Component Interaction within the Generalized Intelligent Framework for Tutoring (GIFT) as a Model for Adaptive Instructional System Standards

Robert Sottilare [0000-0002-5278-2441] & Keith Brawner [0000-0003-4839-6887]

US Army Research Laboratory, Orlando, FL USA

{robert.a.sottilare; keith.w.brawner}.civ@mail.mil

Abstract. This paper discusses the need for adaptive instructional system (AIS) standards and suggests the Generalized Intelligent Framework for Tutoring (GIFT) as a starting point for discussing component level interaction as a potential candidate for standardization. GIFT is an open, modular architecture to support authoring, delivery, instruction, and evaluation of adaptive instruction. Adaptive instruction is usually delivered by Intelligent Tutoring Systems (ITSs) and like most ITSs is composed of four basic components: a learner model, an instructional model, a domain model, and an interface model. We are suggesting that the data exchanged between these four models (that are in the form of messages in GIFT) are candidates for standardization in that they solve the problem of interoperability while simultaneously allowing for flexibility of form and function within each of the common components. This paper examines the type and form of GIFT messages and makes a case for their consideration as an initial starting place for AIS standards for interoperability.

Keywords: Adaptive Instructional Systems (AISs), Domain Models, Generalized Intelligent Framework for Tutoring (GIFT), Intelligent Tutoring Systems (ITSs), Instructional Models, Learner Models, User Interfaces

1 Introduction

Adaptive instructional systems (AISs) use human variability and other learner/team attributes along with instructional conditions to develop/select appropriate strategies (domain-independent policies) and tactics (actions). The goal of adaptive instruction is to optimize learning, performance, retention, and the transfer of skills between the training environment and the work or operational environment where the skills learned during training are to be applied. Adaptive instruction is usually delivered and managed by an Intelligent Tutoring System (ITS), computer-based technology that "aims to provide immediate and customized instruction or feedback to learners, usually without requiring intervention from a human teacher" [1], but could provide learning experiences through intelligent media. Sottilare & Brawner defined AISs as "*computer-based systems that guide learning experiences by tailoring instruction and recommendations* *based on the goals, needs, and preferences of each learner in the context of domain learning objectives*" [2]. So what is the motivation to standardize AISs and why start with GIFT?

In December 2017, the IEEE Learning Technologies Steering Committee (LTSC) formed a 6-month Standards Study Group to investigate the possible market need for standards across AISs. A recent AIS standards workshop in Orlando, Florida highlighted several problems related to the authoring and maintenance of AISs that could be resolved by improving the interoperability of AIS components. The Generalized Intelligent Framework for Tutoring (GIFT) is an open-source, modular architecture used to author, manage instruction, and evaluate the effect of adaptive instructional technologies (tools and methods) in a variety of training and educational domains [3, 4]. This paper focuses on standard practices adopted in the design of GIFT that might be considered useful as models for future AIS standardization.

ITSs, as a subset of AISs, have four commonly mentioned components in the literature: a model of the learner, the instruction, the domain, and the tutor-learner interface [5], but it might be more complicated than we first assume. In GIFT, the domain model has many subcomponents including content for developing knowledge, tests for assessing knowledge, practice environments for applying knowledge and developing skill, and instructional interventions and their triggers based on learner behaviors [4]. Other ITSs have similarly complex configurations. So how will we ever get to a standard that will be beneficial (e.g., save time, reduce skill required) and be widely used? We discuss standards through design goals and functional modularity in the next section.

2 Toward Standardization through Design Goals

Devedzic, Radovic, and Jerinic [6] mention several desirable design features for ITSs, but three are specifically relevant to our argument to begin with GIFT as a model for standardization and include the ability to:

- easily assemble new ITSs from existing and pretested software components;
- easily replace any ITS software component with a logically and functionally similar component without degrading the performance of the rest of the system;
- logically organize and catalogue software components in a repository for future use

One way to meet each of these goals is to build functional modularity into the ITS architecture. GIFT does this by defining modules, software components with a primary purpose, the ability to process data received from other modules, and share derived measures with the rest of the architecture. GIFT operates on an Apache ActiveMQ network (Figure 1). Common properties shared by all GIFT modules are: module name, ActiveMQ URL, message encoding type (e.g., JSON), ignore IP address allocation (true/false flag), and start XML Rpc Python Server (true/false flag), port and class name.

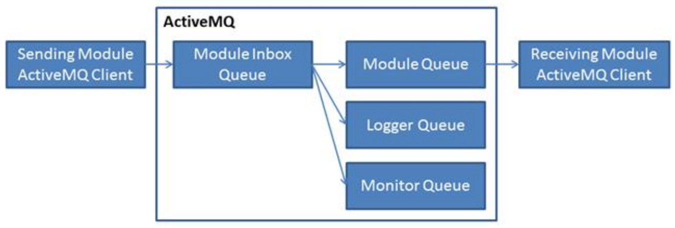


Fig. 5. Message Flow through Active MQ Network (as used in GIFT)

The modules in GIFT include the four primary components (gray) common to most ITSs and some ancillary functions as noted below and shown in Figure 2.

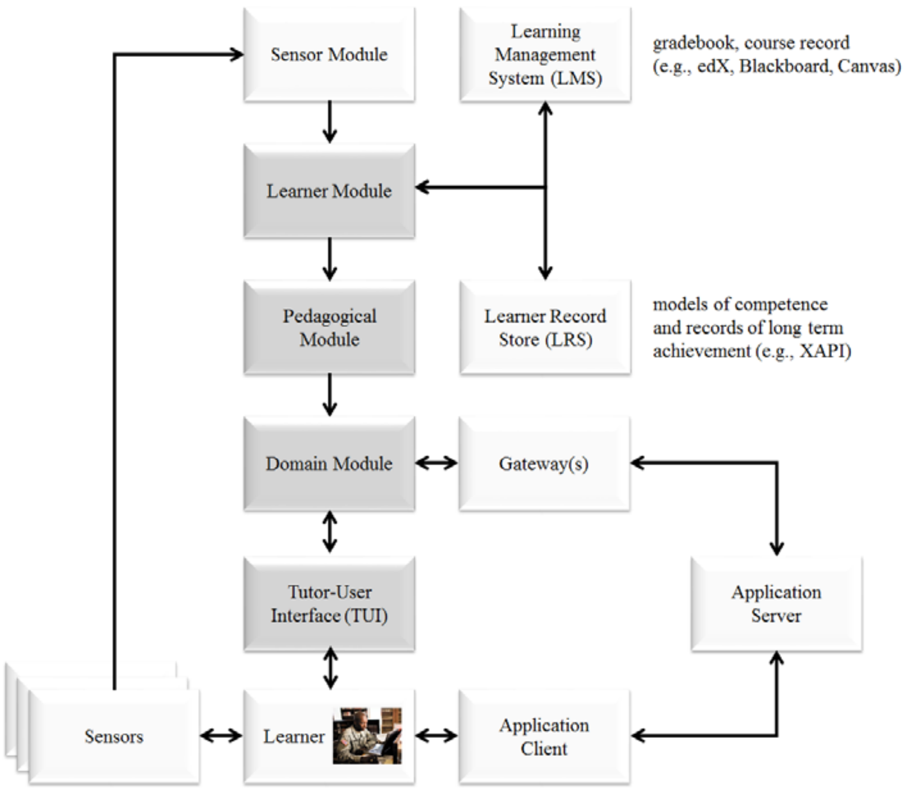

Sensor Module
Learning Management System (LMS)
gradebook, course record (e.g., edX, Blackboard, Canvas)
Learner Module
Pedagogical Module
Learner Record Store (LRS)
models of competence and records of long term achievement (e.g., XAPI)
Domain Module
Gateway(s)
Tutor-User Interface (TUI)
Application Server
Sensors
Learner
Application Client

Fig. 6. General message flow through GIFT where an ActiveMQ network is the backbone of each message connection.

As an example, a message from the Learner Module to the Pedagogical Module is really a message from the Learner Module to the ActiveMQ backbone and routing, which is then routed to the Pedagogical Module. This separates modules from systems; modules

Back to Table of Contents

can each be run as their own webservice, combined into one with their own message routing, or alternative network topologies. With the exception of the messages to/from the Trainee, each of the message communication lines represents an ActiveMQ backbone connection for routing and delivery.

2.1 User Management System (UMS) Module

The primary function of the UMS module is to manage a user session. It is responsible for storing information about the user such as biographical details, in addition to maintaining information about domain sessions. It does not, however, keep scoring records of user's training history. That is handled by the LMS. The UMS also contains the message logger which is responsible for logging all messages sent on a single ActiveMQ network. Functionally, the UMS represents the management of a user session for the remainder of the system.

2.2 Learning Management System (LMS) Module

The primary function of the LMS module is keep track of a learner or team's instructional experiences and achievements as a history of learning. The GIFT LMS saves the scores of every assessment during every lesson experienced in GIFT. Functionally, the LMS represents the longer-term storage of learner data. In recent GIFT developments, the LMS has received its data from a Learner Record Store (LRS), rather than a transaction database, representing the ease of flexibility between data sources while using the same communication standards throughout the rest of the system.

2.3 Learner Module

The primary function of the Learner module is to determine the learner's state (e.g., real-time performance, real-time emotional, or long term domain competency). It might do this by reading domain competency state from a learner record store or by receiving learner states the Sensor Module which interprets states from learner data. Naturally, the function of an individual module is not specified in a specification; only the data inputs and outputs are specified. The Learner Module takes in data about the learner and processes it into assessments of the learner.

2.4 Sensor Module

The primary function of the sensor module is to read and filter sensor data to determine/predict learner states. While this is broken out separately form the learner module, it could easily be made a sub-function of the learner module. The sensor module was broken out separately from the learner module for the following reasons which in turn have made it easier to integrate new sensors into GIFT:

- Sensors are optional, rather than required, for the majority of learner tasks.

Back to Table of Contents

- Physical hardware sensors frequently have specific configuration needs, such as Bluetooth synchronization, EEG electrode placements, or proprietary interfaces.
- Software sensors are frequently tied to the implementation of a specific training system, such as gaze detection within driving simulators.

2.5 Pedagogical (Instructional) Module

The primary function of the pedagogical module is to use information about the learner's state to generate recommendations (e.g., next course to take) and select instructional strategies (e.g., prompt learner to reflect) to enhance learning. Instructional strategies are passed to the domain module for implementation.

2.6 Tutor Module

The primary function of the Tutor module is to provide an interface that allows interaction between GIFT and the learner. Often referred to as the tutor-user interface (TUI), this is not a formal module. The reason for this is so all of the functions of a tutor module, such as adaptations to a scenario or problems, speech of an avatar, or other functions may be performed within a specified training environment. GIFT, however, provides a default implementation which allows for characters, learner inputs, and other baseline functionality for systems which do not have the capability to respond to instructional tactics. Examples of such systems include PowerPoint for feedback delivery, or custom actions within a Distributed Interactive Simulation (DIS; IEEE 1278) compatible simulator.

2.7 Gateway Module

The primary function of the Gateway module is to interface with external environments (e.g., game-based simulations). GIFT listens for external communications and then converts the information into GIFT messages for consumption by GIFT and vice-versa. When a message is received from outside of GIFT (e.g., Virtual BattleSpace Distributed Interactive Simulation (VBS DIS) connection), the Gateway module converts that message into a GIFT message and multicasts the message to appropriate GIFT modules. The Gateway Module has simulation interoperability interfaces with the following standards/products: DIS, VBS, Augmented REality Sandtable (ARES), Microsoft PowerPoint, Tactical Combat Casualty Care (TC3)/Virtual Medic, SCATT Pro Marksman Training Application. The above list illustrates the ease of integrating new simulations with GIFT, rather than intending to be an all-inclusive list of GIFT integrations.

2.8 Domain Module

The primary function of the domain module is to create, maintain and assess domain sessions. This module hosts or points to content used during instruction and contains a domain course file which is an XML file containing information needed to assess the

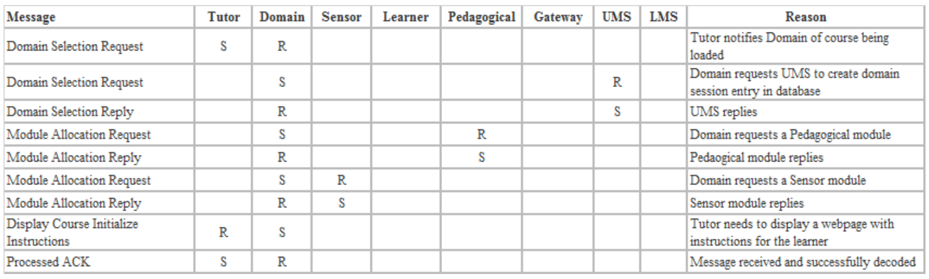
Back to Table of Contents

learner’s progress toward proficiency for the concepts (learning objectives) identified by the course author.

3 Potential Standard Messages

The interaction between the modules described in the previous section provide a view of potential standard messages for some initialization functions in GIFT as shown in Table 1 where S = send and R = Receive.

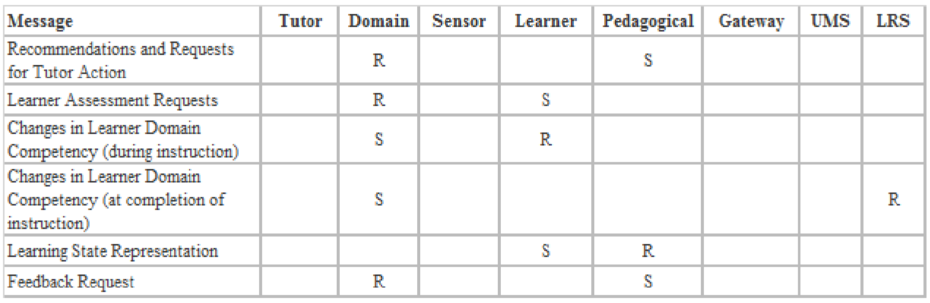
Table 6. A Sampling of Initialization Messages for Standardization in AISs



Message	Tutor	Domain	Sensor	Learner	Pedagogical	Gateway	UMS	LMS	Reason
Domain Selection Request	S	R							Tutor notifies Domain of course being loaded
Domain Selection Request		S					R		Domain requests UMS to create domain session entry in database
Domain Selection Reply		R					S		UMS replies
Module Allocation Request		S			R				Domain requests a Pedagogical module
Module Allocation Reply		R			S				Pedaogical module replies
Module Allocation Request		S	R						Domain requests a Sensor module
Module Allocation Reply		R	S						Sensor module replies
Display Course Initialize Instructions	R	S							Tutor needs to display a webpage with instructions for the learner
Processed ACK	S	R							Message received and successfully decoded

The list shown in Table 2 is adapted from Sottilare & Brawner [2] and the GIFT software documentation to show real-time interaction between modules during instruction. Note that acknowledgments to requests were left out of this table for simplicity.

Table 7. A Sampling of Real-Time Instructional Messages for Standardization in AISs



Message	Tutor	Domain	Sensor	Learner	Pedagogical	Gateway	UMS	LRS
Recommendations and Requests for Tutor Action		R			S			
Learner Assessment Requests		R		S				
Changes in Learner Domain Competency (during instruction)		S		R				
Changes in Learner Domain Competency (at completion of instruction)		S						R
Learning State Representation				S	R			
Feedback Request		R			S			

4 Discussion

We argue that the messages between the GIFT modules mentioned above are a starting point for standardization discussions. The core modules in GIFT are the Learner, Domain, Pedagogical, and Gateway. Further, they should accept and/or communicate be-

Back to Table of Contents

tween the more optional modules of the UMS, LMS, Sensor, and Tutor, dividing functionality further towards systems which may not require significant adaptation or modeling.

These modules have been used across many functional systems, within many studies, and for applied training needs. While the exact format of the messages described in this paper may or may not work as a standard message set for AISs, the types of messages are significant. AISs will require messages that can support instructional initialization, instructional management (including real-time assessment and feedback), and automated after-action review (AAR). The adoption of standardized messages will allow AIS designers and authors to address the design goals laid out by Devedzic et al [6] and promote interoperability of components at the module level.

Additional consideration should be given to the development of community-based interface models [7] that can be developed to integrate external simulation environments and then be shared/reused. As noted earlier (section 2.7 of this paper), GIFT has several external environments for which interfaces have been constructed to the GIFT gateway. GIFT also has structure course objects which represent a variety of content types and forms (e.g., text, static and dynamic media). These allow GIFT authors to drag and drop standard objects that can be configured to pull in content unique to that domain.

Finally, we should consider the impact that team modeling will have on design goals of AISs and particularly the interoperability and reuse components and models for team task domains. It is highly likely that GIFT and any other tutoring architecture that ventures into the team tutoring space will require new models to represent the team and its learning objectives. It is worth noting that structurally similar, data-driven approaches to team tutoring will offer enhanced opportunities for interoperability between other team tutors, and that teamwork, how team members communicate and cooperate, may be the best opportunity for measures and assessments to be generalized across team domains [8].

Readers interested in helping shape AIS standards can obtain more information and participate at www.instructionalsciences.org, or through the IEEE LTSC.

Acknowledgments

A portion of the research described herein has been sponsored by the U.S. Army Research Laboratory. Statements and opinions expressed in this paper do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References

1. Psotka, Joseph, Leonard Daniel Massey, and Sharon A. Mutter, eds. Intelligent tutoring systems: Lessons learned. *Psychology Press*, 1988.

Back to Table of Contents

2. Sottilare, R. & Brawner, K. (2018, March). Exploring Standardization Opportunities by Examining Interaction between Common Adaptive Instructional System Components. In Proceedings of the *First Adaptive Instructional Systems (AIS) Standards Workshop*, Orlando, Florida.
3. Sottilare, R.A., Brawner, K.W., Goldberg, B.S. & Holden, H.K. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). Concept paper released as part of GIFT software documentation. Orlando, FL: *US Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED).* Retrieved from: https://gifttutoring.org/attachments/152/GIFTDescription_0.pdf
4. Sottilare, R., Brawner, K., Sinatra, A. & Johnston, J. (2017). An Updated Concept for a Generalized Intelligent Framework for Tutoring (GIFT). Orlando, FL: *US Army Research Laboratory*. May 2017. DOI: 10.13140/RG.2.2.12941.54244.
5. Sottilare, R., Graesser, A., Hu, X., & Holden, H. (2013). Preface. Design Recommendations for Intelligent Tutoring Systems: Volume 1- Learner Modeling. *US Army Research Laboratory*, Orlando, Florida. ISBN 978-0-9893923-0-3.
6. Devedzic V, Radovic D, Jerinic L. On the notion of components for intelligent tutoring systems. In *International Conference on Intelligent Tutoring Systems* 1998 Aug 16 (pp. 504-513). Springer, Berlin, Heidelberg.
7. Sottilare, R. (2018, July, in press). Community Models to Enhance Adaptive Instruction. In Foundations of Augmented Cognition (pp. TBD). *Springer International Publishing*.
8. Sottilare, R. A., Burke, C. S., Salas, E., Sinatra, A. M., Johnston, J. H., & Gilbert, S. B. (2017). Designing adaptive instruction for teams: A meta-analysis. *International Journal of Artificial Intelligence in Education*, 1-40.8.

Back to Table of Contents