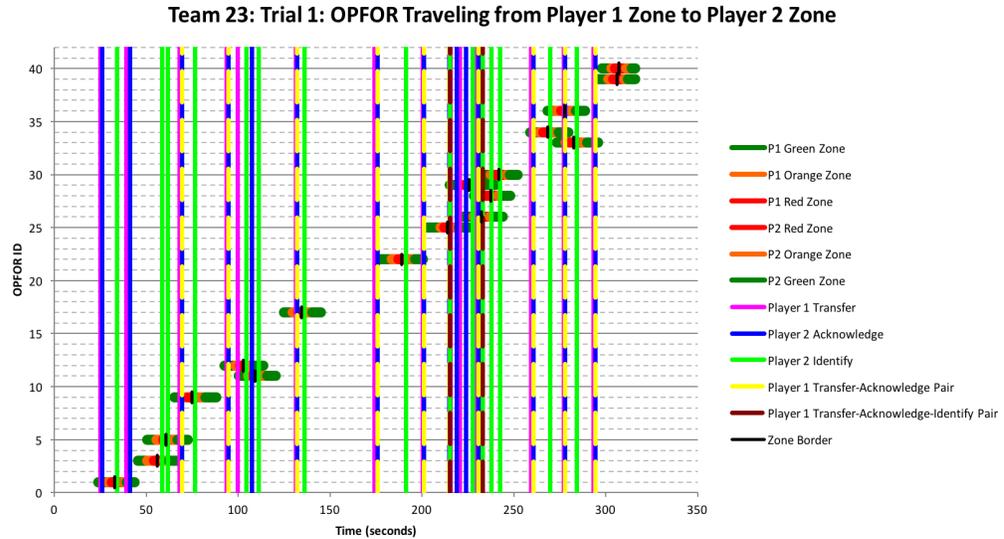


Figure 5. OPFOR Timeline Chart

For example, there may be four OPFOR crossing zones at the same time which should yield four transfer presses. The same ideology can be applied when the partner needs to acknowledge the simultaneous transfer presses and identify the multitude of OPFORs that enter his/her zone. The researchers also wanted to see if the transfer of one player would align with the acknowledge of another. The measure for this metric is known as the Transfer-Acknowledge Pair and is highlighted in yellow. The researchers also wanted to see if there were any transfers from player 1 and a correlating acknowledge along with identify from player 2. This metric is known as the Transfer-Acknowledge-Identify Pair and is represented with a brown line on the chart. In summary, if the participants performed perfectly there would be as many Transfer-Acknowledge-Identify pairings as number OPFORs.

The researchers then found the OPFOR Timeline Chart could be difficult to understand when there were simultaneous OPFORs crossing the zone midpoint. As seen in Figure 5, there are too many cluttered lines between 200 and 250 seconds which made it extremely difficult to understand what was happening. The chaotic sections containing multiple OPFOR crossings usually comprised a large amount of triggered events by each participant in a short time span. To battle cluttered player events the researchers re-scaled and focused the timeline chart a finite time interval as seen in Figure 6. Now each individual OPFOR could be analyzed on a more detailed scale to see when each participant triggered a transfer event while their partner acknowledges the OPFOR transfer and identified when the OPFOR appeared in their zone. In the case of team 23 represented in Figure 6, the chart shows the instructor the participant is overwhelmed due to the lack of necessary transfers, and their partner may be struggling with identifying the five OPFOR crossing during the short time span.

Team 23: Trial 1: OPFOR Traveling from Player 1 Zone to Player 2 Zone

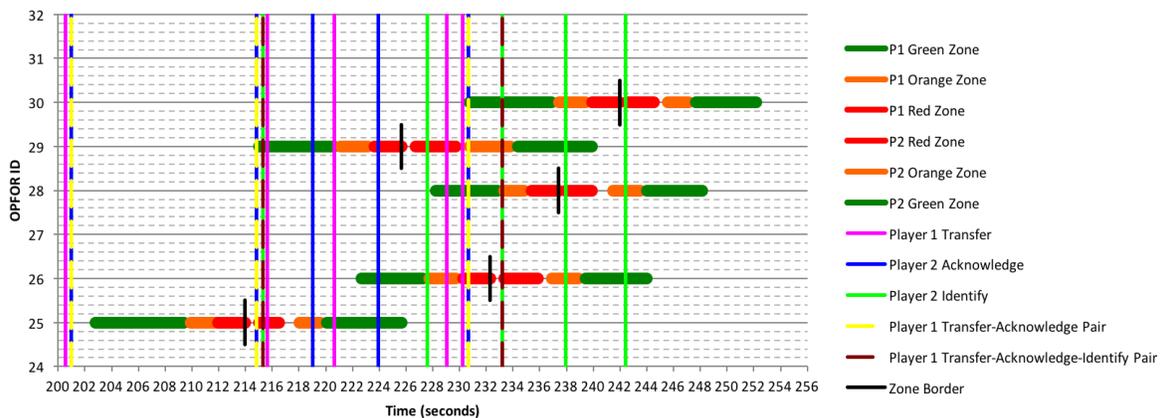


Figure 6. OPFOR Timeline Section

The overall timeline chart shows the cognitive load varies for each participant throughout the simulation. Since the timeline chart represents the time sensitive tasks required by each participant the instructor can evaluate if the tasks are too demanding for the participant to handle or if the simulation needs to be altered to create a set of more challenging tasks.

CONCLUSIONS AND FUTURE WORK

In conclusion, developing data analysis strategies for a complex team training scenario is a challenging task. Once a scenario is developed data needs to be collected and analyzed. This data collection can be a challenge with all the different systems that often make up training applications. Parsing through the data in order to understand how multiple trainees performed on a task both individually and as a team takes considerable skill. An analyst must consider not only the behavior of the trainees, but also the potential for unintended data capture that may obscure the results. The strategies discussed above illustrate how important it is to not only consider macro measures of performance, but also a time event based visualization tool to capture all the information possible. By employing different visualization and performance metrics a more complete picture of the simulation is drawn, providing more information on team performance. Moving forward, the researchers will continue to refine the visualization aid and develop automated scoring metrics to make evaluating performance of the recon task easier.

REFERENCES

- Abbud, J., & De Miguel, G. (2014). Air Traffic Management and Systems. *Lecture Notes in Electrical Engineering*, 290(April), 175–191. <http://doi.org/10.1007/978-4-431-54475-3>
- Anderson, E. W. (2014). *Global Geopolitical Flashpoints: An Atlas of Conflict*.
- Bonner, D., Gilbert, S., Dorneich, M. C., Winer, E., Sinatra, A. M., Slavina, A., ... Holub, J. (2016). The Challenges of Building Intelligent Tutoring Systems for Teams. In *Paper presented at the Human Factors & Ergonomics Society (HFES) Annual Meeting*.
- Bonner, D., Gilbert, S., Slavina, A., Dorneich, M., Winer, E., Holub, J., ... Kohl, A. (2016). The Hidden Challenges of Team Tutor Development. In *Proceedings of 4th Annual GIFT Users Symposium (GIFTSym4)* (pp. 49–60).
- Bonner, D., Walton, J., Dorneich, M. C., Gilbert, S. B., Winer, E., & Sottilare, R. A. (2015). The Development of a Testbed to Assess an Intelligent Tutoring System for Teams. In *Proceedings of the Workshops at AIED 2015* (Vol. 6, pp. 31–37).
- Bowman, B., Elmqvist, N., & Jankun-Kelly, T. J. (2012). Toward visualization for games: Theory, design space, and patterns. *IEEE Transactions on Visualization and Computer Graphics*, 18(11), 1956–1968. <http://doi.org/10.1109/TVCG.2012.77>
- Dobashi, K. (2016). Development and Trial of Excel Macros for Time Series Cross Section Monitoring of Student Engagement: Analyzing Students' Page Views of Course Materials. In *20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems* (Vol. 96, pp. 1086–1095). The Author(s). <http://doi.org/10.1016/j.procs.2016.08.133>
- Fowlkes, J., Dwyer, D. J., Oser, R. L., & Salas, E. (1998). Event-Based Approach to Training (EBAT). *The International Journal of Aviation Psychology*, 8(3), 209–221.
- Fu, C., Derue, D. S., Karam, E. P., & Hollenbeck, J. R. (2011). The impact of feedback frequency on learning and task performance: Challenging the “more is better” assumption. *Organizational Behavior and Human Decision Processes*, 116(2), 217–228. <http://doi.org/10.1016/j.obhdp.2011.05.002>
- Gibson, D., & Jakl, P. (2007). DATA CHALLENGES OF LEVERAGING A SIMULATION TO ASSESS LEARNING, (Celda), 141–149.
- Goldberg, B., Amburn, C., Brawner, K., & Westphal, M. (2014). Developing Models of Expert Performance for Support in an Adaptive Marksmanship Trainer. *Proceedings of the Interservice/Industry Training Simulation and Education Conference*, (14214), 1–12.
- Hou, M., & Fidopiastis, C. (2016). A generic framework of intelligent adaptive learning systems: from learning effectiveness to training transfer. *Theoretical Issues in Ergonomics Science*, (January), 1–17. <http://doi.org/10.1080/1463922X.2016.1166405>
- Hu, Y.-H., Lo, C.-L., & Shih, S.-P. (2014). Developing early warning systems to predict students' online learning performance. *Computers in Human Behavior*, 36, 469–478. <http://doi.org/10.1016/j.chb.2014.04.002>
- Krüger, A., Merceron, A., & Wolf, B. (2010). A Data Model to Ease Analysis and Mining of Educational Data. *Proc.*

- of the 3rd Int. Conf. on Educational Data Mining (EDM'2010), 131–140. Retrieved from http://educationaldatamining.org/EDM2010/?page_id=278
- Kulik, J. A., & Kulik, C. C. (1988). Timing of Feedback and Verbal Learning. *Review of Educational Research*, 58(1), 79–97.
- Medler, B. & Magerko, B. (2011). Analytics of Play: Using Information visualisation and gameplay practices for visualising video game data. *Parsons Journal for Information Mapping*, 3(1), 1–12. Retrieved from <http://piim.newschool.edu/journal/issues/>
- Nesbit, J. C., Nesbit, J. C., Liu, A., & Liu, Q. (2015). Work in Progress: Intelligent Tutoring Systems in Computer Science and Software Engineering. *122nd ASEE Annual Conference & Exposition*.
- Peña-Ayala, A. (2014). Educational data mining: A survey and a data mining-based analysis of recent works. *Expert Systems with Applications*, 41(4 PART 1), 1432–1462. <http://doi.org/10.1016/j.eswa.2013.08.042>
- Richards, R. A. (2002). Principle Hierarchy Based Intelligent Tutoring System for Common Cockpit Helicopter Training. *Intelligent Tutoring Systems*, 473–483. http://doi.org/10.1007/3-540-47987-2_50
- Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33(1), 135–146. <http://doi.org/10.1016/j.eswa.2006.04.005>
- Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 40(6), 601–618. <http://doi.org/10.1109/TSMCC.2010.2053532>
- Schaafstal, A. M., Johnston, J. H., & Oser, R. L. (2001). Training teams for emergency management, 17, 615–626.
- Shubeck, K. T., Craig, S. D., & Hu, X. (2016). Live-action mass-casualty training and virtual world training: A comparison. In *Proceedings of the Human Factors and Ergonomics Society 2016 Annual Meeting* (pp. 2113–2117).
- Sottolare, R. A. (2013). Adaptive Intelligent Tutoring System (ITS) Research in Support of the Army Learning Model—Research Outline, ARL-SR-028(December). Retrieved from <https://gifttutoring.org/attachments/520/ARL-SR-0284.pdf>
- Sottolare, R. A. (2015). Challenges in moving adaptive training & education from state-of-art to state-of-practice. *CEUR Workshop Proceedings*, 1432, 1–8.
- Sottolare, R. a., Holden, H. K., Brawner, K. W., & Golberg, B. S. (2011). Challenges and Emerging Concepts in the Development of Adaptive, Computer-based Tutoring Systems for TEam Training. *Interservice/Industry Training, Simulation, and Education Conference (IITSEC)*, (11007).
- Sottolare, R., Goldberg, B., Brawner, K., & Holden, H. (2012). A modular framework to support the authoring and assessment of adaptive computer-based tutoring systems. *Interservice/Industry Training, Simulation and Education Conference (IITSEC)*, (12017), 1–13. <http://doi.org/10.13140/2.1.4863.2329>
- Steenbergen-Hu, S., & Cooper, H. (2013). A meta-analysis of the effectiveness of intelligent tutoring systems on college students' academic learning. *Journal of Educational Psychology*, 105(4), 970–987. <http://doi.org/10.1037/a0032447>
- Stottler, D., Fu, D., Jackson, T., & Afb, B. (2006). Applying a Generic Intelligent Tutoring System (Its) Authoring Tool To Specific Military Domains. *Security*.
- Stottler, R. H., & Vinkavich, L. M. (2000). Tactical action officer intelligent tutoring system (TAO ITS). *The Interservice/Industry Training, Simulation & Education Conference (IITSEC)*, 2000(1). Retrieved from <http://ntsa.metapress.com/index/6UUFH1NFY0417ALP.pdf>
- Wallner, G., & Kriglstein, S. (2013). Visualization-based analysis of gameplay data--a review of literature. *Entertainment Computing*, 4(3), 143–155.