The GIFT 2017 Architecture Report

Keith Brawner
U.S. Army Research Laboratory
Keith.W.Brawner.civ@mail.mil

Zach Heylmun
Synaptic Sparks Incorporated
Zach@SynapticSparks.org

Michael Hoffman
Dignitas Technologies, LLC
mhoffman@dignitastechnologies.com

INTRODUCTION

The first version of the Generalized Intelligent Framework for Tutoring (GIFT) was released to the public in May of 2012. One year later, the first symposium of the GIFT user community was held at the Artificial Intelligence and Education conference in Memphis, Tennessee. Since then, the GIFT development team has continued to gather feedback from the community regarding recommendations on how the GIFT project can continue to meet the needs of the user community and beyond. This current paper continues the conversation with the GIFT user community in a few important ways. The current paper invites and encourages members of the GIFT user community to continue to share their feedback, research findings, and technology innovations with the development team and with one another in order to strengthen the power, usability, and flexibility of the GIFT project. As a follow up to the "GIFT 2015 Report Card and State of the Project" (Brawner & Ososky, 2015), and GIFT 2016 Community Report, the feature requests and responses have been broken out among a number of papers discussing research vectors. This paper discusses the ongoing architectural workings and changes in support of the various sets of projects.

The research and technology innovation efforts presented in the current document include those that are informed by the GIFT user community, and only represent a *fraction* of the overall research, development, and implementation work associated with GIFT. We invite the reader to review the other chapters in this volume, publications on GIFTTutoring.org, and other references described below, to get a sense of the total body of work on the GIFT project. Major themes in this current, 2017 GIFT community discussion include integration with wide-scale systems such as EdX and LearnSphere, further work in enhancing authoring, hosting your own experimental server on Amazon Web Services, and the first GIFT Summer Camp.

GIFT SUMMER CAMP

This year marks the first year that the GIFT Summer Camp was held. The audience is primarily constructed from among the military training and acquisition communities. A pilot program is being run for a select, invite-only, audience, initially modeled after the Pittsburgh Science of Learning Center (PSLC) Summer School. Similar to the PSLC program, the GIFT Summer Camp covers the concepts of building courses, creating adaptive instruction, and analyzing the data produced by students.

Two items were required to more fully demonstrate the features of GIFT within a dedicated course. The first of these was that an instructor needed a way to check up on students and progress. This emerging requirement necessitated the development of firmer user roles and an instructor dashboard. In order to create the course, a complete tutor on the subject of anatomy is being created from sample materials provided from the Total Army Distance Learning Center. Both of these items are expected to deploy publicly to GIFT Cloud during June of 2017.

GIFT CLOUD

It has been an open secret that the GIFT Cloud instance hosted on cloud.gifttutoring.org has not been a true cloud deployment. A real cloud deployment should have flexible hardware specifications, flexible bandwidth allocations, regular backups, redundancy, and other items. The GIFT Cloud hosted until February had been stationed on a single server computer in Orlando, Florida, subject to power outages and downtime issues. While these issues haven't affected the vast majority of users, the plan was always to move to an Amazon Web Services system. This was performed in February 2017 after some delay. The move from a desktop-based training system to a server-based training system to a cloud-based training system necessitated changes to a number of modules. These changes are described below for posterity.

User and Content Management Systems (UMS/CMS)

Among the primary features missing from a server-based version of GIFT was the ability to upload and manage content to a user account. Previously GIFT supported the idea of a 'classroom training', where a single user has access to all of the content, with no authentication required in order to run the system. The move to a server-based architecture necessitated the management of users, and the separation of content logic from filesystem logic. Moving to this architecture forced the development of both a content abstraction layer and a user abstraction layer. The abstraction layers were created so that a different CMS or UMS could be used at a later time by a developer with a specific requirement. As such, in the event that an organization would prefer the use of a different CMS/UMS, the port of GIFT to use them should be relatively straightforward through the replacement of function calls in the abstraction layer.

As part of the UMS/CMS selection, the following systems were considered: RUSSEL, Exo Platform, Fedora Commons, TYPO3, Moodle, Gooru Learning, EdX, Alfresco, Plone, Jahia, Hippo CMS, Nuxeo, LifeRay, JackRabbit, and Modeshape. Paper space limits a thorough discussion of each of these items, but a list of features and selections is available upon request. Each of these systems was examined for their ability to be a content repository, support different categories of users, differing user roles, sharing of content, built in upload/download, and backend API support. All of the above options are open source, with many used elsewhere in military educational applications. Nuxeo was eventually selected based on its well-developed backend API, Java framework, variety of deployment options, and flexible plugin architecture. Of special interest was the ability to develop plugins to support the individual needs for each Module.

Database

GIFT was initially developed with a mySQL database in place. One of the most notable problems from the first GIFT User's Meeting was the difficulty in the installation of GIFT. Specifically, the use of a mySQL database either: a) required administrator privileges, b) conflicted with a database installation of a previous program with unknown password, or c) was beyond the technical ability of the desired user. Because of these difficulties, the standard database was changed from mySQL to derby in 2013. The derby database program can be embedded within Java applications, needs no installation, and has a small footprint (~2MB), but is not fundamentally equipped to handle server-levels of transactions. The move back to the server necessitated a move to a more fully mature database. This change is invisible to the end user, as GIFT uses postgres in its cloud deployment mode for content, but continues to use derby for the downloadable distribution (both Desktop and Server configuration) and for LMS/UMS functionality as it is intended to be used for classroom-scale experimentation.

Applications

One of the key problems with developing a server-hosted application is that the link to a desktop application is broken. In order to get around this problem, the GIFT program creates a small executable which serves to interface with desktop application. The desktop application, while it does have to have an existing external interface, does not typically need any special configuration. The acceptance of the JWS application is sufficient to allow the server to control the computer-installed application and provide meaningful feedback on events. This includes embedding RapidMiner plugins and other affective modeling software, if appropriate (Rowe, Mott, & Lester, 2015). A similar approach could be taken for Sensor Module components, but has not yet been implemented.

Authentication

Previously, a GIFT version installed on a local machine has no need of authentication to provide services. However, a server-based version must provide a way to distinguish between users. The cloud version of GIFT uses the authentication provided by gifttutoring.org in order to link accounts, content, and access settings. gifttutoring.org uses the RedMine project as its backend, which uses OAuth for authentication. This authentication can be switched to another (e.g. OAuth) authentication, or with other authentication servers with relatively minimal changes to the source code.

Analysis

One of the key features of GIFT was the ability to run the Event Reporting Tool (ERT). The ERT was a tool developed to look at GIFT log files and determine what actions the student or system had taken during the course of tutoring interactions. This ability was key to fuse multiple streams of sensor data with learning interactions and system interactions in order to answer research questions such as "did the system take an action that lead to learning?" and "what was the learner state when the system made this decision?". The ERT output results in a format friendly to SPSS for easy analysis, which is also compatible with RapidMiner and most major independent toolsets. Previously, there was not a way to easily port this functionality to a server deployment. The tool has since been reinvented as the "My Research" tab, which allows for both the sharing of tutoring links and analysis of students who use the links.

DOMAIN-INDEPENDENT INSTRUCTIONAL MODELS

GIFT has been constructed in a manner to be independent to the model of the learner, domain, and instruction. As an example of this independent structure, the Domain Module has a configuration called the Domain Knowledge File (DKF), which references code implementations for assessment. Most of these code implementations inherit from an abstract base class of assessment; some of them redirect to external assessment engines, as is the case for the vMedic assessments.

At the time of writing, the Pedagogical Module, at its output, can send four types of request: *Request for Scenario Adaption*, *Request for Performance Assessment*, *Request for Instructional Intervention*, and *Request for Branch Adaption*. Each of these requests, excepting the last, is independent of the instructional model. As a concrete example, consider the following instructional models for adapting within-scenario instruction: a "drill and kill" model, an "expertise based" instructional model, and an "intensive practice" model. The "drill and kill" model runs the same scenario, with feedback, repeatedly until a passing threshold is met. The "expertise based" instructional model runs a scenario, but only provides feedback to novice users; it saves feedback until the end for expert users. The "intensive practice" model restarts after a single mistake, as is common in medical scenarios with high consequence.

Although a simple example, each of the above-discussed models is supported with the current message structure. The *Request for Branch Adaption* message, however, requires the specification of a Merrill Quadrant Enum, which is decidedly *not* independent to the instructional model. This model is being expanded to encompass a variety of instructional strategies (passive, active, constructive) within the model, but GIFT is unnecessarily tied to the Component Display Theory during authoring and runtime. This new functionality is scheduled for release.

In order to break the tie to CDT, the Brach Adaption Strategy is going to support an extra layer of abstraction that will allow current and future pedagogical models to recommend different attributes necessary for selecting content.. Additionally, during authoring time, it will be possible to select variations among instructional strategies, within boxes. In concrete terms, GIFT currently supports the "Adaptive Courseflow" Course Object for dynamic instruction based on the Engine for Management of Adaptive Pedagogy, which is based on Component Display Theory. GIFT will later support a variety of instructional policy boxes with variations on configurations. Each of these will still be tied to the concepts of instruction, and have access to the same type of information as the Adaptive Courseflow Course Object currently has. More details about these changes are discussed within the the paper which discuss Markov Decision Logic (Rowe, Pokorny, Goldberg, Mott, & Lester, 2017) and dynamic After Action Review technologies (Carlin, Brawner, Nucci, Kramer, & Oster, 2017).

ONTOLOGICAL MEDIATION AND VIRTUAL HUMAN TOOLKIT

GIFT has been designed previously as a modular system with modules are interchangeable. The approach of standardization is useful, provided that the community has come to moderate consensus of the operational parts of the system. The ITS community frequently discusses tutoring systems as a byproduct of their instruction, learner modeling, domain modeling, interface to the student, and authorability. While the authoring tools of any system are frequently tied to the product of the system (e.g. Word and .docx, PhotoShop and .psd, iMovie and .imovieproj, etc.), the modeling performed within the system can be modified and changed through the addition of plugins or extensions. The components of an Intelligent Tutoring System (ITS) which have been agreed upon by the community have been implemented as functional modules within the system. If the default Learner Module, Pedagogical Module, and Sensor module are selected, GIFT, as it currently stands, can be used to create intelligent tutoring systems without the development of plugins, without any programming experience, and without the configuration of any files.

With standardization and componentization comes limitation. Some research projects within the research portfolio conduct studies to verify that learning is occurring. Other research projects conduct studies to test different module configurations, instructional selection algorithms, feedback delivery, or other functionality which is captured within existing modular structure. Some research projects have the goal of developing *new* capabilities for a system – capabilities not necessarily agreed upon as needed by the community; capabilities not necessarily essential for the system at runtime. As example of such a capability is a data mining process for developing a policy for selection of review material after a training interaction (Carlin et al., 2016). Which module should encapsulate the data mining portion?

Efforts have been made this year to expand GIFT into an agent-, or service-based system. This change allows for the easier addition of new capabilities, especially new capabilities which fall outside of the scope of existing modules. Examples of capabilities for early integration into this structure include the Markov Decisions Process (MDP) learning from the North Carolina group, the data mining from the Aptima group, and scenario generation techniques.

The move to a service-driven, instead of module-driven, framework additionally allows for the incorporation of external utilities, such as the super Generalized Learning Utilities (superGLU) (Brawner, Goodwin, & Sottilare, 2016), discussed later in this volume. Additional services, such as those provided from the Virtual Human Toolkit (Hartholt et al., 2013) are planned to be added via this approach within the next 12 months. Services include high-quality animation of agents, non-verbal behavior generation, natural language processing, and other items. This approach addressed concerns raised during the previous GIFT Symposium involving RESTful Webservices (Goodwin et al., 2016). These efforts are not planned to be released in the next GIFT release, but will be tested for functionality in the summer of 2017. More about these developments can be requested, or researched through the matching paper in this proceedings (Nye, Auerbach, Mehta, Hartholt, & Fast, 2017).

LEARNING TECHNOLOGY INTEROPERABILITY (LTI) PLUGIN

The Learning Tools Interoperability (LTI) standard was created by the IMS Global Consortium (IMS Global Learning Consortium, 2010). The LTI standard operates on the idea of consumers, such as Moodle and EdX, and providers, such as game-based interactions and homework problems. In this model, a GIFT "course" can operate as a portion of an overall learning interaction. The GIFT course can be developed with standard GIFT tools and linked to the LTI consumer through the sharing of a few simple pieces of information: the CourseID, launchURL, Key, and Shared Secret, as shown in Figure 1. These are then imported as a part of an EdX (or other LTI) course. Data from the shared course can be analyzed with the GIFT "My Research" for the portions of the course that GIFT manages. This is new architectural functionality where none existed prior. More about this functionality can be read later in this work {Aleven, 2017 #1964}.

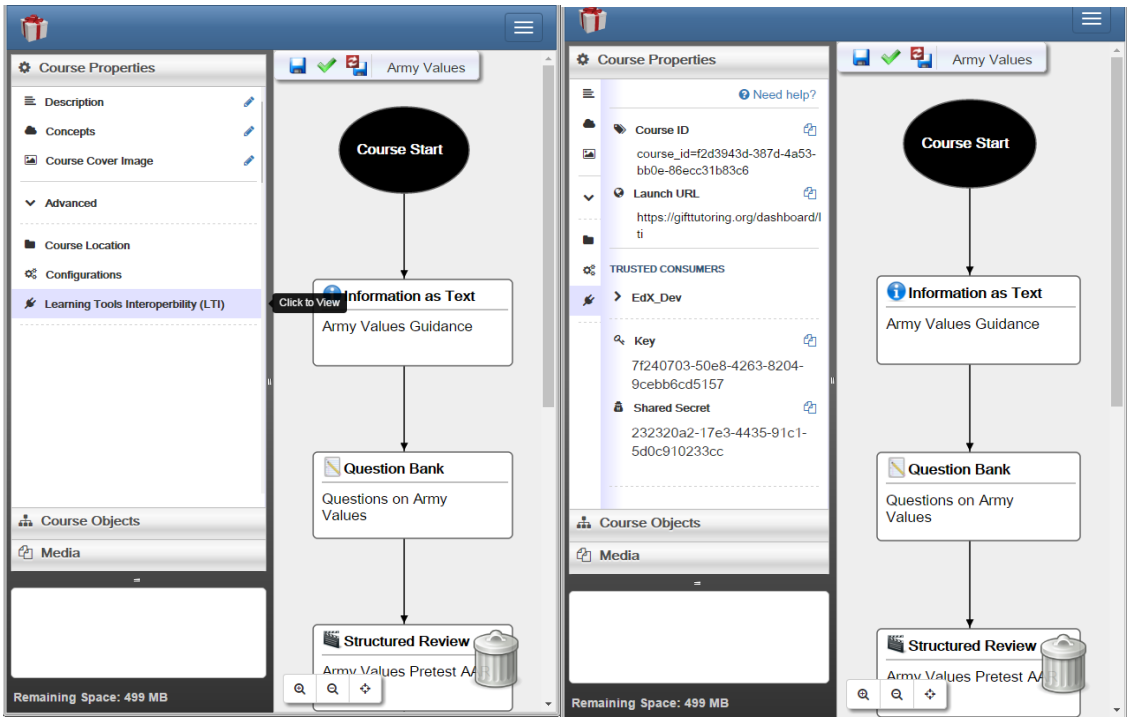

Figure 1 – LTI Interoperability Configuration UI

INCREASED REPORTING

One of the frequently-requested functions in GIFT is the embedding of experience API (xAPI) data (Brawner & Ososky, 2015; Goodwin et al., 2016; Poeppelman, Hruska, Long, & Amburn, 2014). Previously GIFT reported out xAPI data as it matched with simulation-enabled assessments. However, for the purposes of data-mining, storage, and cross-domain synchronization, increased frequency of xAPI reporting has been requested and developed. This increased xAPI reporting has been required for several reasons.

The first of these reasons is for tracking a single user across multiple sessions, such as marksmanship within a classroom, within a simulator, at a live training range, and at a qualification range. Being able to track performance across these items allows for the system to isolate the tutoring actions which not only produced learning in the tutoring system, but eventually led to the end goal of qualification.

The second of these reasons is to enable shared toolsets to make models across multiple domains of instruction, similar to the LearnSphere model. This model allows for multiple systems to store the data in a standard format, such as within a Learner Record Store. Further, reporting xAPI data frequently allows for other, external, systems to use the data, make models of learners or instruction, and share the models with the general GIFT system. For both of these reasons, GIFT now reports xAPI updates as people enter/exit Course Objects, rather than simply at the end of the lesson. Previously, GIFT reported information to the xAPI interfaces less frequently, and at larger gain size.

CONCLUSION AND FUTURE RESEARCH DIRECTIONS

The current deployment of GIFT Cloud is two-fold and split onto a developmental server (dev-cloud.gifttutoring.org) and a production server (cloud.gifttutoring.org). This allows for the rapid prototyping of features in a testable developmental environment before pushing changes cyclically to the production environment. The advantage of this is rapid deployment, while a disadvantage is that GIFT Cloud is now frequently out of sync with GIFT Desktop. Synchronized versions of GIFT Desktop with limited testing/reliability measures are available upon request, or published privately into the SVN.

Future architectural and ontological research is anticipated in a few key areas. The first of these is team tutoring, where a team tutoring version of GIFT has been used in a few private experiments. A team tutor involves a team DKF, team model (instead of learner model), and team pedagogy. Additionally, each individual may have one or more 'role' DKFs, on which they are assessed, in the event that they play multiple roles. The research understandings for each of these items is limited, the architecture functional, and authoring tools non-existent. The second area is in the further breakout of module processes into webservice function calls, as discussed above. The third area of architectural research is how the integration of new training environments can populate authoring tools to enable within-game creation of tutoring content in areas such as the GIFT Wrap project.

Finally, GIFT is intended to provide members of the training, educational, and research communities with the tools and technology needed to efficiently create, manage, and deliver adaptive tutoring content, through leveraging a flexible and extendable framework. GIFT will be continuously improved and developed for the foreseeable future. The authors would like to remind the members of the GIFT community that they have a valuable opportunity to help shape how these and other features are designed and implemented into GIFT. The GIFT development team encourages members of the GIFT community to continue to communicate feedback, issues, suggestions, and results (of research) in order to help us provide the useful tools, powerful technologies, and positive user experiences that will make adaptive tutoring technology accessible to the broadest possible audience.

REFERENCES

Brawner, K., Goodwin, G., & Sottilare, R. (2016). *Agent-Based Practices for an Intelligent Tutoring System Architecture.* Paper presented at the International Conference on Augmented Cognition.

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

Carlin, A., Brawner, K., Nucci, C., Kramer, D., & Oster, E. (2017). *Educational Data Mining using GIFT Cloud.* Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym5), Orlando, FL.

Carlin, A., Kramer, D., Nucci, C., Oster, E., Freeman, J., & Brawner, K. (2016). *An Adaptive AAR Capability for GIFT.* Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym4).

Goodwin, G., Medford, A., Diaz, G., Murphy, J., Twitchell, M., & Blink, M. J. (2016). *Integrating an Interoperable Competency Model with GIFT using the Experience API (xAPI).* Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym4).

Hartholt, A., Traum, D., Marsella, S. C., Shapiro, A., Stratou, G., Leuski, A., . . . Gratch, J. (2013). *All together now: Introducing the Virtual Human Toolkit.* Paper presented at the International Workshop on Intelligent Virtual Agents.

IMS Global Learning Consortium. (2010). IMS GLC learning tools interoperability basic LTI implementation guide.

Nye, B., Auerbach, D., Mehta, T., Hartholt, A., & Fast, E. (2017). *Building a Backbone for Multi-Agent Tutoring in GIFT.* Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym5), Orlando, FL.

Poeppelman, T., Hruska, M., Long, R., & Amburn, C. (2014). *Interoperable Performance Assessment for Individuals and Teams Using Experience API*. Paper presented at the GIFTSym2, Pittsburgh, PA.

Rowe, J., Mott, B., & Lester, J. (2015). *Opportunities and Challenges in Generalizable Sensor-Based Affect Recognition for Learning*. Paper presented at the Artificial Intelligence in Education, Madrid, Spain.

Rowe, J., Pokorny, B., Goldberg, B., Mott, B., & Lester, J. (2017). *Toward Simulated Students for Reinforcement Learning-Driven Tutorial Planning in GIFT.* Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym5), Orlando, FL.

ABOUT THE AUTHORS

Keith Brawner, PhD *is a researcher for the U. S. Army Research Laboratory's Human Research & Engineering Directorate (ARL-HRED), and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT). He has 11 years of experience within U.S. Army and Navy acquisition, development, and research agencies. He holds a Masters and PhD degree in Computer Engineering with a focus on Intelligent Systems and Machine Learning from the University of Central Florida. His current research is in ITS architectures and cognitive architectures. He manages research in adaptive training, semi/fully automated user tools for adaptive training content, and architectural programs towards next-generation training.*

Michael Hoffman *is a senior software engineer at Dignitas Technologies, and Dignitas's lead GIFT engineer. He has been responsible for ensuring that the development of GIFT meets the evolving customer requirements in addition to supporting both intelligent tutoring for computer based training and intelligent tutoring technology research of the growing user community. Michael manages and contributes support for the GIFT community through various mediums including the GIFT portal (www.GIFTTutoring.org), annual GIFT Symposium conferences, and technical exchanges with ARL.*

Zach Heylmun *graduated from the University of Florida with a degree in Digital Arts and Science Engineering. After graduation, he worked for Lockheed Martin on low-level, high performance graphics as well as virtual reality rendering for flight simulation 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 infrastructure. He is currently the technical lead for Synaptic Sparks engineering efforts for the GIFT project.