INTEGRATING GIFT, COMPETENCIES, VIRTUAL REALITY, AND BIOMETRICS TO PRESENT TRAINING PERSPECTIVES ON GAUGING CURRENT SQUAD CAPABILITY

Zach Heylmun(a), Mike Kalaf(b), Chris Meyer(c), Christofer Padilla(d), Lucy Woodman(e)

(a),(b),(c),(e) Synaptic Sparks, Inc.,(d) Dignitas Technologies

ABSTRACT

The Generalized Intelligent Framework for Tutoring (GIFT) software suite provides many capabilities, and this paper explores and demonstrates the subsets of GIFT External Application Integration, GIFT Sensors, and the training value that becomes achievable as a result through integration with an Intelligent Tutoring System (ITS) such as GIFT. This paper describes experimental hypothesis for new training value, and the process of integrating GIFT together with Virtual Reality (VR), other relevant software suites such as Competency and Skills System (CASS), and some of the low-cost Internet-of-Things (IOT) hardware packages. Keeping

U.S. Army modernization priorities in mind, the authors present the methodologies and results from creating a prototype software engineering package dedicated to the integration of these systems. Conclusions and community interest encourage performing further research and experimentation as future programmatic priorities allow.

Keywords: intelligent tutoring systems, virtual reality, internet of things, team training framework

INTRODUCTION

ITSs have continued their conceptual evolution and implementation in parallel to modern technologies that can be configured to add value to ITSs’ overall effectiveness. Along with the evolution of technology and software, the Department of Defense (DoD) and the U.S. Army have evolved their programmatic practices to include modernization priority updates for 2018-2019. As part of this volunteer effort; it is to experiment-with and add-value-to one of those modernization priorities, namely Soldier Lethality, that the authors proposed that GIFT could be used to supplement.

GIFT was configured to integrate with and utilize Commercial-Off-The-Shelf (COTS) hardware for both VR solutions and external sensors in this effort to create an experimental scenario for modernized adaptive training. GIFT also had some of its internal user interfaces updated to display dynamic mastery and competency information using a CASS test database at https://cassproject.github.io/cass-editor/. By combining GIFT’s capability to adapt training content with a competency standards system such as CASS, the authors hoped to enable formal experiments that measure training value as a system when compared to the individual software components alone.

Primarily, this effort created the framework with which to further experiment integrating GIFT with biometrics and virtual reality, along with other future IOT devices and software suites. The authors’ teams consisted mainly of working professional engineers contributing to nonprofit efforts, but the direct team member roster did not contain Army Subject Matter Experts (SMEs) or behavioral scientists / doctors with which to form experimental hypothesis and validations. It was the authors’ intent that the framework may now be tailored to suit specific scientific needs in the community having enabled the prototype functionality.

The framework was secondarily created to provide GIFT with another set of sensors and applications with which to integrate and perform future training scenarios with. The training scenario produced as part of the paper’s effort was not constructed as an actual DoD course, but rather as an example on “the art of the possible” on how to use many of the technological evolutions that the IOT- style of hardware production has provided modern society. Sample IOT devices included with this experiment include pulse monitoring sensors, indicator lights, haptic motors, and real-time situational team knowledge simulation, all of which can be purchased through COTS providers for under $20 total per trainee.

Combining GIFT, VR, CASS, COTS IOT hardware, and the team’s engineering experience, the authors created, to best effort, a breadth-first course containing a shallow dive into all of these different areas proving high levels of interoperability from GIFT. Proposed future experimentation variables include the type and fidelity of VR content, learner characteristics such as familiarity with VR and trainee career background, length of time in VR while in a scenario, difficulty of the scenario referencing the number of IOT devices selected, and the quality of results as different types of adaptation are used during a scenario. This project was not funded from any source and is released with full rights to any interested public entity. Even in this case where the quantity of training content was minimal, the results of the effort should prove to be of interest to the research community, the GIFT team, and possibly multiple branches of the military as training scenarios continue to modernize along the path of higher technology.

PARTS LIST AND DESCRIPTIONS

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

GIFT Software Suite

At the time of this writing, GIFT 2019-1 has been released at [www.gifttutoring.org/projects/gift/files.](http://www.gifttutoring.org/projects/gift/files) If the reader has not yet created a GIFT account to enable the download, registration is free by following the link to ‘Register’ on the web page. The GIFT 2019-1 download will allow the reader to install and configure their own local GIFT server for any purpose. Instructions for configuring a GIFT server and discussions on the matter can be found included with the download and on the Forum tab at the [www.gifttutoring.org](http://www.gifttutoring.org/) homepage.

CASS Online Test Database

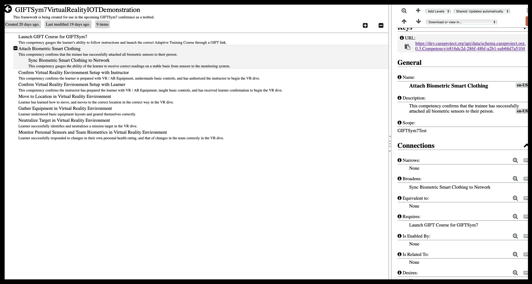
By following the https://cassproject.github.io/cass- editor/ link, readers may explore the site to edit and configure their own competency and mastery framework. This server is maintained by the Advanced Distributed Learning (ADL) CASS team, but any user has full permissions to create and edit their own framework. Readers may search for ‘GIFTSym7’ to examine the framework created for this paper’s effort. Readers may also register with the CASS project at https://[www.cassproject.org,](http://www.cassproject.org/) download the open source code from the referenced GitHub project, and build/configure/maintain their own CASS server.

Figure 1: Screenshot of CASS Framework for GIFTSym7 in CASS Editor Web Tool

MQTT Online Network Communication Server

A tool called Mosquitto (with two T’s) is a lightweight communication protocol that follows a publish/subscribe model of network communications. Furthermore, a public test server is available at https://test.mosquitto.org. Readers may also download the server/client itself to setup their own instance on any available machine. The online server is free for the public to use, and was incredibly valuable as an online communication server through which to test messages being passed between the various components of the paper’s system. While building the prototype system simply using ‘localhost’ methodologies was possible, the team wished to demonstrate the potential power and distributed nature of open API software such as GIFT, CASS, Unity VR, and modern IOT devices.

Unity

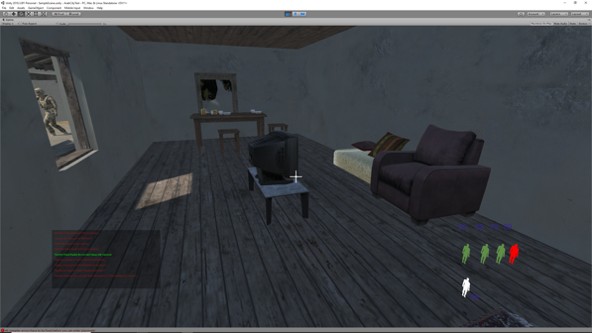
The Unity game engine was chosen to host the prototype virtual environment due to GIFT’s pre-existing integrations and course content utilizing Unity. A personal version of Unity may be downloaded from https://unity.com for noncommercial users. It should be noted that GIFT is an agnostic ITS, meaning that there is no restriction on which virtual environment that adaptive tutoring can be performed in. For instance, GIFT already contains a Course Object to manage many available interactions with Virtual Battle Space (VBS): https://bisimulations.com/products/virtual-battlespace. Proper configurations of network communications according to GIFT’s open API instructions allow for any environment to become part of adaptive training and allow for virtual environment events to be monitored or injected according to GIFT course direction. New experimental features in GIFT that are currently being developed also allow for more generic communications and management of scenarios within virtual environments.

Figure 2: Unity Scenario Room with HUD

Learner Record Store – Learning Locker

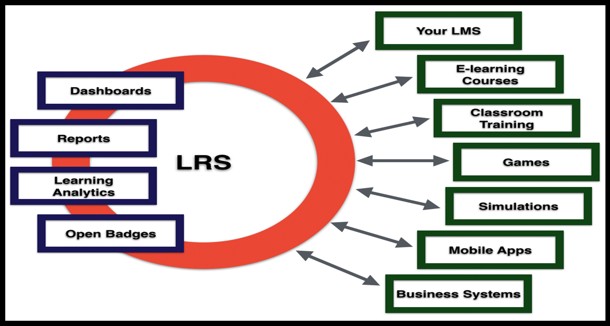
Interfaces to a Learning Management System (LMS) and/or Learner Record Store (LRS) were not integrated as part of this paper’s effort, but GIFT does contain existing interfaces to an LRS called Learning Locker, https://[www.ht2labs.com/learning-locker/.](http://www.ht2labs.com/learning-locker/) When tracking permanent student performance, specifically as it relates to competency and mastery acquisition, GIFT can agnostically communicate with any open API LMS or LRS given software development time to create the simple translations of existing network messages. This capability will allow GIFT to store permanent individual and team performance measures and progressions accessible in the CASS database as Army Subject Matter Experts (SMEs) enter in competency information and relationships.

Figure 3: Learning Locker xAPI-Enabled Communications Across Different Devices

Hardware

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

* Vive VR System: https://[www.vive.com/us/](http://www.vive.com/us/)
* 3-Watt, 8-Ohm Single Cavity Mini Speakers
* ¼-Watt, 470-Ohm Resistors
* Jumper Wires for Arduino
* Arduino Uno Microcontroller
* Arduino Network Shield
* 8,000 RPM Micro DC Motors
* 40V, 600mA, 300MHz, 625mW Transistors
* Red and Green 6-13V LED Diodes
* 10uF, 50V, 105c Capacitors
* Solderless PCB Breadboard
* Pulse Sensors
* ¼-Watt, 1k-Ohm Resistors
* 1000uF, 25V Capacitors

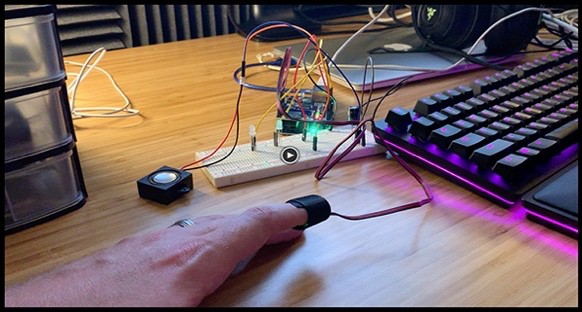


Figure 4: Arduino Hardware Components

The hardware listed above resulted in a combined system that was capable of monitoring a trainee’s pulse rate tracking “health readiness,” sending Go/No-Go visual and audio cues, sending indicator vibrations at different strengths through haptic motors attached to a trainee, all while performing a scenario in a desktop or VR Unity simulation tracking masteries referenced in CASS being monitored and managed by GIFT.

METHODOLOGIES

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

GIFT Installations, Configurations, and Modifications

It is assumed that the reader has a working knowledge of the GIFT software suite in order to fully understand this section. Special attention is given to the new areas of study that this paper explores, but only brief mention is given to basic GIFT topics. For more information, the reader may refer to GIFT documentation at [www.gifttutoring.org,](http://www.gifttutoring.org/) or How-To YouTube videos at https://[www.youtube.com/channel/UCWtI\_V8f2mN5X](http://www.youtube.com/channel/UCWtI_V8f2mN5X) D6h2lCjsAA.

Prerequisites to this section include having downloaded GIFT 2019-1, fully having configured GIFT server, and having configured network communications and hosting to the point of understanding the reader’s system being localhost vs. hosted online at a specific IP address or DNS web address. The authors wished to fully demonstrate GIFT’s interoperability with the distributed online world, and thus used fully-hosted online servers for all following development described in this paper.

A key concept to understand about configuring GIFT to be part of a distributed system is the nature of ActiveMQ / MQTT network communication. GIFT maintains its own ActiveMQ server on the machine it is running on, as well as being able to communicate with other non- centralized network servers through many different methods such as RESTful web service calls or custom ActiveMQ/JSON messages. Fully explaining security, safety critical messaging, and ActiveMQ server configurations is beyond the scope of this paper, but the reader is encouraged to visit [http://activemq.apache.org](http://activemq.apache.org/) for further information. Of important note for this paper topic is the knowledge that the authors setup and configured another Mosquitto (MQTT) server to enable distributed communications.

CASS Server Installations and Database Entries

At the time of this writing, the authors are in the process of installing a CASS Server, the code of which is accessible by following links in the CASS Developer Guide here: [http://devs.cassproject.org/index.html,](http://devs.cassproject.org/index.html) on an Amazon Web Service (AWS) Elastic Compute Cloud (EC2) instance. While the instance hosting this new CASS server was not completed at the time of this writing, the authors are in the process of finalizing the configurations and will be offering connection information to the community as an additional test CASS server based on the most-current GitHub code base.

During many programs, such as ADL’s Total Learning Architecture (TLA) initiative, various entities have interfaced with a CASS server and entered in various maturities of frameworks. One such entry, for instance, resulted in Army SMEs creating a custom framework for use in a Fort Benning experiment in late 2018. By enlisting subject matter experts to populate a CASS database with relevant competency relationships, an ITS such as GIFT becomes enabled to read and analyze these relationships to better-adapt training content tagged with similar metadata. The authors created a faux-framework for use during this paper’s efforts at https://dev.cassproject.org/api/data/schema.cassproject. org.0.3.Framework/ba049c98-0d69-4fc3-96e1- 931b90035fe3 which can be accessed in a GIFT course through a RESTful API call. CASS servers return data about frameworks and competencies in JSON formats.

By linking a GIFT Course Property menu item to the CASS database information, it became dynamically possible to populate a GIFT course that read in competencies that the course could then be linked to. For instance, a competency of “Attach Biometric Smart Clothing” <Broadens> a competency of “Sync Biometric Smart Clothing to Network.” In reverse, the 2nd competency in the previous sentence <Narrows> the 1st competency according to CASS vocabulary. Other examples of CASS relationships include <Equivalent to>, <Requires>, <Is Enabled By>, <Is Related To>, and <Desires>. Every competency can be related to every other competency within a framework, and even to other framework’s competencies as well.

The GIFT code enabling these CASS properties to be linked to a GIFT course will be made available to the public community in a following release of GIFT, and also made live in CLOUD GIFT at https://cloud.gifttutoring.org when ready for release by the development team.

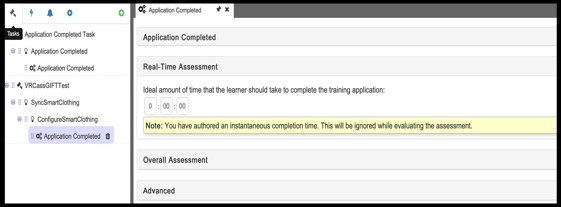
This paper’s effort concluded with a GIFT course reading information in from a CASS database and displaying it in a read-only fashion, but not storing any information permanently as exact specifications on how to integrate GIFT and CASS are still being discussed by the GIFT community.

Figure 5: GIFT Unity Course Object Assessment

The live Unity simulation was also coded to read in CASS database information dynamically over the internet, and during the scenario was able to temporarily store “competency progression” as goals in the virtual world were completed, completing the GIFT-CASS-VR distributed system proof as all communication was handled through an intermediary and agnostic Mosquitto server online at https://test.mosquitto.org/ (no 2 systems in this paper’s effort were localhost, viewing all IOT devices together as “one system”).

Virtual World Scenario in Unity

The Unity level was built primarily using existing art assets or low-cost acquisitions from the Unity Asset Store. It is recommended that if the reader wishes to integrate a Unity application with GIFT for the first time, that the WebGL version of a Unity build be used according to existing integration instructions in the GIFT documentation and Course Object meant for such interactions.

The authors treated this paper effort’s scenario as an external application, however, as demonstrating GIFT in a distributed environment was a desired outcome. The virtual world scenario can operate on any PC capable of running a basic Unity application, and communicated to the distributed Mosquitto server and the online CASS test server through the UnityWebRequest Unity library with which REST calls were made. For further information on enabling stand-alone Unity scenarios to communicate with non-local systems, the reader is encouraged to reference https://docs.unity3d.com/ScriptReference/Networking. UnityWebRequest.html.

The Unity scenario can be run in Desktop or VR modes that Unity allows for with first class citizen libraries. The scenario was also coded to display identical CASS database competency information in the Heads-Up Display (HUD). Also included in the HUD were representational self-health and squad-health biometric symbols, each of which represented the trainee in the lesson and simulated squad members, respectively. By allowing the trainee to see both the virtual world complete with buildings, non-player characters, equipment, furnishings, weather effects, while simultaneously displaying HUD information with “extrasensory” information about the environment, the authors wished to demonstrate not only virtual reality but what an augmented reality system could, in the future, begin to take the form of when integrated with GIFT.

The GIFT course operates in parallel with, and preferably launches, a virtual world scenario similar to how interfaces between GIFT and VBS are utilized. Any course training enabled through (and constrained by) virtual reality and/or any game engine can be constructed through using GIFT Course Objects and adaptive training measures. Examples include Real Time Assessments, Tasks, and Conditions that can be authored using the GIFT Authoring Tools (GAT) that respond to and inject events into the virtual environment scenario. Thanks to the open API design of both systems, training within a course topic and mix of competencies read in from CASS can be communicated from and to each of the discussed systems to great effect.

Shown below in Figure 6 is an example of how independent systems can communicate with a virtual world scenario. By listening to messages from the Mosquitto server, a simple HUD can listen for changes in self or squad health, offer visual indicators of masteries waiting to be acquired or already achieved, and offer advice to the trainee at a level the instructor or a GIFT course deems to be most-valuable to the training experience.

Figure 6: Unity HUD with CASS and Biometrics

IOT Devices and Communications

The complete setup instructions to configure an Arduino microcontroller with breadboards, custom voltages, wiring, and IOT devices is beyond the scope of this paper, but if interested, the reader is encouraged to reference the following online documents for instructions:

* Speakers:https[://w](http://www.deviceplus.com/how-)ww[.d](http://www.deviceplus.com/how-)e[viceplus.com/how-](http://www.deviceplus.com/how-) tos/arduino-guide/entry019/
* LEDs:https://[www.deviceplus.com/how-](http://www.deviceplus.com/how-) tos/arduino-guide/entry\_002/
* Long\_Range\_Communication:https://www.de viceplus.com/how-tos/arduino-guide/arduino- long-range-communication-tutorial-lorenz- shield/
* Pulse\_Sensor:https://pulsesensor.com/pages/in stalling-our-playground-for-pulsesensor- arduino

With the prototype system libraries, hardware listed in the above 2.6 Hardware section, and GIFT-CASS-VR system operational, the same network paradigm of minimal Mosquitto messaging was used to link in IOT devices to the software suite. The authors began by connecting IOT devices to each other and a microcontroller, all of which are traditionally very limited in all of power, range, “disk space,” APIs, and capabilities, and then connected these devices to network/long range communication capabilities. After integration efforts inspired largely by open source communities, the authors were able to have all IOT devices managed through a single microcontroller, report their sensor data, and respond to system messages passed-to and received-from the Mosquitto server.

While for this paper’s effort the authors only managed the simplest of message passing between the IOT devices and the rest of the system, the primary goal of proving interconnectivity of a system of systems with GIFT acting as the ITS was satisfied. In addition, communication with parallel virtual environments (eventually formally multiplayer instead of simulated) and the distributed nature of the system as a whole was satisfied as a secondary goal. The authors look forward to future discussions with the community in these areas of interest.

Team Training Perspectives

The framework described thus-far resulted in a prototype system built with multiplayer and squad perspectives in mind. By creating a system in a distributed network with no (or at the very least, reconfigurable) single point of communication failure, a system in which nodes can be stood up-or-down on demand has had its foundation formed. Using principles and paradigms of load balancing made commercially-ready from companies such as AWS, nodes such as GIFT can better recover from a downtime perspective upon degraded performance being detected (a sensor slipping off, network communications being jammed, etc.). This also means a plug-and-play smart IOT uniforms could be developed and switched out with minimal interruption, or virtual scenarios dynamically configuring themselves to detect squad data for each team member and simulating or removing entities based on scenario configuration settings. With these distributed system paradigms in mind, frameworks such as the prototype presented in this paper form the basis for managing squad training in scenarios with ITSs (GIFT) and competency systems (CASS) in an active role, with completely variable sensor data and number of active players being switched out as training needs dictate.

ENABLED EXPERIMENT FRAMEWORKS

The authors wish to provide the results of this effort back to the community as a nonprofit effort and will work with the GIFT team and community to determine the best path forward in this regard.

Some options include formal delivery of GIFT code updates to builds pushed out to CLOUD GIFT, specifically those updates relating to CASS database interactions. Other options include experimental branches that the community can request specific access to, or downloadable builds and configurations that enable capabilities based on individual or organizational needs.

As a prototype volunteer effort, the authors will work to the best of their ability to discuss any GIFT team and community interest to provide access to the experimental framework that can enable future research in the areas of:

* GIFT communication with LMS, LRS, or Competency (CASS) systems
* Competency metadata incorporation into GIFT courses
* Biometric and IOT device status monitoring in a GIFT course
* Virtual reality scenario creation and integration into a GIFT course
* Mixing 1-to-1 GIFT course and virtual world scenarios, 1 GIFT-course and single-player- team-simulation courses, and n-GIFT courses to n-virtual world scenario experiments in parallel

CONCLUSIONS AND FUTURE RESEARCH

The authors set out to perform a software engineering feasibility study that GIFT would be able to act as a centralized ITS in a decentralized system of systems. Through combining GIFT, CASS, Unity VR scenarios, and IOT devices, the authors have shown how GIFT’s open API communication protocols allow for efficient integration into larger-scale training systems. Of specific interest to the authors is future involvement with systems such as CASS that enable, among other capabilities, a system of standards with which to track relevant learner skills and metadata to allow for improved adaptive training. Combined with an LRS and/or LMS, the authors hope to continue these paper’s efforts to improve adaptive training by incorporating modern technologies and extending GIFT course functionality to include even more capability to handle simultaneous team member squad training.

REFERENCES

All references for this paper are listed in-line within the content as all references were online resources.

AUTHORS BIOGRAPHY

**Zach Heylmun:** Zach Heylmun graduated from the University of Florida with a degree in Digital Arts and Science engineering and supports SSI on the GIFT program as an as-needed consultant for the past two years. After graduation, he worked for Lockheed Martin on low-level, high performance graphics as well as virtual reality rendering for flight simulation and training. Since starting his own company, Voidstar Solutions, as well as helping to form Synaptic Sparks, a 501c3 charity dedicated to STEM education, he has worked with a wider variety of technologies. Through a combination of efforts, both for- and non-profit, he has worked on web technologies, mobile applications, and server infrastructures.

**Mike Kalaf:** Mike 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. In his most recent efforts, he has been leading new opportunities applying front end modeling, simulation and analysis. Mike has led several programs integrating “state of the art” technology and delivering highly successful technology and business innovation. Mike has been collaborating with educational organizations and exploring conceptual frameworks, platforms and business models to transform our current system and elevate the performance and quality. He is involved with the University of Central Florida’s College of Education on a unique system of teacher training via classroom simulators. These projects fit well to advance science, technology, engineering and mathematics learning to lay the groundwork for a new generation of engineers and scientists. Mike volunteers his time to numerous education organizations including serving as a board member for the Central Florida STEM council and the Seminole County Public Schools Foundation. Mike’s formal education includes an earned Mechanical Engineering degree from Rochester Institute of Technology, RIT.

**Christopher Meyer:** Christopher brings a breadth of leadership experience and technical knowledge to the team. And, most recently, Christopher has supported the GIFT program for two years under the most current contract. He received his Bachelor and Master of Science degrees in Computer Science from Kansas State University, also receiving minors in Economics and Modern Languages, and studied abroad for a year during a tour in Japan at Chukyo University dedicated to the specialized study of Artificial Intelligence. After completing traditional education phases, Chris was employed at Lockheed Martin for 10 years working hand-in-hand with representatives from the Departments of Defense, Health and Human Services, Energy, and Education to assist in the creation of solutions to solve challenges at a national level. Having now co-created his own business segment, Chris enjoys utilizing entrepreneurship, international experience, leadership knowledge, and his own engineering skills alongside his peers to advance world technology, health, and opportunity efficiently and responsibly.

**Christofer Padilla:** Christofer graduated from the University of Central Florida with a Bachelor of Science in Computer Science where he researched creating a usable concurrency library for submission to the Boost C++ libraries. He is currently employed by Dignitas Technologies, after completing a two-year internship as a developer on the GIFT program through two major releases. His contributions include supporting the development effort on the Real-Time Assessment Editor and the Course Authoring Tool. Christofer's early passion for software engineering has garnered him experience across many domains including virtual reality systems, computer graphics, web frameworks and intelligent tutoring systems.

**Lucy Woodman:** Lucy has recently graduated from Seminole State College with a Bachelor of Science in Information Technology. Lucy has supported Synaptic Sparks for one year during a successful internship and transitioned to a be a major supporter of big data services within SSI in December of 2018. Lucy is a certified Amazon Web Service specialist and is currently studying to obtain further AWS certifications in System Architecting. Lucy also supports the team by providing research and development support in new fields of network and social technology.