Adaptive Course Flow and Sequencing through the Engine for Management of Adaptive Pedagogy (EMAP)

Benjamin Goldberg[1] and Michael Hoffman[2]

[1]U.S. Army Research Laboratory, Orlando, FL
[2]Dignitas Technologies, Inc., Orlando, FL
benjamin.s.goldberg.civ@mail.mil;
mhoffman@dignitastechnologies.com

Abstract. The Engine for Management of Adaptive Pedagogy (EMAP) is the Generalized Intelligent Framework for Tutoring's (GIFT) first implementation of a domain-independent pedagogical manager. It establishes a framework within GIFT that adheres to sound instructional system design, while also providing tools and methods to create highly personalized and adaptive learning experiences. In this paper, we present the components of the EMAP, we highlight their utility when authoring an EMAP managed lesson, and we review the limitations associated with its first instantiation.

Keywords: Adaptive Instruction; Pedagogical Model; Instructional Management; Engine for Management of Adaptive Pedagogy; Generalized Intelligent Framework for Tutoring

1 Introduction

The Generalized Intelligent Framework for Tutoring (GIFT) is being developed as a domain-agnostic solution to authoring, delivering, and evaluating adaptive training solutions across an array of domains and training applications. While GIFT's initial development focused on establishing a standardized architecture for building Intelligent Tutoring System (ITS) functions to support distributed learning events, recent work has centered on extending the adaptive capabilities the framework affords. As a result, the Engine for Management of Adaptive Pedagogy (EMAP) was developed. The EMAP is based on an extensive literature review of instructional strategy focused research within computer-based training [3], and organizes its findings in a domain-independent fashion. At the moment, there are papers that highlight the literature and theory that fed the EMAPs design [3, 4] and that highlight the authoring tools and processes required for implementing its functions [5], but there is nothing that reviews EMAP interactions from the learner's perspective as it relates to event sequencing. In this paper, we present a usecase of a GIFT lesson managed by the EMAP and we review the various architectural components that make it run. We will first highlight the work that went into formalizing the EMAP, the dependencies the EMAP has with other portions of the GIFT architecture, and we present a usecase of lesson interaction and transitions managed by EMAP logic and configurations.

2 Formalizing the EMAP

The EMAP design was the resulting outcome of a collaborative project between the U.S. Army Research Laboratory (ARL) and the Institute for Simulation and Training (IST) at the University of Central Florida. Following an extensive literature review, the team selected David Merrill's Component Display Theory (CDT) as the theoretical framework to structure EMAP requirements around [3,5].

The CDT was conceptually integrated within GIFT as a domain-agnostic framework used for course construction and building guidance/remediation configurations [3]. This requires linking learner relevant information with generalized descriptors of learning content and instructional techniques, strategies and tactics. These relationships were used to establish an initial decision tree that informed real-time adaptations.

It is important to highlight the current attributes represented in a GIFT learner model and their relationship with metadata used to describe learning content. As these variables moderate EMAP configurations that are set and adapted at run-time, it is important to review how each level of data operates and what decisions they inform. For learner model data forms, these include determinations for knowledge states, skill states, affective states, and individualized traits that have been empirically found to impact learning and retention.

2.1 Learner Model Dependencies

The EMAP uses pedagogical configurations that are moderated by attributes being tracked in GIFT's learner model. These configurations are coupled to the customized value ranges of available variables supported within the architecture's standardized schema. The configurations implemented in the EMAP are based on both historical and real-time inferences across the various trait and state attribute spaces. As such, the EMAP uses information on prior knowledge along with a set of trait characteristics to personalize lesson materials across the CDT's four quadrants (i.e., Rules, Examples, Recall, and Practice) upfront, and then uses real-time assessment information on knowledge, skill, and affective states to moderate guidance, remediation, and problem selection. The goal is to establish generalized configurations that can translate across different domain spaces and varying training platforms and applications.

For knowledge and skill states, performance is monitored at an objective level. In the latest release, GIFT tracks individual learners across a hierarchy of concepts as they relate to a set of tasks within a specified domain. These concepts are established in the Domain Knowledge File (DKF), where bottom level sub-concepts (i.e., leaf nodes) are assessed against data made available by the training application itself. For each concept and set of sub-concepts, there are currently four possible state determinations: (1) above-expectation, (2) at-expectation, (3) below-expectation, and (4) unknown. Each of these representations can be associated with either a knowledge state or skill state, where this division is used to differentiate 'knowledge' from 'ability to execute'. This falls in line with the mention of Knowledge/Skills/Abilities (KSAs) defined in most doctrine and helps to make competency badging within a domain more granular. Inference procedures are performed across all concepts to determine a competency level for the domain of instruction, with values being entered as Novice, Journeyman, or Expert.

Variables based on traits found to impact learning are of importance to the EMAP. The individual traits of a learner are believed to be more stable over time and are used to set initial configurations of a lesson based on these associations. Current EMAP logic informed by traits includes motivation, self-regulatory ability, and grit. These items are not inherently tracked in the DKF, but they are used offline to configure lesson materials and sequencing when a lesson is initialized.

In terms of affect represented within GIFT's learner model, these state spaces associate primarily with data made available through sensor technologies that monitor both physiological and behavioral data sources. Affective states of interest include engagement, frustration, boredom, confusion, etc. Regardless of the state space, GIFT is very flexible with respect to affective modeling, as the researcher and/or training developer has the ability to configure what variables to track and what classifiers to apply. These classifiers are used to produce a state determination that is represented in GIFT's DKF across short-term, long-term, and predicted values. For adaptation purposes, much of the affect related information is used to adapt instruction during runtime, as this form of assessment provides insight into a learner's reactive tendencies to an event or interaction.

2.2 Metadata Dependencies

Learner model attributes are linked with generic content descriptors that the EMAP is designed to act on. This metadata is used to take domain-independent representations of pedagogical practice and associate it with domain-specific content. The metadata currently in use is based on the Learning Object Metadata (LOM [6]) standard put in place by the Institute for Electrical and Electronics Engineers (IEEE). This provides a set of high level categories (e.g., interactivity type, difficulty, skill level, coverage, etc.) and value ranges (i.e., skill level is broken down into novice, journeyman, and expert) that inform characteristics for a type of interaction. GIFT uses two authoring processes to build the EMAP linkages. First, a lesson developer needs to build metadata files for all associated content and practice materials. Next, the lesson developer must establish what learner model attributes moderate metadata selection, and what value ranges serve as strategy selection thresholds.

2.3 EMAP Course Flow Example

The following use case represents the interaction of GIFT transitions across lesson elements and materials. Each event is described in relation to the EMAP and the type of data that informs its application. The usecase is broken down by learner login and course selection; pre-lesson learner model updates and assessments; adaptive lesson delivery via a Merrill's branching; and After Action Review (AAR) and lesson completion.

Learner Login and Course Selection. When a learner interacts with GIFT to initialize a course or lesson, they are first required to login using associated IDs and passwords. Once logged in, the first function GIFT performs is checking for long-term learner model information, such as records of prior training events and any persistent trait variables being stored over time (this latter function is currently being developed). Presently, all prior training events are stored under experience Application Programming Interface (xAPI) specifications within a designated Learner Record Store (LRS) [1]. Out of the box GIFT isn't configured to use an LRS, just the SQL database we have been using for years. However the GIFT in the cloud instance will be configured to use the ADL LRS (but even that clears data out every day or so). No matter if the data is stored in either place, GIFT makes use of that information. Information related to courses taken along with performance outcomes on a concept by concept level are communicated. This information is used to recommend courses based on if any prior training events resulted in below-expectation outcomes. This is the current role xAPI plays in this process. We expect this capability to become more robust over time. Following this update, a learner is then able to select a course from GIFT's Tutor User Interface (TUI). Following this update, a learner has the ability to select their course and progress into the first transitions of a lesson.

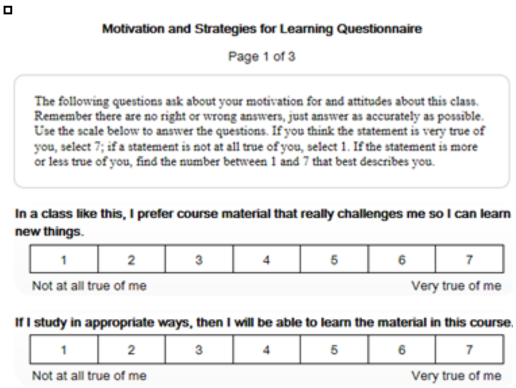

Fig. 1. GIFT Survey Interface

Pre-Lesson Updates and Assessments. Upon course initialization, GIFT references the EMAPs pedagogical configuration file to determine the trait-based variables that moderates adaptations to the lesson structure. In the current baseline, these variables include motivation, prior knowledge, self-regulatory ability, and grit. Other variables such as skill and goal-orientation can also be applied, which is the current case when a learner enters a practice quadrant of the CDT. A lesson developer has the ability to select which variables to moderate their lesson adaptations around, which impacts the first transitions experienced by a user in a new lesson. GIFT will first check an individual's persistent long-term learner model to identify any existing data. If no record is located, GIFT will administer an available survey to collect that information. This interaction is authored in GIFT's Survey Authoring System and is presented directly to the learner on the TUI (see Figure 1). Scoring rules are associated with all administered instruments, which are then used to update learner model attribute values in real-time.

GIFT then establishes learner knowledge and skill states based on associated xAPI data that exists for that domain. If no data is available, then knowledge and skill attributes are set to 'Novice'. Next, if a lesson pre-knowledge assessment is made available by the lesson developer, then the test is presented to the learner through GIFT's TUI. Based on established scoring conditions for that assessment, the learner model is updated accordingly to reflect new predicted competency levels. This information is used to bypass lesson materials on concepts that the learner has exhibited expert understanding of. Bypassing concepts is dependent on the separation of concepts not only in how they are sequenced in the course.xml but also in the content presented. i.e. if there is only 1 piece of content that covers A+B, how can either one be skipped and not the other?

Adaptive Lesson Delivery via Merrill's Branching. Once all trait-based information has been established in the learner model and all pre-test assessments have been administered, a learner is then progressed into the adaptive lesson deliver through a set of pre-defined Merrill's Branching points. This entails customized sequencing through the CDT quadrants. This interaction will be outlined through the following collection of bullet points.

- *Rules and Examples Quadrants*: Configure material around defined concepts being instructed and known attributes of the learner that match entries within the EMAP's decision tree
  - Attributes
    - Knowledge; Motivation; Self-Regulatory Ability; Grit
  - Proposed Assessments
    - Affective State: monitor learner to assess emotional and cognitive reactions
    - Behavior: monitor behavior within learning environment to assess gaming behaviors
  - No knowledge/skill updates in learner model will occur within these quadrants
- *Recall Quadrant* (Knowledge Assessment):
  - If a bank of questions for this concept has been authored within the SAS, then deliver randomized recall assessment based on EMAP configuration (configuration is defined within GIFT's Course Authoring Tool; see Goldberg et al., 2015)
    - If established scoring conditions exist, then update learner model based on assessment outcomes
      - Assumption: Only cognitive knowledge is updated based on performance outcomes within a survey delivered assessment within the recall quadrant
  - Guidance Configuration (currently being developed)

- Use known attributes of the learner to configure timing and specificity dimensions
  - Question by Question Feedback vs. Following All Items
    - Attributes that may dictate this decision: Knowledge and Self-Regulatory Ability
  - General to Specific vs. Specific to General Feedback
    - Attributes that may dictate this decision: Knowledge and Grit

– Remediation
  - If learner is reported at 'below expectation'/'at expectation' on any items (i.e. concepts), then initiate remediation loop within the defined Merrill Branch
    - Remediation path is dependent on reported cognitive knowledge state based on defined scoring logic in the Course Authoring Tool
      - For each concept:
        - If learner is scored at 'below expectation' based on scoring configuration, select that concept for Rule quadrant remediation
        - If learner is scored at 'at expectation' based on scoring configuration, select that concept for Example quadrant remediation (can be in addition to Rule quadrant remediation)
      - If there is any concept remediation needed, present the Rule remediation for all identified concepts followed by Example remediation.
        - This is where the metadata selection algorithm is used to select different content to deliver to the learner (if available).
  - Remediation ends back in Recall Quadrant
    - If items report at 'below expectation' again and there is no new content to present; then allow the learner to select the quadrant they prefer to remediate in (currently being developed).

– If all items in the Recall Assessment are reported at 'above-expectation' then move onto Practice.

- If no questions exist for the concepts within the SAS or the author removed the recall quadrant from the branch, then move onto Practice (not currently supported).
- *Practice Quadrant* (Skill Assessment):
  - If no practice has been authored/configured, and the Recall Quadrant has been satisfied, then move onto next transition in the course file
  - If a training environment/scenario has been configured, then deliver practice materials through pre-established Gateway and DKF
  - Configure material around known attributes of the learner that match entries within the EMAP's decision tree (to be developed)
    - Attributes
      - Skill; Motivation; Self-Regulatory Ability; Grit; Goal-Orientation
    - Proposed Assessments
      - Affective State: monitor learner to assess emotional and cognitive reaction
      - Behavior: monitor learning environment to assess gaming behaviors
      - Skill: monitor performance in real-time across all identified sub-concepts based on pre-defined assessments authored around Evidence Centered Design (Stealth Assessment; [2])

        - Using established scoring conditions, update learner model based on assessment outcomes
        - Assumption: Only cognitive skill is updated based on performance outcomes within a practice environment
        - A survey authored in the SAS can also be defined as a practice environment (currently being developed).
- Guidance Configuration (currently being developed)
    - Use known attributes of the learner to configure timing and specificity dimensions
        - Number of violations before triggering guidance/feedback
            - Attributes that may dictate this decision: Skill and Self-Regulatory Ability
        - General to Specific vs. Specific to General Feedback
            - Attributes that may dictate this decision: Skill and Grit
        - Static (text or audio alone) vs. interactive (AutoTutor reflection)
- Remediation
    - If learner is reported at 'below expectation'/'at expectation' on any items, then initiate remediation loop within the defined Merrill Branch
        - Remediation path is dependent on a combination of skill and knowledge
            - If learner is novice in skill and expert in knowledge, then re-initialize practice
            - If learner is novice in skill and journeyman in knowledge, then navigate to examples quadrant
    - Remediation ends back in Recall Quadrant (currently being developed)
        - If items report at 'below expectation' again and there is no new content, then allow the learner to select the quadrant they prefer to remediate in
- If all items in the Practice Assessment are reported at 'above-expectation' then move onto next transition in the course file

This sequence of interaction will occur for all identified Merrill's Branching points authored. For instance, in a lesson that instructs across four concepts, an author can decide to break up the material across two branching points. Regardless of the number of Merrill's Branching points, once all exit criteria has been reached, then the lesson transitions into post-test assessments, after-action review and lesson completion.

Post-Lesson Assessment, After Action Review, and Lesson Completion. Upon completion of all adaptive lesson transitions across the designated Merrill's Branch points, a course developer will have the ability to administer a post-knowledge and/or post-skill assessment as a means for determining overall competency levels following lesson interventions. These interactions are intended to be void of guidance functions to determine how learners perform on their own. The outcomes are used to establish final score and attribute values for a lesson, with future development offering extended remediation events.

Assessment exercises are followed by a GIFT managed AAR used for reflective and summarization practices. It is during this interaction that a student is directed to reflect on the experience of the instructional event and their resulting performance outcomes. GIFT's current AAR capability is a web-page that reviews the objectives and concepts of a lesson taken, along with recorded performance measures for all items. A goal is to provide an interactive AAR function that utilizes technology to engage a learner in reflective exercises. Following execution of the AAR transition, the EMAP managed GIFT course is complete. At this instance, GIFT communicated xAPI data for the purposes of updating the LRS with outcomes values of knowledge and skill attributes for all concepts and sub-concepts scored. The learner is then given the option to logout of the system, or to select a new course or lesson to complete.

3 Conclusion

In this paper we presented a use case of a conceptual course flow for a GIFT lesson managed by the EMAP. We highlighted architectural dependencies associated with building out an EMAP lesson and we reviewed logic associated with lesson transitions. This paper highlights the EMAP's function at the lesson level, where you can see the various decisions being made and the type of data informing its strategy selection. Enhancements to the EMAP continue, with current developmental plans looking at personalized feedback delivery options. In addition, the authoring process is being converted to web-based interfaces. For an overview of the current authoring process and to see the underlying features of the tools and methods put in place to support a pedagogical model like the EMAP, see [5] for a nice breakdown.

References

1. Poeppelman, T., Hruska, M., Long, R., Amburn, C. *Interoperable Performance Assessment for Individuals and Teams Using Experience API.* Paper presented at the Second Annual GIFT Users Symposium, Orlando, FL. 2015.
2. Shute, V. J., Kim, Y. J. Formative and stealth assessment *Handbook of research on educational communications and technology* (pp. 311-321): Springer. 2014.
3. Wang-Costello, J., Goldberg, B., Tarr, R. W., Cintron, L. M., Jiang, H. *Creating an Advanced Pedagogical Model to Improve Intelligent Tutoring Technologies.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC). Orlando, FL. 2013.
4. Goldberg, B., Brawner, K. W., Sottilare, R., Tarr, R., Billings, D. R., Malone, N. *Use of Evidence-based Strategies to Enhance the Extensibility of Adaptive Tutoring Technologies.* Paper presented at the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2012, Orlando, FL. 2012.
5. Goldberg, B., Hoffman, M., Tarr, R. Authoring Instructional Management Logic in GIFT Using the Engine for Management of Adaptive Pedagogy (EMAP). In R. Sottilare, A. Graesser, X. Hu & K. Brawner (Eds.), *Design Recommendations for Intelligent Tutoring Systems: Authoring Tools (Volume 3)*: U.S. Army Research Laboratory. 2015.
6. Mitchell, J. L., Farha, N. Learning Object Metadata: Use and Discovery. In K. Harman & A. Koohang (Eds.), *Learning Objects: Standards, Metadata, Repositories, and LCMS* (pp. 1-40). Santa Rosa, CA: Informing Science Press. 2007.