More Than the Sum of Their Parts:
Case Study and General Approach for Integrating Learning Applications

Michael Freed	J.T. Folsom-Kovarik	Sae Schatz
SRI International	Soar Technology Inc.	ADL Initiative
Menlo Park, CA	Orlando, FL	Alexandria, VA
freed@ai.sri.com	jeremiah@soartech.com	sae.schatz@adlnet.gov

ABSTRACT

Learning system interoperability standards let applications share or connect infrastructure, reducing development costs, administrative burden, and user friction. We propose a more comprehensive form of interoperability that supports shared user data, user interfaces, and session management services. This will let applications leverage differences in how and when they are used to achieve better learning outcomes than is possible with only native capabilities. To demonstrate this concept, we integrated two complementary learning applications. PERLS is a Personal Assistant for Learning phone app that supports adult self-regulated learning. It recommends learning activities across a range of applications and devices, and provides a coherent user experience for learners as they progress through parallel trajectories varying in topic, timespan, intensity, and (in)formality. PALMs is a web service that uses flashcard-like interactions to train visual knowledge. It accelerates learning by adaptively optimizing the content sequence based on a model of learning rate and retention.

The applications were integrated to let PERLS recommend and launch PALMs content, then seamlessly hand off interaction control so that users are unaware of switching applications. PALMs was made able to schedule user interactions in future PERLS sessions, making it possible to space practice on a longer timescale to increase retention of PALMs content. Without PALMs, PERLS has no principled way to accelerate visual knowledge learning. Without PERLS, PALMs has no way to extend its retention enhancing model to long timescales. Integration required substantial time and cooperation to define objectives, co-design APIs, and synchronize engineering across organizations. The Total Learning Architecture (TLA), currently in development, will reduce integration requirements with the goal of making cross-application synergy more common. In this paper, we describe our technical approach to combining PERLS and PALMS, then describe how TLA, treating this effort as a use case, could reduce barriers for comparable future integration efforts.

ABOUT THE AUTHORS

Michael Freed, Ph.D. is a Program Director with SRI International in the Artificial Intelligence Center. His research focuses on augmenting human capabilities using intelligent personal assistants and other human centered computing technologies.

J.T. Folsom-Kovarik, Ph.D. 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.

Sae Schatz, Ph.D., currently serves as the Director of the Advanced Distributed Learning (ADL) Initiative, a research and development program under the U.S. Office of the Assistant Secretary of Defense (Readiness).

More Than the Sum of Their Parts:
Case Study and General Approach for Integrating Learning Applications

Michael Freed	J.T. Folsom-Kovarik	Sae Schatz
SRI International	Soar Technology Inc.	ADL Initiative
Menlo Park, CA	Orlando, FL	Alexandria, VA
freed@ai.sri.com	jeremiah@soartech.com	sae.schatz@adlnet.gov

INTEGRATING COMPLEMENTARY LEARNING APPLICATIONS

Rapid innovation and increasing demand for technology-enhanced learning continues to produce many new products and approaches. This represents an opportunity to improve learning outcomes by using the best of an increasingly rich assortment of approaches. It also presents a challenge for training managers, instruction designers, teachers, and self-directed learners who need to select learning resources from many and diverse options. For example, people routinely learn job-related knowledge from a self-assembled mixture of resources when no course or formal instruction is available. This informal learning requires identifying resources, obtaining access, and coping with gaps, redundancies, implicit prerequisites, and other difficulties that challenge the learner's determination and meta-cognitive faculties. In this light, it is reasonable to expect self-learners and other decision-makers to limit the range of options they consider, even when better ones may be available.

Technology offers two potential solutions. The most common is to provide a platform, such as a Learning Management System (LMS) that lets developers extend built-in functionality by creating and integrating plug-ins (Graf and List, 2005; Al-Ajlan and Zedan, 2008). An advantage of the platform-based approach is shared infrastructure such as common data models for users and content, shared data management and analytics services, and centralized administrative functions. This lets developers focus on implementing the distinctive aspects of the new learning activity rather than creating a full standalone application. The disadvantage of this approach is that the design and functionality of the platform inevitably favors some learning technology concepts, but makes it difficult or impossible to incorporate others.

A second approach is to define extensible interoperability standards that let several applications operate independently and share as little or as much as desired (Kelley et al., 2007). Compared to the platform approach, this looser coupling offers greater flexibility but less cost savings from shared infrastructure. For example, one application might be designed for occasional, high value training requiring extensive computational resources and strong security guarantees. A second might be designed for informal daily use on a low-powered mobile consumer device. A single platform amenable to one of these use cases would likely create implementation barriers for the other use case. But loosely coupled integration would let specialized front-end and back-end components implement each use case and share data as appropriate. Loose coupling is especially suitable for legacy applications that cannot easily be re-engineered to meet platform requirements and for innovative technologies that do not conform with preconceptions designed into a platform's core services and data models.

So far, the LMS single platform approach has proven the far more popular of the two approaches. However, changes in the costs and benefits of creating standalone, loosely integrated applications promise to shift the balance. On the cost side, more capable cloud computing services make standalone applications easier and cheaper to build, while better tools and practices for micro-service architectures reduce integration costs (Namiot and Sneps-Sneppe, 2014); at the same time, increasing device diversity and use-pattern variability makes it more challenging to build one-size-fits-all platforms. Benefits depend on the range of use cases supported by the interoperability model. However, existing standards generally focus on use cases related to the beginning and end of sessions – e.g. support for single-sign-on to streamline session start, and reporting outcome data at the end.

We propose a more expansive concept supporting data and control sharing within and across sessions. Using a semantic interoperability approach (Oberst 2003), applications that acquire general information about user needs, preferences, competencies, and interaction patterns can provide that information to other applications. This makes it advantageous to combine applications that vary in how, when, and where they interact with a user. Similarly, allowing an application to schedule interactions in or supply content to the others effectively expands the available range of learning strategies each can employ beyond what would be possible with only native capabilities and usage patterns. With each able to take advantage of session contexts available to the others, applications become more than the sum of their parts.

To demonstrate this concept, we integrated two complementary learning applications. PERLS is a Personal Assistant for Learning (PAL) phone app that supports adult self-regulated learning (Freed et al., 2014). It recommends learning activities across a range of applications and devices, and creates a coherent experience for users as they progress through parallel learning trajectories that vary in topic, timespan, intensity, and degree of formal structure. It is intended to be used for a few minutes a day over months or years. PALMs is a web service that uses flashcard-like interactions to train visual knowledge. It accelerates learning by adaptively optimizing content sequence based on a model of retention and learning rate. It is designed for occasional use lasting several hours to meet specific learning goals. The applications were integrated to let PERLS recommend and launch PALMs content, then seamlessly hand off interaction control. PALMs was made able to effectively push content into PERLS and conditionally schedule future review sessions. This made it possible to space practice on a much longer timescale with the aim of greatly increasing retention of PALMs content. Without PALMs, PERLS has no principled way to accelerate visual knowledge learning. Without PERLS, PALMs has no way to extend its retention enhancing model to long timescales.

Integration required substantial time and cooperation to define objectives, co-design APIs, and synchronize engineering across organizations. The Total Learning Architecture (TLA), currently in development, promises to reduce the burden of future integration efforts, with the goal of making cross-application synergy both more common and more powerful (Regan, Raybourn, & Durlach, 2013). In this paper, we describe our technical approach to combining PERLS and PALMS, then describe how TLA, treating this effort as a use case, could reduce barriers for comparable future integration efforts.

A PERSONAL ASSISTANT FOR SELF-REGULATED LEARNING

PERLS (Pervasive Learning System), developed under ADL's Personal Assistant for Learning program, is a mobile app supporting adult self-regulated learning. Self-regulated learning (SRL) complements formal course-based instruction, emphasizing "autonomy and control by the individual who monitors, directs, and regulates actions toward goals of information acquisition, expanding expertise, and self-improvement" (Paris and Paris 2001). For adults, self-learning is the predominant form of learning, with average time spent ranging from 200–650 hours per year (Livingstone 1999), accounting for at least 70% of total learning effort (Kim et al., 2004).

Although self-learning can incorporate e-courses and other elements of formal instruction, it falls mainly to the learner to recognize learning needs, set goals, acquire resources, monitor progress, detect and remedy problems, and otherwise structure their own learning experience. Each such challenge represents an opportunity for technology to enhance learning, and collectively, a strong reason to integrate diverse technologies addressing different facets of the overall problem.
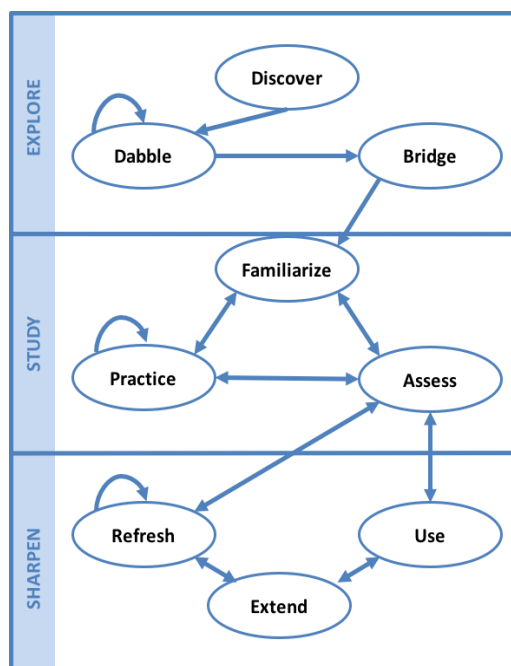
A Model of Self-Regulated Learning

PERLS incorporates a model of the learning process (Figure 1) involving three main phases – Explore, Study, and Sharpen – each with a set of key activities or "sub-phases." The Explore phase starts with Discovery, where the learner becomes aware of a topic and why it might be important to learn more. Learners who are naturally curious, perceptive, and social will tend to become aware of important topics in a timely way. Others may struggle, and benefit from technology that supports Discovery. Dabbling is undemanding interaction with topic materials, consistent with a low level of commitment to long-term learning. Content is typically brief and easy to absorb, providing the learner with an opportunity to assess or nurture motivation, gain confidence, and become oriented to the basic terms and ideas in the topic domain. Bridging is preparing for intensive, high commitment learning, by, e.g., defining goals, setting time and effort expectations, and identifying learning resources.

Learners may progress past Exploration and transition to the Study Phase, indicating a commitment to gain a competency, complete a course, or achieve some other goal. Study activities include Familiarization to obtain a foundation of knowledge, Practice to develop skills, and Assessment to establish a starting point and measure goal progress.

Figure 1. SRL Phase and Activity Model

Formal instruction is a special case of Study since learners might achieve their goals by formal or informal methods. After achievement, learners transition to Sharpen Phase sustainment activities including Refreshing for retention, Extension to stay current and build on prior learning, and Using what they have learned to enhance fluency. In each phase and activity, some learners are largely self-sufficient while others can benefit from technological support.

How PERLS Supports Self-Learning

Self-learning differs from formal instruction in how and when people engage with learning content. For example, pre-study exploration and post-study sustainment lengthen the timeline of learning compared to a self-contained course. Effective support technology must be engaging and habit-forming so that self-learners use it regularly during learning trajectories that can last months or years. And because self-learning takes place in the context of the learner's daily activities, support technology will be used more reliably if users can take advantage of unplanned time slots, whenever and wherever these occur. Thus, it should be easy to use, quick to start, and provide "microcontent" (Hug et, al. 2006) suitable for brief time slots.
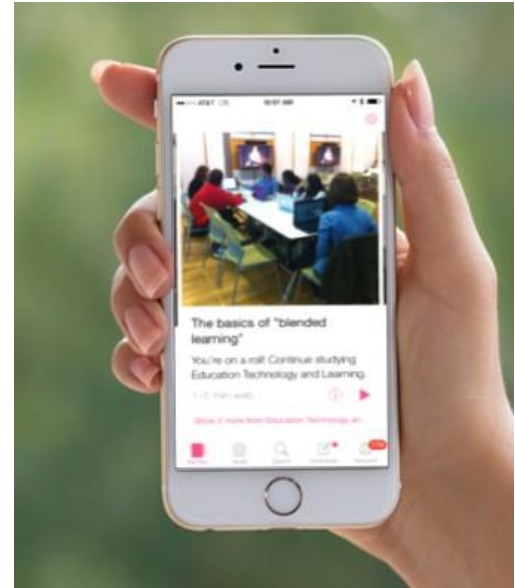
Figure 2. PERLS uses a card-based UI

Modern mobile apps use a range of user experience design approaches to meet these criteria. PERLS adopts a card-based approach (Figure 2), where each card displays a content or action recommendation, and a "swipe" gesture is used to advance to the next recommendation item in a sequence. Some recommendations cover topics that are unfamiliar to the learner, but potentially relevant. Others cover topics of recent exploratory interest, ongoing study, or past study and ongoing interest. In other words, recommendations span SRL phases as well as topics. The recommendation engine ranks each topic for degree of interest based on user behavior data, user goals, interests of similar users, and organizational requirements. Content on high interest topics are then evaluated for fitness to the learner's current SRL phase for that topic. For example, if the learner is determined to be Dabbling on a topic, then brief, entertaining, low difficulty content such as news articles will be rated as high fitness. Lectures and complex skill practice tasks will be rated as low fitness, even if overall topic interest is high. Cards are sequenced so that content on high interest topics with high fitness are prioritized and presented early.

In line with Knowles's (1984) first principle of adult learning – "There is a need to explain the specific reasons things are being taught" – PERLS generates a human-readable explanation for each recommendation based on a taxonomy of motivation types. For example, one recommendation might advance a declared mastery goal, while another might be explained as a trending interest among peers, an organizational learning requirement, or as highly rated by people who shares the user's interest on a given topic. Explanations are shown on the card as text "sell points" to help persuade the learner to accept the recommendation and to enhance their awareness of learning needs.

PERLS recommends different kinds of content to address different self-learning needs. For example, short duration documents, videos, and other standard media are well-suited to Dabbling and other low intensity learning phases. Specialized content types such as event notifications, "quiz cards" (single multiple choice questions), and "action cards" (suggesting strategies such as setting an explicit learning goal) are used for phase-specific needs not easily addressed with standard formats. PERLS also recommends content residing on other applications. This includes e-courses that can only be accessed using a different application or are best experienced on a larger screen. It also includes specialized applications providing distinctive learning experiences and advantages.

USE CASE: INTEGRATING PERLS AND PALMS

For users, the benefit of using a variety of learning technologies must be balanced against the cost of having to become aware of technology options, acquire access, learn functionality, identify useful content, and make good choices about when and how to switch between them. In practice, self-learners need to limit the number of applications they use to some manageable variety. To investigate how technology might be used to maximize the advantages of combining applications while eliminating or minimizing costs, we integrated PERLS with several applications and developed solutions to user experience issues that arose. One such effort focused on PALMs (Kellman and Massey, 2013), a web application that uses flashcard-like
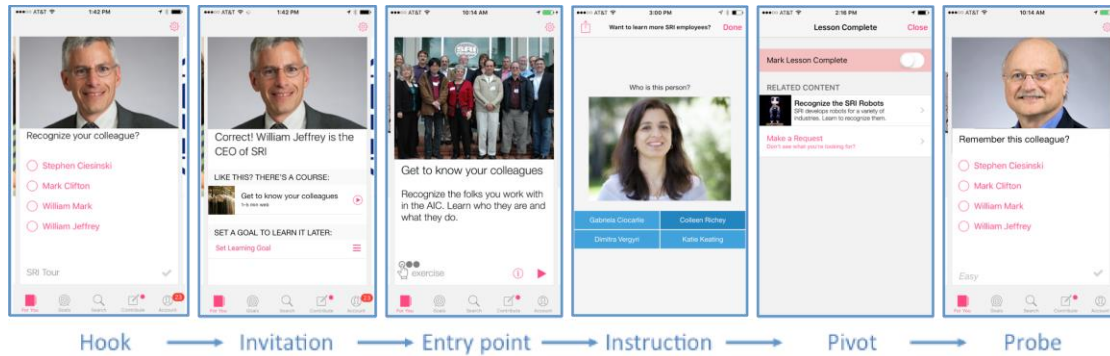
Figure 3. PERLS supports use of PALMs for Explore, Study, and Sharpen phases

interaction design to train visual knowledge. It accelerates learning by adaptively optimizing the content sequence based on a model of learning rate and how spaced practice affects retention (Dempster, 1988; Greene, 1989).

Both PERLS and PALMs benefit from integration. For PERLS, PALMs offers a distinctive capability to rapidly train visual knowledge. It also represents an important integration use case: a web service that can be rendered effectively on a small, phone-sized screen. For PALMs, PERLS offers a way to make more people aware of the system and of specific, personally relevant content. It also provides a way to greatly increase retention of PALMs content. This last advantage relies on leveraging differences in how users are meant to interact with the two systems. PERLS is designed for brief but frequent interaction over months or years, as users check out what PERLS has to offer, possibly select one or a few, and then proceed with their day. PALMs is designed for occasional use, with sessions expected to last several hours while the learner progresses to mastery with a single Perceptual Learning Module (PLM). On its own, PALMs counts on users to realize that they will need to repeat training at spaced intervals to retain knowledge and maintain mastery (Greene, 1989). Integration allows PALMs to schedule periodic brief assessments that take in PERLS sessions, and to have the PERLS recommender suggest retraining when needed.

Figure 3 shows an example of the integrated workflow in a PERLS instance used to help onboard new employees. The PERLS user first encounters PALMs content in the form of a *hook* card, a single quiz question extracted from a PALMs PLM. In this case, the topic is Fellow Employees, with PALMs helping the user learn faces and names. The quiz card supports Discovery of the topic. If they choose to answer the question, they are shown the result and *invited* to begin Studying the Fellow Employees PLM. If they refuse the immediate opportunity, which is likely given the several hours expected completion time, the PERLS recommendation engine will recommend that content again in the future.

Upon selecting the content, control of the interaction is transferred to the PALMs server, which provides *instruction* until mastery is achieved (by an internal PALMs metric) or the user chooses to pause the session and return to PERLS. Taking advantage of PALMs being a mobile-friendly web service that allow PERLS to deep-link content, it was possible to design a seamless workflow, as if PALMs was a built-in PERLS function rather than a separate system. After completing the PLM, the user is offered additional recommendations (*pivots*) on related topics. Days or weeks later, PALMs provides PERLS with information to *probe* retention of mastered content. PERLS presents the probe as a quiz card and, if warranted by the results, begins recommending that the learner repeat the training. Supporting this integration design required developing or enhancing three PERLS APIs:

Catalog API. The catalog allows a curator using the PERLS Content Management UI to request a list of all PERLS-compatible content from the target application. To be compatible, a learning object must include metadata that PERLS needs in order to recommend the content at the right time (e.g. expected duration), launch it (e.g. URL for web services), synthesize hook content for it (e.g. a question, answer, and distractors), and render it as a card (e.g. a picture). Curators can review the list from within the CMUI and select items to ingest into a PERLS corpus.

Launch API. The launch API is used to initiate a session for a specified user and content object residing within another application. For PALMs, this includes specification of user credentials and the target PLM. The PALMs implementation of the Launch API included support for deep linking of content and single sign on, allowing a seamless transition from PERLS aimed at maximizing user experience quality.

Probe API. Probes are a short series of quiz cards used to assess retention of previously learned content. By default, the Probe UI is called one week after completing Study Phase content object such as a PALMs PLM. A call using the probe API required the application (PALMs in this case) to return three questions. PERLS presents these questions and triggers a re-training recommendation when two or more are answered incorrectly. Because forgetting rate varies considerably across individuals, content, and situations (Loftus, 1985), the probe API is called periodically to help determine when enough forgetting has transpired that retraining will be valuable.

INTEGRATION VIA TLA

The Total Learning Architecture (TLA) is an ADL Initiative concept for lifelong, ubiquitous, tailored learning through interoperation of learning systems (such as PERLS and PALMS) in a standardized architecture. Key to the TLA vision are its applicability to multiple learning use cases, ability for individual learning system providers to support part of the TLA without committing to all of it, and low technical barriers or effort required to begin using TLA. After research, development, and community discussion, the TLA may evolve into a standard which can make learning system integrations more *efficient* in cost to develop and set up learning, as well as more *effective* in terms of impact on learning.

TLA Efficiency

The main sources of cost savings envisioned include costs to develop a learning system and costs for purchasing or otherwise acquiring a learning solution that has more than one learning system.

We have described some of the design and engineering effort required to integrate PERLS and PALMS. Since that time, TLA has begun defining APIs and data models that support these types of learning system interoperation. Several are similar to the APIs developed for PERLS and PALMS, because TLA is being designed to include adaptive learning and lifelong learning use cases. TLA APIs and data models describe how one learning system can record learner needs and experiences, find content and launch another system, or request adaptive content. For future integration tasks, TLA will reduce development costs by contributing *unity* of design to these points of contact between systems. TLA-enabled systems will *agree on data models* that PERLS and PALMS had to negotiate in a one-off manner, such as defining the possible topics to be taught in the catalog or the available question types for a probe. TLA-enabled systems will also have access to built-in transport mechanisms and data transfer metaphors that structure communication.

The cost savings that arise from unifying interoperation via the TLA can be emphasized in cases where an API does not already exist. For example, PERLS might want to start transmitting learning context, such as demographics or current location, to PALMS. A TLA API already supports this new data sharing, so it reduces the corresponding design, implementation, and technical communication tasks (and their associated risks). So, unified design is one way the TLA data models and APIs provide a cost-saving head start on integration and interoperation between learning systems, but at the same time the TLA does not limit how each learning system represents or uses the same data *internally*. If one system needs to hold certain kinds of data that the other system does not, then the TLA does not prevent it. This represents an important difference from a single platform approach, where unity of design leads to many design commitments which cannot satisfy all needs at once.

TLA will also help to reduce the cost of acquiring a learning solution. When different learning systems are available that respond to a particular subset of the APIs, those learning systems may be interchanged. The purchaser may pick and choose one learning system or another. There is no increase in cost if two systems from different providers operate together with each other, or if a system from one provider takes the place of the corresponding system built by a different provider. As a result, the ADL Initiative vision for the TLA is to create a *marketplace of ideas* that serves to reduce costs through healthy competition. Purchasers in this notional marketplace will be able to identify which parts of the API are worthwhile for their specific use cases. In our example where one use case with strong security needs contrasted against another use case for low-stakes mobile learning, purchasers would be able to select appropriate learning systems for their requirements based on which parts of the TLA APIs each learning system implements. Although they do not exist yet, future APIs might even define (optional as always) different security models for data communication to meet each of these needs.

Total cost of ownership includes not only initial purchase but also cost to maintain a learning solution. In a solution with many learning systems interoperating, the maintenance costs can increase. TLA is designed to mitigate this risk by enabling separate maintenance of each learning system. TLA-enabled learning systems will be effectively hot swappable, enabling cloud deployment and approaches in the future such as horizontal scaling whenever many learners need to use the system at

once. Deploying or changing one system will minimally impact, or be impacted by, the other systems present. Furthermore, each purchaser may make individual decisions about where data is stored, what data may be shared, and with what requestors. Data security and privacy controls are an ongoing focus of research in the context of TLA integration (Raybourn et al., 2015).

TLA Efficacy

Learning systems that communicate via the TLA will be more effective because of that communication, and also because of increased learning science support that will benefit across the TLA.

PERLS and PALMS demonstrate an example of benefits realized by joining two learning systems. When it becomes less costly and easier for learning systems to communicate, the systems that take advantage of communication will be able to both share data and call on other systems for functionality they do not contain. Sharing data will provide increased knowledge about individual learners that let each system better understand knowledge, skills, and abilities as well as cognitive states and traits, needs, and context. As a result of sharing these types of data, learning systems will no longer need to start from zero when encountering a new learner. Calling on other systems means that it is no longer necessary for each system to implement all functionality that is deemed helpful for learning. Just as PERLS does not need to contain all the learning content in PALMS, and PALMS does not need to understand how location and schedule affect the type of content PERLS recommends, learning system interoperation in general enables a single experience from the learner's point of view to draw on the best capabilities of several systems behind the scenes.

The science of learning has much to offer applied training when systems work together in the TLA. Evidence-based instructional design can help determine exactly what changes are necessary in how content is portrayed so that it helps individual learners. The TLA makes this kind of evidence technically easier to gather. It is possible to experiment with different components, quickly swapping out or reconfiguring one while holding all others constant to create comparisons and even randomized controlled trials. In addition, the increased amount of data being shared provides opportunities for longitudinal training transfer or decay studies and valuable grist for now common approaches such as data mining or data-hungry deep learning. Finally, making a learning system compatible with the relevant parts of the TLA is a good way to increase the reach of research it encapsulates and makes the research utility more readily available for the rest of the applied world to adopt.

In summary, integration via the TLA has potential to decrease costs and increase the impact of learning systems such as PERLS and PALMS. Engineering work for the integration can happen one time instead of many, and once complete the integration gives access to compatible learning systems. The reduced cost of integration and maintenance can yield increased focus on engineering that enhances learning goals instead of technical constraints.

NEXT STEPS

We are extending PERLS to use TLA data models and APIs, allowing it to recommend and launch content on TLA-integrated learning systems as it currently does for PALMS. The first applied test of the TLA concept will be in an adaptive cybersecurity training setting with soldiers at Ft. Bragg. In that study, PERLS will form the starting point and hub of the learner experience. Individual learners will check in to PERLS and select from recommended content, including content provided by 3rd party learning systems on various devices such as a laptop or a tablet. In turn, each of the other learning systems communicates back to PERLS (and each other) information about the learner experience within the system. As a result, PERLS will adapt its recommendations, offering content that matches a learner's stated interest or skipping over content that learners have already mastered. We anticipate that in this study, the TLA as a whole will enable system interoperation and collect detailed, synthesized data that would not be available without it.

The TLA is an evolving set of interfaces and models that we hope will continue growing into a future specification and standard for public use. Starting with successful integrations such as PERLS and PALMS as an example, we hope to introduce a powerful tool for replacing one-off integrations between pairs of learning systems. The research and development we are carrying out now is iteratively gathering requirements and refining designs in order to ensure TLA will fully support adaptive training and many other needs that exist in the learning community today.

ACKNOWLEDGEMENTS

REFERENCES

Al-Ajlan, A. and Zedan, H. (2008) Why Moodle? In Proceedings *Twelfth IEEE International Workshop on Future Trends in Distributed Computing Systems*.

Dempster, F. (1988) The spacing effect: A case study in the failure to apply the results of psychological research. American Psychologist, Vol 43(8), Aug 1988, 627-634.

Freed, M., Yarnall, L., Dinger, J., Gervasio, M., Overholtzer, A., Pérez-Sanagustin, M**.**, Rochelle, J., Spaulding, A. (2014) PERLS: An Approach to Pervasive Personal Assistance in Adult Learning. Proceeding of the 2014 Interservice/Industry Training, Simulation, and Education Conference.

Graf, S. and List, B. (2005) An evaluation of open-source e-learning platforms stressing adaptation issues. Proceedings of Advanced Learning Technologies, Fifth IEEE International Conference, ICALT, pp. 163-165

Hug, Theo; Lindner, Martin; Bruck, Peter A. (eds.) (2006): Microlearning: Emerging Concepts, Practices and Technologies after e-Learning. In *Proceedings of Microlearning 2005*. Innsbruck: Innsbruck University Press, 2006.

Kellman, P.J. and Massey, C.M. (2013) Perceptual learning, cognition, and expertise. In B.H. Ross (Ed.) The psychology of learning and motivation (Vol. 58, pp 117-166). Amsterdam: Elsevier Inc.

Kelly, B,. Wilson, S, and Metcalfe, R. (2007) Openness in Higher Education: Open Source, Open Standards, Open Access. In *Proceedings of the Conference on Electronic Publishing*, Vienna, Austria.

Kim, K., Collins Hagedorn, M., Williamson, J., and Chapman, C. (2004). Participation in Adult Education and Lifelong Learning: 2000–01 (NCES 2004-050). U.S. Department of Education, National Center for Education Statistics. Washington, DC: U.S. Government Printing Office.

Knowles, M. (1984). The Adult Learner: A Neglected Species (3rd Ed.). Houston, TX: Gulf Publishing.

Knowles M. S., Holton E. F., Swanson R. A. (2011). The adult learner (7th ed.). Burlington, MA: Butterworth-Heinemann/Elsevier.

Livingstone, D.W. (1999). Exploring the Icebergs of Adult Learning: Findings of the First Canadian Survey of Informal Learning Practices. Canadian Journal for the Study of Adult Education. 13(2): 49-72.

Loftus, Geoffrey R. (1985). "Evaluating forgetting curves". *Journal of Experimental Psychology: Learning, Memory, and Cognition*. 11 (2): 397–406.

Marsick V. J., Watkins K. E. (1990). Informal and incidental learning in the workplace. London, England: Routledge.

Namiot, D. and Sneps-Sneppe, D. (2014) On micro-services architecture. International Journal of Open Interfaces Technology. 9 (2), 24-27.

Oberst, L. (2003) Ontologies for semantically interoperable systems. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*. Pages 366-369.

Paris, S., Paris, A. (2001). Classroom Applications of Research on Self-Regulated Learning. Educational Psychologist. 36 (2), 89-101.

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).* Paper presented at the 20th International Conference on Intelligent User Interfaces.

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.