Learning Ecosystems Using the Generalized Intelligent Framework for Tutoring (GIFT) and the Experience API (xAPI)

Michael Hruska[1], Ashley Medford[1], Jennifer Murphy[2]

[1] Problem Solutions 1407 Eisenower Blvd., Johnstown, PA 15904
[2] Quantum Improvements Consulting 812 Oak Chase Dr., Orlando, FL 32828

Abstract. Learning ecosystems provide a combination of technologies and support resources available to help individuals learn within an environment [1]. The Experience API (xAPI) is an enabling specification for learning ecosystems, which provides a method for producing interoperable data that can be shared within a learning ecosystem [2]. Version 4.1 of the Generalized Intelligent Framework for Tutoring (GIFT) provides support to both produce and consume xAPI data. A number of use cases are enabled by this support. This paper will explore the use cases, functionality enabled, setup and design guidance in addition to exploring practical applications for using GIFT and xAPI within learning ecosystems.

Keywords: adaptation, Experience API, intelligent tutoring systems, learning, xAPI, GIFT, computer-based tutoring systems, learning ecosystems

1 Introduction

Organizations in the U.S. alone invested approximately $164.2 billion on employee training and development in 2012 [3], and in 2013, an average of over $1,200 per employee was spent for direct learning [4]. With 38% of this training being delivered using technology [4], this investment is increasingly being spent on non-traditional training methods and technologies. As learning ecosystems continue to grow in complexity, so too do the challenges faced by education and training professionals.

Personalizing education and assessing student learning are grand, educational challenges being faced today [5]. Recent efforts on learning ecosystems reflect this movement towards adaptive and tailored learning [5,6]. In general, the goal in a learning ecosystem is to leverage performance data in order to assess and adapt learning and in turn, increase training effectiveness and lower associated training time and costs [6]. By capturing the massive amount of learning data tied to each individual and bound within a learning ecosystem, the ability to meet these educational challenges by intelligently tailoring learning and assessing performance is possible.

Research and development efforts by the Advanced Distributed Learning (ADL) initiative of the Department of Defense (DoD) and the U.S. Army's Research Laboratory (ARL) are striving to meet these complex challenges. The Experience API specification (xAPI), developed by ADL, provides an interoperable means to describe and track learning in various learning ecosystem components [7]. ARL's work on interoperability of performance data and intelligent tutors, specifically the Generalized Intelligent Framework for Tutoring (GIFT), along with xAPI provide a basis for this paper. The use of xAPI in conjunction with intelligent tutoring (e.g., GIFT) permits the creation of a reference architecture and provides functionality for a number of use cases. Installation and configuration of open source software components enable testing and experimentation around these use cases. This paper outlines the technical information, reference architecture, use cases, configuration, and expected behaviors of the technology components surrounding this work.

1.1 Existing Efforts

The ARL effort on Interoperable Performance Assessment (IPA) focuses on uniformly defining and describing learning experiences [8]. IPA defines methods for encoding human performance data using xAPI statements [9]. The goal of such encoding is to create data with *inter-system data value* to support adaptation in learning ecosystems. Additionally, interoperable encoding can provide rich data analytics and visualizations.

ARL's IPA research works primarily toward the goal of defining uniform performance measures in simulation and providing summative assessments towards these measures from multiple sources. Additional IPA efforts, focused on using small group and team data, also indicate the potential of such approaches to adapt and even drive team formation [10]. Overall, IPA efforts aim to address the following use cases: show a historical view of proficiency; show a live view of performance; enable macro and micro training adaptation, and; collect Big Data for trends analysis.

1.2 Experience API and Learning Ecosystems

The xAPI is a supporting specification for learning ecosystems. The xAPI specification defines an interface for a common and interoperable data store for xAPI statements, known as a Learning Record Store (LRS). The LRS provides a single storage point in a learning ecosystem. Systems within a learning ecosystem either act as a "producer" of xAPI statements or as a "consumer". [7]

1.3 The Generalized Intelligent Frameworks for Tutoring (GIFT)

GIFT, developed by ARL's Human Research and Engineering Directorate (HRED), provides a service-oriented framework of tools, methods and standards to make it easier to author computer-based tutoring systems (CBTS), manage instruction, and assess the effect of CBTS, components and methodologies [11]. GIFT was enhanced to interoperate with xAPI in Version 3.02 to provide a *consumer* functionality and in version 4.1 to provide *producer* functionality.

1.4 Reference Architecture

The Figure below (Fig. 1) shows a reference architecture for a learning ecosystem using GIFT. The architecture is composed of components that might comprise the learning ecosystem like a Learning Management System (LMS), a Simulator, GIFT, and other systems such as games or virtual worlds. In this example, any system would be capable of producing xAPI statements. Further, GIFT provides both consumer and producer functionality as it (a) produces xAPI statements for other elements in the ecosystem and (b) consumes xAPI statements. [12]

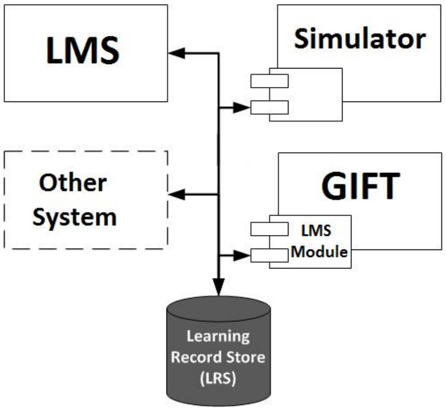


Fig. 1. A reference architecture for a learning ecosystem is shown [12]. The architecture shows a *Learning Record Store (LRS)* where data is stored and retrieved by elements of the ecosystem. A simulator or other system may produce or consume data from the LRS. *GIFT* uses the *LMS / LRS Module*, which is enabled to both produce and consume xAPI data via the *LRS*.

1.5 Use Cases

A number of use cases for learning ecosystems are supported by GIFT and its xAPI functionality. GIFT may be used in conjunction with an LRS and other systems to demonstrate and test these use cases. The following are some potential use cases that may be built upon GIFT and the xAPI functionality:

1. **Multiple System Performance Assessment** – Multiple systems including live scenarios using observer based tools, simulations, LMS, and games can be used to assess performance and produce xAPI data. This data can be used to drive adaptation as GIFT acts as a consumer.
2. **Using Simulation for Assessment –** A simulation may be used for performance assessment. The simulation produces xAPI data. This data may also be used to drive adaptation as GIFT acts as a consumer.
3. **GIFT-Driven Data Production –** xAPI data about course content and concepts contained within a course can be created and stored.
4. **Macro Adaptation –** GIFT can provide macro adaptation or outer loop adaptation based upon the data it consumes. Performance deficiencies produced by GIFT or other systems that are stored as xAPI data can be used to intelligently navigate or recommend courses or other learning experiences.
5. **Inter-system Driven Micro-Adaptation –** GIFT can provide micro-adaptation within a scenario based upon data it consumes from other systems.

2 Using GIFT and xAPI

GIFT (Version 4.1) is capable of both producing and consuming xAPI statements. Minimal configuration is required to setup this functionality in GIFT. Version 4.1 natively supports use cases 1, 2, 3, and 4 outlined in Section 1.5. Additional programming related to content development is required to support use case 5.

2.1 GIFT LMS/LRS Module

The LMS module within GIFT, responsible for retrieving and storing training and assessment history, enables xAPI support. The LMS module has been enhanced by creating an LRS submodule within which it allows both polling of and writing to the LRS.

2.2 Setting up GIFT with xAPI Support

In order to enable xAPI functionality for GIFT, an LRS must be available and connected to the network which GIFT is installed on. The following steps need to be completed to enable xAPI support in GIFT:

1. Install GIFT framework (refer to www.gifttutoring.org)
2. Install an LRS (see below)
3. Configure GIFT to communicate with the LRS end point

Several open source LRS options exist as well as commercial options. The following open source LRS solutions are currently available:

- Open source LRS from ADL - https://github.com/adlnet/ADL_LRS
- Hosted LRS from ADL- https://lrs.adlnet.gov/xapi/
- Open source LRS from learning locker - http://learninglocker.net/

2.2.1 Configuration of xAPI End Point

Once GIFT and the LRS are installed, GIFT must be configured to communicate with the LRS endpoint. The following steps must be undertaken to allow GIFT to communicate with the LRS:

1. Open the *LMSConnections.xml* file located in the `<GIFT Root>\GIFT\config\lms` directory
2. Select edit, and add a new connection entry under the `<LMSConnections>` root using the following information format and entering *the `username`, `password`, and `URL` for the LRS installation* between the XML elements:

```
<Connection>
 <enabled>true</enabled>
 <impl>lms.impl.Lrs</impl>
 <name>LRS Name</name>
 <Parameters>
   <networkAddress>https://lrs.url</networkAddress>
   <username>username</username>
   <password>password</password>
 </Parameters>
</Connection>
```

2.3 GIFT as a Producer of Interoperable Data

Once configured, GIFT is enabled to act as a producer of xAPI data. As a producer, once a training scenario is completed, the course records and scores are passed to the LMS module for storage. This data is then passed to the LMS database as well as the LRS sub-module. An xAPI statement is generated for each level of the graded score nodes, and each statement is linked to their parent statement. The figure (Fig. 2.) below outlines an example of data that is created and defined for the elements in the xAPI format.

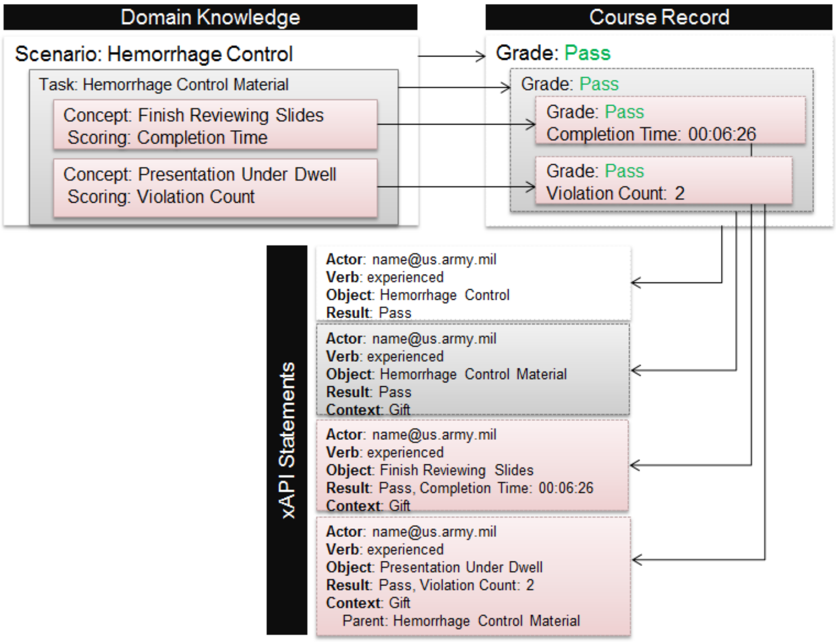

Domain Knowledge
Scenario: Hemorrhage Control
Task: Hemorrhage Control Material
Concept: Finish Reviewing Slides
Scoring: Completion Time
Concept: Presentation Under Dwell
Scoring: Violation Count
Course Record
Grade: Pass
Grade: Pass
Grade: Pass
Completion Time: 00:06:26
Grade: Pass
Violation Count: 2
xAPI Statements
Actor: name@us.army.mil
Verb: experienced
Object: Hemorrhage Control
Result: Pass
Actor: name@us.army.mil
Verb: experienced
Object: Hemorrhage Control Material
Result: Pass
Context: Gift
Actor: name@us.army.mil
Verb: experienced
Object: Finish Reviewing Slides
Result: Pass, Completion Time: 00:06:26
Context: Gift
Actor: name@us.army.mil
Verb: experienced
Object: Presentation Under Dwell
Result: Pass, Violation Count: 2
Context: Gift
Parent: Hemorrhage Control Material

Fig 2. An example of data from a *Domain Knowledge File, Course Record,* and *xAPI Statements* is shown. The example outlines the scenario, tasks, concept, and grades that are used to define the xAPI data elements. [12]

2.3.1 Editing Domain Knowledge File

In order for GIFT to produce xAPI data, the concepts that are represented within a course must be added to the XML file that represents the course. The following steps must be taken to update the file:

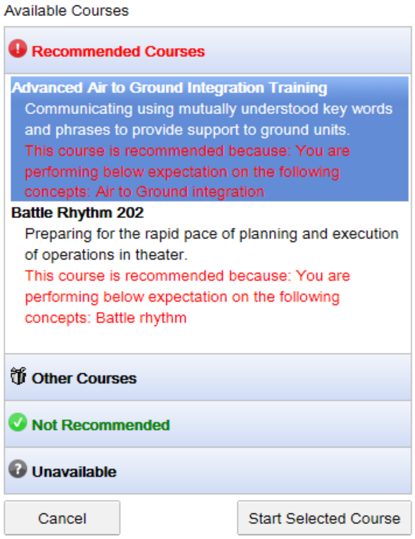
1. Edit the XML file for the course located at `<GIFT Root>\Domain`
2. Add a `<concepts>` section under the `<Course>` root. Below is an example of the addition of the `<concepts>` elements:

```
<Course name="Course Example" …>
 …<concepts>
    <concept>Skill 1</concept>
    <concept>Skill 2</concept>
```

```
   <concept>Skill 3</concept>
 </concepts> …
</Course>
```

2.4 GIFT as a Consumer of Interoperable Data

The LMS module of GIFT also provides consumer functionality. The consumer function allows GIFT, via the LRS submodule, to poll the LRS end point. xAPI statements are used to extend GIFT's course suggestion capabilities. The LMS polling function gets a user's history, using their email address as an identifier, when the user logs into GIFT. The LMS module examines available course metadata definitions to find courses with concepts that match the user's deficiencies. The LMS module then recommends concepts matching deficiencies noted in xAPI statements for which the user is "below" concept proficiency. Dynamic filtering of course suggestions is presented through the "Recommended Courses" (See Fig 3).



Advanced Air to Ground Integration Training
Communicating using mutually understood key words and phrases to provide support to ground units.
This course is recommended because: You are performing below expectation on the following concepts: Air to Ground integration

Battle Rhythm 202
Preparing for the rapid pace of planning and execution of operations in theater.
This course is recommended because: You are performing below expectation on the following concepts: Battle rhythm

Fig 3. A screen shot of GIFT *Available Courses* is shown. The example outlines recommended courses as determined by the LMS module by examining course metadata and deficiencies stored in xAPI statements within the LRS. [12]

3 Conclusions

GIFT allows enhanced functionality via its LMS module to integrate external data sources in a learning ecosystem. GIFT also enables data created within GIFT to be stored in an interoperable way that supports learning ecosystems via xAPI in an LRS. This functionality enables GIFT and other systems to evaluate incoming student competencies in order to better inform instructional strategy. Systems in the learning ecosystem are also enabled to make recommendations for the next training events based on performance data. Using this functionality, researchers may test a number of different use cases and functions of adaptive learning in learning ecosystems. Usage of xAPI data in learning ecosystems with GIFT and other producers will allow consumers in learning ecosystems to assess and tailor learning and ultimately, to leverage Big Data analytics to discover trends over time.

References

1. Kelly, D.: What Is a Learning Ecosystem? The eLearning Guild, http://twist.elearningguild.net/2013/11/what-is-a-learning-ecosystem/ (2013)
2. Advanced Distributed Learning: xAPI Specification. GitHub, https://github.com/adlnet/xAPI-Spec (2015)
3. American Society for Training & Development: 2013 State of the Industry. ASTD DBA Association for Talent Development (ATD), Alexandria (2013)
4. Association for Talent Development: 2014 State of the Industry. ASTD DBA Association for Talent Development (ATD), Alexandria (2014)
5. Woolf, B.P: A Roadmap for Education Technology. National Science Foundation #0637190 (2010)
6. Foreman, S. & Rosenburg, M. J.: Learning and Performance Ecosystems: Strategy, Technology, Impact, and Challenges. Santa Rosa: The eLearning Guild, http://www.elearningguild.com (2014)
7. Advanced Distributed Learning: Training and Learning Architecture (TLA): Experience API (xAPI), http://www.adlnet.gov/tla/experience-api (2014)
8. Poeppelman, T., Ayers, J., Hruska, Long, R., Amburn, C., Bink, M.: Interoperable Performance Assessment using the Experience API. In: The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), vol. 2013. (2013)
9. Hruska, M., et al. (2013)
10. Hruska, M., Long, R., Amburn, C., Kilcullen, T., Poeppelman, T.: Experience API and Team Evaluation: Evolving Interoperable Performance Assessment. In: The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), vol. 2014. (2014)
11. Sottilare, R.A., Brawner, K.W., Goldberg, B.S., & Holden, H.K.: The Generalized Intelligent Framework for Tutoring (GIFT). U.S. Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED), Orlando (2012)
12. Hruska, M., Long, R., Amburn, C.: Human Performance Interoperability via xAPI: Current Military Outreach Efforts. In: Fall Simulation Interoperability Workshop, 14F-SIW-035, Orlando (2014)