SAND2017-8377C

Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2016

Total Learning Architecture (TLA) Enables Next-generation Learning via Meta-adaptation

Jeremiah T. Folsom-Kovarik
Soar Technology, Inc.
Orlando, FL
jeremiah@soartech.com

Elaine M. Raybourn
Sandia National Laboratories* / ADL
Orlando, FL
emraybo@sandia.gov, elaine.raybourn@adlnet.gov

ABSTRACT

Technology is becoming ever more central to teaching and training. In classrooms, students use intelligent tutors and adaptive tests instead of textbooks and worksheets. In daily life, mobile devices enable blended, on-demand and ubiquitous life-long learning applications. Connected, pervasive media enables compelling transmedia learning (Raybourn, 2014) experiences. However, learning opportunities are still often implemented in “walled gardens” or stand-alone technology systems that must be manually curated and coordinated with all the others through costly, one-off efforts of developers or individual instructors.

Next-generation learning refers to a vision for breaking down technical barriers between different learning technologies so that learners and instructors can transition seamlessly between them, increasing usability and impact on learning. The Advanced Distributed Learning (ADL) Initiative proposes an open-source set of specifications together called the Total Learning Architecture (TLA) to achieve this vision. TLA enables technologies to interoperate by sharing data about learners and content, mixing media and delivery methods as context changes, and sequencing recommendations (Regan, Raybourn, & Durlach, 2013). The initial reference implementation demonstrates multiple systems interacting via the TLA in an adaptive training use case for cyber operators.

In the present paper, we describe recommendations for achieving new learning opportunities via *meta-adaptation*, or recommendations for more adaptive learning experiences that cross technical boundaries and connect systems, making each one more effective than it would be alone. Through system interoperability and meta-adaptation, the TLA facilitates an ecosystem of technologies that can work together to enhance their impact on learning.

ABOUT THE AUTHORS

J.T. Folsom-Kovarik is a Lead Scientist with Soar Technology, Inc. in the Intelligent Training research group. His research focuses on making intelligent automation capable and robust with advanced planning, user modeling, and contextual interpretation approaches to make technology meet individuals' needs. In learning systems, J.T. researches computer representations and algorithms that make adaptive training more effective via capabilities such as automation of instructor tasks that previously required laborious attention or technical knowledge engineering, and planning ahead under uncertainty in order to construct learning sequences that are more effective than individual choices. A natural continuation of his work is research into machine learning and planning approaches that extend across multiple learning systems to efficiently build a cohesive learning experience.

Elaine Raybourn, Ph.D. is a Principal Member of the Technical Staff in Cognitive Science and Systems in the Center for Computing Research at Sandia National Laboratories. She has a background in the design of games, immersive transmedia storytelling to support informal learning, social computing, and next generation learners’ interactions with future learning technology such as personal assistants for learning (PAL). She is an ERCIM (European Research Consortium for Informatics and Mathematics) Fellow and has worked in research laboratories in Germany, UK, and France. Elaine speaks regularly on the topics of games, transmedia learning, and next generation learning ecosystems. She is on assignment to Advanced Distributed Learning Initiative, Office of the Deputy Assistant Secretary of Defense (Readiness), where she explores the future of learning science & technology and fuels her passion for the design of culturally-aware, next generation transmedia learning ecosystems.

Total Learning Architecture (TLA) Enables Next-generation Learning via Meta-adaptation

Jeremiah T. Folsom-Kovarik
Soar Technology, Inc.
Orlando, FL
jeremiah@soartech.com

Elaine M. Raybourn
Sandia National Laboratories* / ADL
Orlando, FL
emraybo@sandia.gov, elaine.raybourn@adlnet.gov

INTRODUCTION

According to the U.S. Navy's Sailor 2025 Initiative, there is a strong operational need for *ready, relevant learning* that continues outside of schoolhouses, whenever it is most relevant throughout each individual's operational career (Greenert, 2011; Richardson, 2016). The goal to make individual Sailors and the Navy as a whole better trained and more ready, at lower cost, is possible explicitly because of modularized learning content that can be reused and repurposed, that can assess and track Sailor skill progression, and that can be tailored to meet different individuals' needs. Similar directives are reflected across the DoD (Neller, 2016; U.S. Department of the Air Force, 2015; U.S. Department of the Army, 2011). All of these requirements suggest the new research and development that can be explored if we unify the content, assessment, and tailoring that currently reside in many different systems in such a way that the whole, shared performance becomes greater than the sum of its parts.

Advances in technical capabilities and the learning sciences present new challenges and opportunities. The frequency, acceptance, and pervasiveness of learner contact with technology outside the classroom and after schoolhouse instruction have increased in recent years. Today, a *learning system* (or computing system that delivers learning experiences to individuals, teams, and groups) can be more than a stand-alone intelligent tutor or computer-delivered courseware (Raybourn, 2014). Instead, learning systems can appear on different devices and be coordinated to create *distributed learning* opportunities (DODI 1322.26, 2006) that extend outside of the classroom or stand-alone solutions and into more aspects of a learner's everyday life (Raybourn, 2007, 2013). Transmedia learning is one example, defined as the scalable system of messages representing a narrative or core experience that unfolds across multiple media and leverages the unique properties of each medium to encourage learning (Raybourn, 2014). Quite often today, informal learning takes place on demand through always-available systems such as YouTube and Wikipedia. In a broad view, blended learning classes, LVC training, and on-the-job performance monitoring constitute working examples of distributed learning systems.

The Advanced Distributed Learning (ADL) Initiative was created by Presidential order in 1999 to help the United States Department of Defense craft a vision for tailored, anytime, anywhere learning, and provide research and end-user facilitation to help reach that vision. In this role, ADL has promulgated standards such as the Sharable Content Object Reference Model (SCORM) that enable content modularity, interoperation, and reuse across the DoD and civilian learning institutions (Dodds & Fletcher, 2004). Today, Service labs and research communities continue to pursue the goal of adaptive, ready, relevant training. Where in the past the emphasis was on sharing content, ADL's research thrust now lies in sharing learner analytics across systems.

To this end ADL designed a high-level structure called the *Total Learning Architecture* (TLA) to unify diverse and disparate learning systems (Regan et al., 2013; there called the Training and Learning Architecture). TLA specifies a set of data models, development patterns, and interfaces (APIs) that anyone can implement in whole or in part. Enabling individual learning systems with TLA concepts lets them share vital data and engage in coordinated action. A key vehicle for data sharing to date is the *Experience API* (also xAPI; see second subsection below). xAPI is part of the TLA specification which has been adopted by DoD and others for the purpose of describing what individual learners experience at a fine-grained level. The fine granularity of communication enables unprecedented learner analytics and understanding across learning systems. The TLA effort we describe in the present paper builds upon the xAPI specification and further adds functionality relevant to sharing learner data across more learning topics, facilitating ever more comprehensive interoperation.

The key takeaways of the present paper for the I/ITSEC community are as follows: 1) TLA offers a modular ecosystem where many developers and research approaches may work together, 2) TLA supports existing and new learning systems, even if they can adapt learning internally, and 3) TLA overcomes technical challenges introduced by one-off integrations.

TECHNICAL CHALLENGES TO READY, RELEVANT LEARNING

We describe the array of learning systems one institution chooses to make available as that institution's *learning solution*. There is an ongoing research challenge of unifying the proliferating systems within each learning solution via shared data and coordinated action. As a result, we posit that a general and reusable approach to this research challenge will yield improved understanding of learners and improved learning effectiveness for the unified systems.

Consider a hypothetical situation. A vehicle has evolved to maintain two different crew training simulators. Each simulator is appropriate for a different revision of the vehicle. The two simulators are examples of learning systems, and together with systems for tracking learner attendance and scores they make up one learning solution for training vehicle crews. However, several problems arise because the two simulators were developed separately and do not effectively share information or interoperate. For example:

- Learners who transfer between vehicle types are required to repeat training because their learning record does not transfer across learning systems.
- Instructors must coordinate and manually transfer scores between simulators and record systems.
- Instructional designers who create and assign training may not freely choose between simulators for non-specialized content (perhaps assigning the one with greater fidelity, or the one with more availability).
- Researchers are not able to easily compare or aggregate data collected in the two simulators, because the parallels that exist are not explicit.
- Developers of the two training simulations do not have a technically easy way to begin sharing learner data. They can accomplish a one-off integration, but if they do, then that work may not be reused whenever one of the two simulators changes or a third system comes online.

This example helps make concrete some of the costs that motivate a general architecture for integrating learning systems. Across Defense, K-16, and other learning communities, similar situations are apparent. Learning Management Systems (LMSs) commonly maintain instructor-created content and individual student records. One could imagine an LMS that lets a teacher create slides and homework assignments for their class, but that content is tantalizingly restricted from another instructor at a school with a different LMS. Furthermore, change is needed to enable repurposing the same content to deliver exciting new perspectives such as blended or flipped classrooms (Baepler, Walker, & Driessen, 2014), social learning (McLoughlin & Lee, 2010), and on-the-job training that might be delivered just-in-time (Coppus et al., 2007). These examples indicate some of the potential that could be attained by freeing learning content within a particular system for greater reuse and making it available to a wider audience *without* removing it from the learning system that owns it.

A need exists to keep learning content within its native simulator, LMS, or other learning system, but also make it available through that system to a wider audience of learners under more diverse circumstances. Specifying a common method for sharing data and switching between learning systems enables this objective. Keeping content within its native system retains system-specific functionality and protections such as licenses and subscriptions. The approach also maintains continuity with legacy SCORM LMSs (Poltrack, 2014). Because it enables diverse systems to fill appropriate niches in the learning needs of an individual, we call a solution built from effectively interoperating learning systems a *learning ecosystem* (Siegfried & van den Berg, 2015).

Subsequent sections describe how a learning ecosystem approach enables new research opportunities. TLA describes open, publicly available APIs with the aim to help build a learning ecosystem. TLA also offers example software implementations for reference and reuse, but does not impose requirements on implementation or how a learning system makes use of the integration features it offers. Since TLA-enabled components are instantiated locally within an institution, data sharing can be locally controlled and need not be made public. Also, an institution may rearrange components to provide a range of functions and meet its particular learning use cases. As a result, the TLA design supports straightforward, scalable interoperation in an upgradeable and customizable ecosystem.

Need for Meta-adaptation

Within an ecosystem that contains several learning systems, a method is needed to match learning systems with the immediate needs of a learner. Once multiple systems are sharing data and working together toward one goal, it becomes possible to identify learning goals or circumstances that may make a particular system well suited. The capability to identify differences in how learning systems address learner needs enables *meta-adaptation*. Meta-adaptation (Folsom-Kovarik, Jones, & Schmorrow, 2016; Nye, 2016) refers to adapting the learning experience by selecting between learning systems or directing a change that crosses boundaries between learning systems. In order to enable precise and timely meta-adaptation, a need exists for fine-grained and comprehensive sharing of learner data from different systems. xAPI and related learning data specifications enable meta-adaptation in the TLA.

In the fields of intelligent tutoring and adaptive training, micro-adaptation might tailor the feedback a learner sees in response to an error while macro-adaptation might select the next course content that is most appropriate for the learner (Shute & Towle, 2003). Micro-adaptation refers to fine-grained responsiveness to learner actions, for example providing an immediate hint after a learner makes an error during a simulation or providing delayed, but still responsively prioritized, after-action review of the performance. Macro-adaptation refers to more sweeping changes that affect the broad course of the learning experience, for example choosing between an introductory video or an advanced practice based on how well the learner has mastered a skill to date, or choosing when to move from training that skill to another one because the target mastery has been reached.

While TLA might provide new information to influence micro-adaptation or macro-adaptation within a learning system, great research potential and learning value may lie in enabling adaptive recommendations external to a system, or meta-adaptation. For example, a meta-adaptive response to a learner error might be an immediate recommendation to pause the simulation for a moment and view a remedial document on a handheld device. Neither the simulator nor the document viewing software need special programming to know about the other. Meta-adaptation thus represents a decision made from a perspective outside any one learning system.

Frequently related to micro-adaptation and macro-adaptation is the distinction between inner-loop and outer-loop adaptation (VanLehn, 2006). The terms imply (but do not require) a view of adaptation wherein control over the learning experience is handed back and forth between the fast inner loop and slower outer loop. In theory, meta-adaptation does not need to impose a third and even less responsive type of loop. Instead, it may someday be carried out as more of a software interrupt, suggesting adaptations can occur at any time during learning. This may become possible if meta-adaptation is triggered from outside a learning system by information about the learner experience.

In summary, we argue that distributed learning in current practice often takes place in disconnected walled gardens, which restrict by design the amount and types of data available to each system and therefore limit the ability to tailor or adapt learning experiences. This challenge impacts Government, industry, national lab, and university stakeholders in our community, in roles such as developers, researchers, instructors, and most importantly, learners who do not receive a maximally tailored and coherent learning experience. However, we have argued in this section that there is an opportunity for improved impact on learning if these systems from different developers can work together. The learner can seamlessly move from one learning system to another, and interact with the particular system out of all those available that is most needed at any given moment.

AN ARCHITECTURE AND A LEARNING ECOSYSTEM APPROACH

In order to research a learning ecosystem approach to interoperation, we developed the shared APIs and models of the current TLA. The TLA offers learning systems developers an agreed-upon *lingua franca* supporting communication and interoperation as follows. First, TLA builds on the messaging model of xAPI and defines how learning systems communicate about additional topics in addition to learner experience. As part of communication, TLA specifications define methods for easily discovering other learning systems, sending and responding to queries to one or several systems, and storing or retrieving data in one or several stores. Second, TLA-enabled specialized components facilitate translating data between learning systems without requiring developers to change their internal data, and recommending adaptation including adapting across systems. The specialized components make use of data communicated in TLA by automating certain tasks of knowledge engineering and signal interpretation.

In its role of specifying communication, the TLA does not constrain how components internally process data, it only helps to ensure they understand each other when they communicate it. The authors like to say informally that the TLA "provides the wires in your solution diagram, not the boxes." The boxes, or components, can each be developed independently by different institutions. They may interoperate, and they may be interchanged to achieve different desired functionality. As such, the vision of TLA development is not to produce only one recommended way to assess learners, or one recommended way to select adaptations. Instead, the TLA specifications create an overall ecosystem for many such products to both compete and effectively work together in solving real problems. The learning ecosystem approach represents a philosophical choice.

Typically, an institution that uses the TLA to unify learning will instantiate TLA-enabled learning systems on their own servers. Their learning solution will only contain the learning systems desired at that particular institution. In this way, each organization that integrates TLA specifications becomes a metaphorical ecosystem where entities can coexist or compete to fill a niche. Learning systems are not expected to encompass every possible learning interaction in a monolithic manner, because even those systems that provide only few functions may be called on when their functions are needed. Philosophically, it may be that putting components on an equal footing via the TLA will benefit institutions by letting them select the particular components that best meet their functional need with reduced concern of technical lock-in because all offerors comply or can be tested for compliance with the TLA technical standards.

Approach to Meta-adaptation Using TLA

An organization using TLA-enabled learning systems is perhaps uniquely well suited to explore meta-adaptation because of the fine-grained learner knowledge afforded by xAPI and the stream of learner experience. It is the xAPI stream that gives TLA the ability to bring together information about not only a learner's coarse-grained properties such as mastery estimates, but also the moment-to-moment descriptions of what the learner is seeing, attempting, and accomplishing. The detail of the xAPI stream builds a clear picture of a learner's progress and lets every component, not only the currently active learning system, recognize right away when an opportunity for instructional intervention should arise.

An API (Application Programming Interface) defines standard computer instructions and data structures that let programs make requests and share data in a universally understood manner. Within TLA, xAPI is the API that is most fully mature and widely adopted. The xAPI specification, officially version 1.0.0 as of April 26, 2013, defines how to track, store, and expose fine-grained data about individual actions and interactions a learner has with a learning system (ADL Initiative, 2016a). Data collected via the xAPI can be exposed for assessment, statistical analyses, data mining, traditional reports, and data sharing. The data in xAPI is fine-grained in that this standard lets learning systems describe what an individual learner is doing from moment to moment, rather than summarizing only at the end of a long interaction. As an example, a simulator for training gunnery tasks could report outcomes of individual exercises testing different firing conditions as each one is completed, rather than simply recording a final score (Long et al., 2015). More information about xAPI is in (ADL Initiative, 2016a) and (ADL Initiative, 2016b).

By leveraging the existing work of xAPI design and specifications, meta-adaptation within a learning ecosystem has the potential to create amplifying or new effects that would be impossible with a single system. Meta-adaptation can introduce new users into a learning system from outside paths, exactly when the system is relevant and likely to make a positive impression on users. By sharing data between systems, meta-adaptation also may work to reduce the cold-start problem created when a learning system does not know enough about a learner to provide effective adaptation internally. Meta-adaptation can also help get learners unstuck and quickly return them to productive learning in the original system. In these ways, meta-adaptive interactions can benefit learning systems that allow them, even when those systems might seem to surrender some control of the learning experience.

Adding meta-adaptation by participating in a learning ecosystem does not mean that individual systems that carry out adaptation internally are not needed or will be overridden. Instead, the TLA as a whole provides adaptive systems more learner analytics, adaptation suggestions (not overrides), and other information with which systems may carry out the beneficial adaptation they already accomplish.

AN EXPERIMENTAL IMPLEMENTATION OF THE TLA APPROACH

Figure 1 represents a high-level diagram of an implementation that uses TLA specifications to integrate several learning systems and carry out meta-adaptation to choose between them.

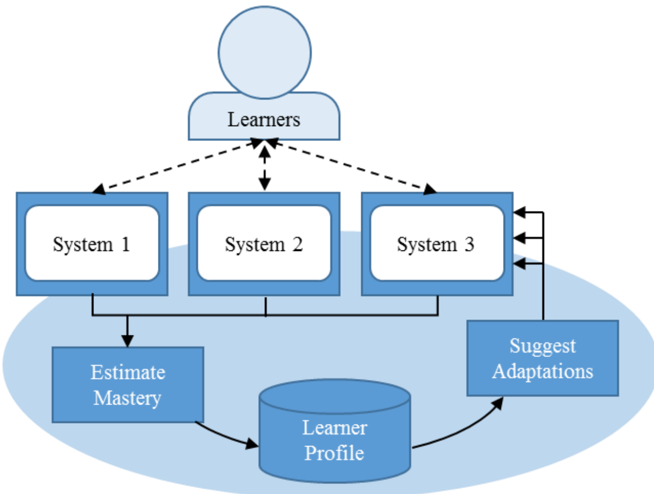

Figure 1: Combining data from multiple learning systems builds a learner profile that may support all systems.

Figure 1 represents a learning solution wherein several learning systems communicate within the ecosystem provided by the TLA. At the top of the figure, the systems that learners interact with directly could be viewed as standalone products. They might be able to work without communicating with other components or participating in the TLA, although their functions that require TLA components would be degraded. But in the depicted ecosystem, supporting back-end components (shaded area) collect data from the learning systems via TLA APIs and process the data to provide meta-adaptation.

The paths passing data, signified by arrows in the figure, represent only one way of connecting TLA-enabled components. The TLA specifications support many more configurations to meet alternative use cases. In this simple case, each learner-facing system publishes information about skill mastery through TLA services. One process queries each service and uses their responses to estimate mastery, updating a learner profile with estimates that combine data from each system. A second process acts on changes in the learner profile to suggest adaptations that each system may act on as it works with the learner. This brief example shows how TLA supports data sharing that can maximize available learner analytics, reduce one-off effort needed to make learning systems work together, and let every learner access just the right learning content in just the right way through meta-adaptation.

Communication

Five kinds of learning topics, or broad data categories, that TLA specifications can help integrate are listed in Table 1. These topics correspond to API specifications that together comprise the TLA. All specifications are open to community discussion or contribution and, at present, all specifications other than xAPI are experimental. Any learning systems that input, process, or generate data within these learning topics has an opportunity to participate in the TLA and receive new data about these topics from other systems within a shared technical environment.

Table 1: Learning Topics Specified in the TLA.

Data Categories	Examples of Data that TLA Helps Share	Integrated at Publication
Learner	Skills Needed, Skills Mastered, Detailed Experience Stream	https://moodle.org
Competency	Prerequisites, Job Skills, Relationships between Skills	http://www.cassproject.org
Activity	Skill Coverage, Content Ratings, Learner Observation Meanings	http://learningregistry.org
Circumstance	Devices Available, Affect, Schedule and Training Duration	https://perlsapp.com
Meta-Adaptation	Recommended Content, Recommended Interaction Patterns	

Table 1 also shows that within the architecture as currently implemented, certain components already exist that provide successful examples of using the TLA to store, process, and communicate learning data. In addition to these, reference implementations of components that comply with the specifications are publicly available as open source, at http://www.adlnet.gov/tla/, so that developers may follow the examples or reuse the code freely.

The TLA offers a relatively low-cost and well-understood approach to software development, using a service-oriented architecture exposed through REST (Representational State Transfer) APIs (Webber, Parastatidis, & Robinson, 2010). This structure was chosen to minimize the technical barrier to entry within the TLA, as new components may be implemented in any programming language, manage their own data as they wish, run independently of any other components, and handle only lightweight communication and management as opposed to more demanding distributed processing protocols. In this way, the TLA aims to increase the capability and reduce the complexity of potential learning solutions. Complexity is reduced because it is not necessary for other component developers or the solution creators to understand components' underlying technologies to make sure they work together. It is only necessary to assemble TLA-compliant components whose functional features meet the specific teaching or training need.

Operating within the REST viewpoint, TLA views individual learning systems or sub-system components as services which can answer particular types of requests. A valuable feature of the service-oriented architecture is that TLA does not specify a single pathway for linking components or learning systems. Instead, each component is available to be queried by each other component. Thus, TLA enables a large number of next-generation learning use cases. For example, TLA can be used to build a system with no adaptation component or with multiple components providing adaptation functionality. TLA defines methods for discovering which components are present and linking to the right ones for desired functionality. When different components exist that implement the same interfaces, discovery capabilities defined in the TLA spec help steer data to the proper storage locations and synthesize queries as needed. As examples, one may wish to duplicate learner test scores to different LMSs, query one or another of the LMSs as the authoritative record for different instructional domains, and average scores across all available records to make a high-level visual display of progress. The service-oriented architecture enables compliant components to carry out each of these tasks.

Within the TLA it does not matter what internal processing, programming languages, operating systems, or computer hardware each component uses. As a result, administrators who are creating a TLA instance can swap components from different developers, upgrade one component without disturbing other components, and run different components of the same type in parallel. TLA administrators keep all data and learning systems under their own control, on their own system, inside their own firewalls. They choose which learning systems participate in handling the data. The goal of the TLA is to encourage participation from all in a common API, rather than to mandate a one-size-fits-all implementation.

Specialized Components

Empirically, many learning systems use different internal representations and logic to reason about *learner mastery* and propose *individualized adaptations*. To reduce the friction introduced by such differences, within our experimental TLA implementation we defined two services specifically to translate or otherwise mediate between different learning systems. These specialized components, the evidence mapper and the adaptation pipeline, aim to reduce integration effort by letting every learning system share data via TLA interfaces without needing to change the internal representation and processing of the learning system. A dynamic approach to mediating between systems is hypothesized to avoid some pitfalls of very large ontologies and manual semantic web approaches that have been attempted in the past. In keeping with the learning ecosystem approach to designing a TLA, the specialized functions are encapsulated within service interfaces that allow any developer to choose different implementations of them or to create a new implementation.

An *evidence mapper* is a service which outputs assertions about an individual that it infers from evidence. Like other services, the TLA interfaces define how the evidence mapper may communicate inferences without proscribing methods or algorithms for making inferences. The reason the evidence mapper plays a vital role is because different learning systems need to contribute to a shared inference about an individual, such as learner mastery, but they may each contribute evidence in a variety of formats.

The TLA design supports both learning systems that publish the detailed semantic meaning of evidentiary xAPI statements, and also those that do not. The TLA does not impose a single ontology or vocabulary for all statements, because any designed vocabulary is unlikely to meet the needs of all learning systems. Semantic markup that explains an xAPI vocabulary is allowed but not required, because it would impose a high barrier on participation in the TLA. Finally, semantic meaning is known to be highly sensitive to context and to change over time, requiring a non-static and intelligent approach to interpreting evidence in context and over long periods.

As a result, a service is needed that can interpret the learner experience into rollup or other high-level form. Typical evidence will be in the form of xAPI experience statements (though any data may become evidence). The assertions of the evidence mapper may be recorded in xAPI and refer back to the supporting statements. Examples of inference might include a mapping from a test score to an estimate of declarative knowledge, or from a specific pattern of actions in a simulation to an inference about habitual behavior or mastery of underlying skills.

Like the evidence mapper, an *adaptation pipeline* is a specialized service which implements interfaces defined in the TLA. The adaptation pipeline provides a location for decisions about adapting the learner experience. Examples of decisions include recommending which learning system best meets an identified need, or providing information that can influence how individual learning systems carry out adaptation internally. Because many learning systems are expected to carry out adaptation internal to their own content and interventions, the design philosophy of the TLA is that adaptations from the pipeline can be usefully viewed as suggestions that add information to each system, rather than as absolute directives. Again, in keeping with the TLA philosophy, there are no requirements on a learning system to carry out externally suggested adaptations.

The adaptation service is a pipeline because the relevant interfaces are designed such that individual components may be assembled in sequence to carry out desired recommendations. This gives users excellent control and ability to override specific functions or reuse highly modular components to build the pipeline that fits a specific use case. For example, adaptation components in a pipeline might each take responsibility for recalling relevant information, aggregating data, identifying currently available devices, generating initial recommendations, filtering and prioritizing recommendations, and down-selecting to find one or a few adaptations that should be presented to a learner. Again, developers can make their specialized processing or "secret sauce" available as a service and carry out any of these steps without needing to share their source code or rewrite it to account for others' internal details.

The implementation status of our TLA demonstration at the time of publication includes working code for service wrappers, reference examples of specialized components, and multiple different learning systems that together use TLA APIs and data models to interoperate. We next describe how the experimental TLA facilitates these systems from different developers working together to produce adaptation across all learning systems, or meta-adaptation.

Implemented Use Case for Meta-adaptation: Adaptive Cyber Training

We describe at a functional level one example of meta-adaptation that can occur in our current reference implementation of components for an adaptive training use case. The example use case has been implemented in the current reference software (Figure 2). As a stepping stone to the meta-adaptation vision, implementing a straightforward example of meta-adaptation helps explore essential aspects of the concept.

The implemented example takes place in a cyber training domain (Nicholson, Massey, O'Grady, & Ortiz, 2016). A trainee, Angela, is studying on her own time to move into a Cyber Security Intelligence Analyst role. The relevant competencies defined by the National Initiative for Cybersecurity Education (NICE; http://csrc.nist.gov/nice/) are stored in an instance of the Competencies and Skills System (CASS), a back-end component which participates in the TLA (http://www.cassproject.org/).

Angela begins an interaction by choosing to read a static document (slide show or textbook chapter) that introduces NICE *knowledge of penetration testing principles, tools, and techniques*. Facts about her reading the document are broadcast via xAPI statements, including the amount of time reading and which pages she lingered over or skipped. An evidence mapper component reads these statements in Angela's record and infers that Angela's knowledge of penetration testing has increased, but also assigns a decreased confidence to its mastery estimate because Angela might not have fully absorbed what she read. As a result, the adaptation pipeline recommends following up with a formative self-assessment on penetration testing.

The recommended self-assessment is presented in a different learning system than the static content, so a user-facing suggestion offers Angela the opportunity to switch systems. The suggestion displays information drawn from the TLA that Angela might want in order to make a decision, such as her estimated time investment and an auto-generated, specific rationale: self-assessment will help find how much she retained from the reading material.

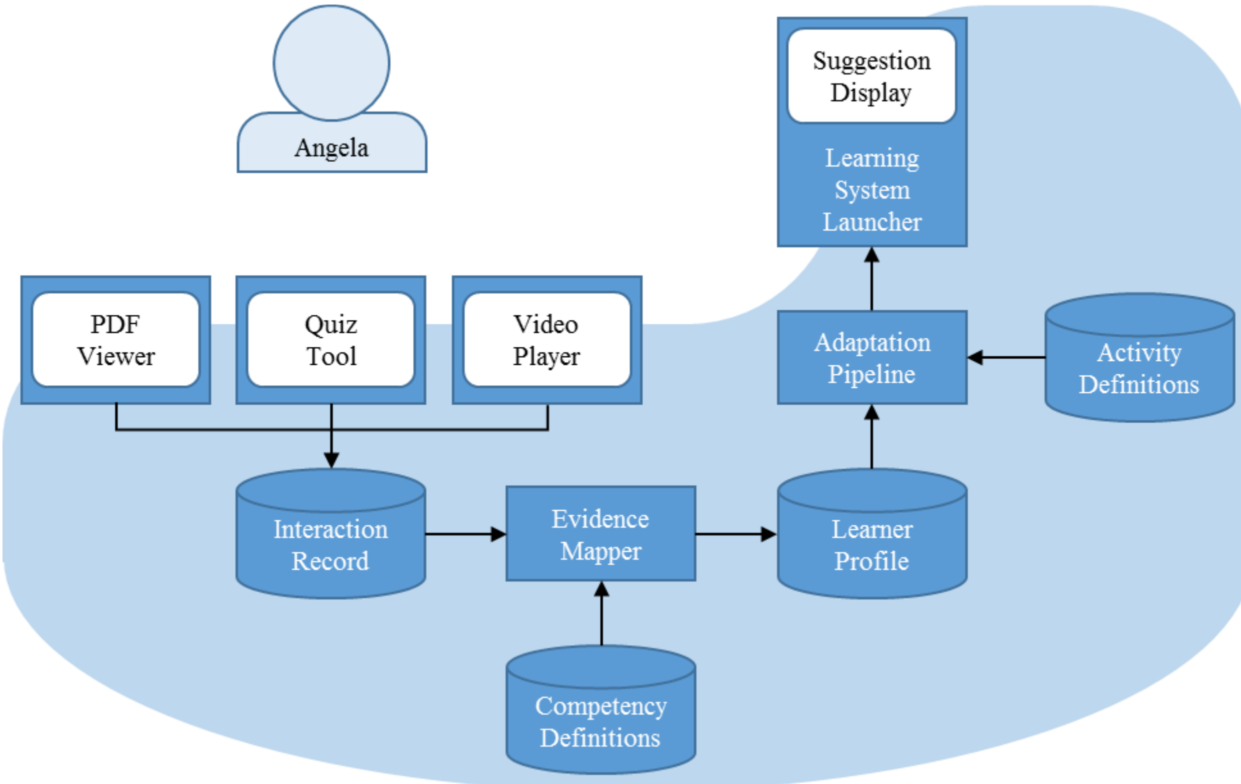

Figure 2: Implemented learning solution that can: capture fine-grained learner analytics from different systems in a common structure, apply the two specialized components Evidence Mapper and Adaptation Pipeline, and carry out meta-adaptation to coordinate the learner's experience across learning systems.

Angela accepts the suggestion and the assessment system is launched with the correct assessment that covers knowledge of penetration testing. As Angela answers the questions, her performance on individual test items indicates that she still needs work on a subset of the topic: recalling penetration capabilities of specific tools. Since Angela has recently read some text-based content on the penetration-testing topic, the adaptation engine next recommends a video relevant to the target material. Angela watches the video and improves her declarative knowledge of the topic, making her ready for effective practice applying her skills.

Detailed documentation and source code implementing the example use case are available in our repository (http://www.adlnet.gov/tla). While the use case presents a simple example, it represents the potential of TLA to advance the state of the art and enable new research and technology development. In the implemented example, the determination that certain learner experiences suggest increased or decreased knowledge about a specific topic is made outside the text viewer, assessment system, and video player, so it does not need to be coded into each. Furthermore, the authors of the text viewer do not need to know in advance what external content such as the video might exist, or need to embed all possible remediation inside their small learning system. Instead, the selection of the remediation content is based on current availability across the TLA ecosystem. In future work, adding to this use case with information about the rich learner experiences in simulators and intelligent tutors may trigger more sophisticated meta-adaptation that learns and plans adaptive responses to specific combinations of learner actions (Folsom-Kovarik et al., 2016).

RESEARCH CHALLENGES

Specific challenges we have encountered in the process of developing our experimental implementation suggest opportunities for immediate future research. First, many components participating in the TLA would benefit from general approaches such as machine learning to automate semantic understanding of the rich data being shared by different learning systems. Second, scalability to incorporate many more users and computer systems could be added via distributed processing approaches. Third, design for privacy and security is needed.

Currently within TLA communication and data specifications, semantic markup on shared data types such as xAPI statements explain the meaning of individual records in a machine-readable manner. This markup is needed because while it may be obvious to a human that a high score is more likely associated with mastery than a low score, a machine such as a TLA component needs to be told explicitly. Furthermore, there are many possible ways heterogeneous systems communicate about related semantic meanings with different assumptions, such as stating a person is a master without defining the comparison population, or broadcasting a score as five (out of five) versus five (out of ten). Finally, nonobvious or context-specific semantic meanings change often and are probably too numerous to enumerate fully, such as whether the fact that a person has searched for a term might indicate an interest in learning about it or merely an assignment to do so, or when a particular action in one scenario indicates mastery if performed with specific timing and not otherwise. As our research explores various limits of static semantic markup (e.g. Yu et al., 2011), we will research how machine learning and dynamic approaches (Folsom-Kovarik et al, 2016) help to understand the semantic meaning of fine-grained and voluminous shared data.

While the current reference implementation of components have not experienced performance problems, we have yet to test their ability to scale under truly massive amounts of data. The scalability of the TLA is not merely a component implementation detail. Instead, it may be that the easily-used REST interfaces the TLA defines are themselves not suited for large and fast-moving data. A research approach to addressing this challenge might be to define parallel interfaces that support greater scalability and fault tolerance through distributed processing. The distributed versions of TLA interfaces should exist in parallel with the existing streamlined versions, in order to reduce barriers to entry for straightforward cases where sophistication is not required.

Finally, since learner data sharing is a goal of the TLA, privacy and cybersecurity should be addressed early. User models and training methods themselves could reveal information that is of operational concern, such as near term activity or the kinds of equipment and procedures used and what personnel are trained to do with them (Raybourn et al., 2015). A properly coordinated attack could access the system to sabotage training or databases in a number of ways to cause harm. For example, removing training material on how to successfully conduct operations would leave the trainees more vulnerable to failure and inefficiency than if they had seen proper material or had proper training. Additionally, the modified system could continue to tell them they are fully trained and ready even if they are not. This loss of system effectiveness may cause a degradation of trust leading to users abandoning the system altogether (Benbasat & Wang, 2005).

Consideration for data privacy and cyber security should not begin at deployment of a large-scale software system. Security and privacy considerations should be interwoven into the software development process from the very beginning. Engaging design assurance experts focused on securing and integrating subsystems into the final system will help address privacy and security concerns (Raybourn et al., 2015). Ongoing dialog allows iterative investigation into the details of particular design decisions and establishment of proper protections. This dialog should carry on from prototype construction all the way to final product to ensure that the final ecosystem delivered is safe, secure, and trustworthy (Raybourn et al., 2015). We are using our current implementation to help define a threat model in order to focus data protection efforts while still allowing valuable data sharing.

Additional Use Cases

In addition to challenges, the present research also suggests a wealth of opportunities to extend into related emerging areas that support learning. While our team initially implemented an adaptive training use case, the TLA is differentiated from a general system for adaptive training because it enables many other learning use cases. Work to select and implement additional specific use cases is predicted to have the effect of making the experimental specifications and architecture in the TLA increasingly robust.

Personnel Management. Instead of adaptive training, improved understanding of individuals in a TLA-enabled ecosystem can be used to identify good candidates for advancement, or connect individuals with recommended fields of work. The same APIs that describe learning experiences can be extended with further development to unify work experience in the same data stores and processing pipelines, thus providing a new and valuable source of data for intelligent personnel decisions.

Usage Monitoring. Fine-grained reporting in a TLA-enabled ecosystem might let institutions recognize which learning content is earning more frequent or more in-depth usage. Instead of simply recording which content is opened, the data defined in the TLA offers information about what individual learners did as they interacted with the content. The data can help decision makers select content to purchase or to emphasize, even without requiring a sophisticated machine learning component in the system. Another outcome of usage monitoring might be to identify effort levels in individual students, helping identify who effectively used the available study tools.

Job Task Support. The TLA could be used to support Just-In-Time (JIT) delivery of help when a person needs support to complete a job task. The delivery device could be an embedded system or a personal mobile device, and the trigger for a tool to offer help might be performance monitoring of job tasks through the same channels TLA uses to monitor instructional performance. The TLA's detailed record specifications would support recommending useful help when a person is confronted at work with an infrequently occurring or otherwise unfamiliar situation.

The Quantified Self. Defense researchers (Blackhurst, Gresham, & Stone, 2012) and even motivated individuals (Swan, 2009) are carefully studying human performance and precursors or mediators that contribute to performance. A TLA-driven approach could help individuals learn about themselves by facilitating the empirical measurement and manipulation of individual experience across many devices and in the context of many different circumstances.

Assistant Integration. At the time of publication, commercial assistants are available in most consumer computer and mobile platforms, including Siri (http://www.apple.com/ios/siri), Cortana (https://www.microsoft.com/en-us/windows/Cortana), and Google Assistant (https://www.google.com/landing/now/). Information passed through the TLA about learners could help tailor each of these assistants to individual needs. For example, when the learner engages in informal learning by searching through a commercial assistant, TLA-enabled components can help identify the appropriate reading level and the background knowledge the learner has. Of course, the TLA ecosystem would also benefit from any information the assistants share about the learner's current context and life experience.

CONCLUSIONS

In a modern learning environment that provides learning across devices and settings, we are researching a general approach to let learning systems communicate and form a coherent experience. Although research and development are ongoing, a learning ecosystem approach is hypothesized to encourage participation from many contributors because it offers all improved learning impact as compared to monolithic operation. The general architecture approach enables valuable meta-adaptation and next-generation learning use cases.

The implemented example of an adaptive training use case described in the present paper will be demonstrated and studied with human participants in 2017. The study will provide an initial characterization of one adaptive training example that can serve as a baseline for future comparisons. Work on the TLA specifications will progress through annual integrations, evaluation studies, and community discussions toward a releasable *version 1* milestone.

The community may engage with the TLA by viewing design documents and source code at the related website, http://www.adlnet.gov/tla/. The research team welcomes contributes and challenges from the community that will help to ensure the TLA supports as many users and learning requirements as possible.

ACKNOWLEDGEMENTS

This material is based upon work supported by the Advanced Distributed Learning (ADL) Initiative under Contract No. W911QY-16-C-0019. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Advanced Distributed Learning (ADL) Initiative.

* Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

REFERENCES

ADL Initiative. (2016a). *Experience API*. Retrieved June 16, 2016, from https://www.adlnet.gov/adl-research/performance-tracking-analysis/experience-api/

ADL Initiative. (2016b). *xAPI-Spec*. Retrieved June 16, 2016, from https://github.com/adlnet/xAPI-Spec

Baepler, P., Walker, J., & Driessen, M. (2014). It's not about seat time: Blending, flipping, and efficiency in active learning classrooms. *Computers & Education, 78*, 227-236.

Baker, R.S., & Inventado, P.S. (2014). Educational data mining and learning analytics *Learning Analytics* (pp. 61-75): Springer.

Baker, R.S.J.d., Rodrigo, M.M.T., & Xolocotzin, U.E. (2007). *The Dynamics of Affective Transitions in Simulation Problem-Solving Environments*. Paper presented at the 2nd International Conference on Affective Computing and Intelligent Interaction, Lisbon, Portugal.

Benbasat, I., & Wang, W. (2005). Trust in and adoption of online recommendation agents. *Journal of the association for information systems, 6*(3), 4.

Blackhurst, J.L., Gresham, J.J., & Stone, M.O. (2012). The quantified warrior. *Armed Forces Journal*.

Coppus, S.F., Emparanza, J.I., Hadley, J., Kulier, R., Weinbrenner, S., Arvanitis, T.N., et al. (2007). A clinically integrated curriculum in evidence-based medicine for just-in-time learning through on-the-job training: the EU-EBM project. *BMC medical education, 7*(1), 1.

Corbett, A.T., & Anderson, J.R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction, 4*, 253-278.

Dodds, P., & Fletcher, J. (2004). *Opportunities for new "smart" learning environments enabled by next generation Web capabilities* (IDA-D-2952). Alexandria, VA: Institute for Defense Analyses.

DODI 1322.26. (2006). *Development, Management, and Delivery of Distributed Learning*. Office of the Under Secretary of Defense for Personnel and Readiness.

Folsom-Kovarik, J.T., Jones, R.M., & Schmorrow, D. (2016). *Semantic and Episodic Learning to Integrate Diverse Opportunities for Life-Long Learning*. Paper presented at the MODSIM World Conference, Virginia Beach, VA.

Greenert, J.W. (2011). *Navy, 2025: Forward Warfighters*. Washington, D.C.

Long, R., Hruska, M., Medford, A., Murphy, J., Newton, C., Kilcullen, T., & Harvey, R.L. (2015). Adapting Gunnery Training Using the Experience API. In *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC). Orlando, FL*.

McLoughlin, C., & Lee, M.J. (2010). Personalised and self regulated learning in the Web 2.0 era: International exemplars of innovative pedagogy using social software. *Australasian Journal of Educational Technology, 26*(1), 28-43.

Neller, R.B. (2016). *FRAGO 01/2016: ADVANCE TO CONTACT*. Washington, D.C.

Nicholson, D., Massey, L., O'Grady, R., & Ortiz, E. (2016). *Tailored Cybersecurity Training in LVC Environments*. Paper presented at the MODSIM World Conference, Virginia Beach, VA.

Nye, B.D. (2016). ITS, The End of the World as We Know It: Transitioning AIED into a Service-Oriented Ecosystem. *International Journal of Artificial Intelligence in Education, 26*(2), 756-770.

Poltrack, J. (2014). *Moving from SCORM to Experience API (xAPI)*. Paper presented at the Interservice / Industry Training, Simulation and Education Conference (I/ITSEC), Orlando, FL.

Raybourn, E.M. (2007). Applying simulation experience design methods to creating serious game-based adaptive training systems. *Interacting with computers, 19*(2), 206-214.

Raybourn, E.M. (2013). *Designing Compelling Transmedia Learning for Military Training & Education* (Tutorial). Orlando, FL: Interservice / Industry Training, Simulation and Education Conference (I/ITSEC).

Raybourn, E.M. (2014). A new paradigm for serious games: Transmedia learning for more effective training and education. *Journal of Computational Science, 5*(3), 471-481.

Raybourn, E.M., Fabian, N., Davis, W., Parks, R.C., McClain, J., Trumbo, D., et al. (2015). *Data Privacy and Security Considerations for Personal Assistants for Learning (PAL).* IUI Companion '15: Proceedings of the 20th International Conference on Intelligent User Interfaces Companion, 69-72: ACM.

Regan, D., Raybourn, E.M., & Durlach, P.J. (2013). Learner Modeling Considerations for a Personalized Assistant for Learning (PAL). In R. A. Sottilare, A. Graesser, X. Hu & H. Holden (Eds.), *Design Recommendations for Intelligent Tutoring Systems: Learner Modeling* (Vol. 1, pp. 217): U.S. Army Research Laboratory.

Richardson, J.M. (2016). A Design for Maintaining Maritime Superiority. Washington, D.C.

Shute, V., & Towle, B. (2003). Adaptive e-learning. *Educational Psychologist, 38*(2), 105-114.

Siegfried, R., & van den Berg, T. W. (2015). *M&S as a Service: Paradigm for Future Simulation Environments.* Paper presented at the Interservice / Industry Training, Simulation and Education Conference (I/ITSEC), Orlando, FL.

Swan, M. (2009). Emerging patient-driven health care models: an examination of health social networks, consumer personalized medicine and quantified self-tracking. *International journal of environmental research and public health, 6*(2), 492-525.

U.S. Department of the Air Force. (2015). *Human Capital Annex to the USAF Strategic Master Plan*. Washington, DC.

U.S. Department of the Army. (2011). TRADOC Pamphlet 525-8-2: The United States Army Learning Concept for 2015. Fort Monroe, VA.

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education, 16*(3), 227-265.

Webber, J., Parastatidis, S., & Robinson, I. (2010). *REST in practice: Hypermedia and systems architecture*: O'Reilly Media, Inc.

Yu, H.Q., Dietze, S., Li, N., Pedrinaci, C., Taibi, D., Dovrolls, N., Stefanut, T., Kaldoudi, E., and Domingue, J. (2011). *A linked data-driven & service-oriented architecture for sharing educational resources.* Paper presented at the 1st International Workshop on eLearning Approaches for Linked Data Age (Linked Learning 2011), 8th Extended Semantic Web Conference (ESWC2011), Heraklion, Greece.