Artificial Intelligence in Education 2018

London, United Kingdom

Workshop Proceedings:

Exploring Opportunities to Standardize Adaptive Instructional Systems (AISs)

Workshop Chair:

Robert A. Sottilare, Ph.D.
US Army Research Laboratory

Workshop Committee:

Avron Barr
IEEE Learning Technologies Standards Committee

Keith Brawner, Ph.D.
US Army Research Laboratory

Arthur Graesser, Ph.D.
University of Memphis

Xiangen Hu, Ph.D.
University of Memphis

Robby Robson, Ph.D.
Eduworks, Inc.

Preface

This workshop was conducted under the auspices of the AI in Education Conference as part of the Festival of Learning in London, June 2018. This workshop is focused on exploring opportunities for standards for a class of technologies known as Adaptive Instructional Systems (AISs). Adaptive instruction uses computers and AI to tailor training and educational experiences based on the goals, learning needs, and preferences of each individual learner and team of learners. In December 2017, the IEEE Learning Technologies Standards Committee (LTSC) approved the formation of a study group to examine the feasibility and efficacy of standards for AISs. This workshop and its associated papers are intended to expose the broader ITS community to recent activities and plans, and solicit input on low hanging fruit (near-term opportunities) to develop AIS standards.

This month, the IEEE Standards Association approved the formation of a new working group to advance the discussion and development of AIS standards that was begun by the AIS study group. The AIS standards working group was formed under IEEE Project 2247. Additional information about this working group and its activities can be found by signing up to participate as a working group member at: instructionalsciences.org.

You don’t have to be an IEEE member to a member of the working group. Please sign up today!

June 2018
Robert Sottilare

Table of Contents

Overcoming Barriers to the Adoption of IEEE Standards for Adaptive Instructional Systems (AISs)

Robby Robson[1], Robert Sottilare[2], and Avron Barr[3]

[1]Eduworks Corporation, [2]Army Research Laboratory, [3]IEEE Learning Technologies Standards Committee

Abstract. This paper reviews the case to be made for IEEE standards for adaptive instructional systems (AISs). AISs are computer-based systems that guide learning experiences by tailoring instruction and recommendations based on the goals, needs, and preferences of each learner in the context of domain learning objectives. Recently, the IEEE Learning Technologies Steering Committee (LTSC) formed a 6-month Standards Study Group to investigate the possible market need for standards across AISs. Several interactions with stakeholder communities point to broad interest in AIS standards, but there are several barriers to their adoption. The most significant challenge is the interdisciplinary nature of these systems and their complexity. This paper discusses the IEEE standards process and provides insight into the types of problems that standards should help solve.

Keywords: Adaptive Instructional Systems (AISs), Learning Technologies Standards Committee (LTSC), IEEE standards, interoperability.

1 Standards

Standards are all around us. We take for granted standardized weights and measures; standards that enable our computers to work; standards that ensure our food won't kill us; and standards that enable us to plug appliances into electric outlets without a thought. Outside of the byzantine world of standards development organizations (SDOs), however, not much consideration is given to how and when standards are (or should be) developed and how to judge their success. This paper starts with these issues.

1.1 What Standards Do

To understand why and how standards are developed, we must first understand what they do. In a nutshell

> (1) *Standards solve marketplace problems – usually involving a supply chain of goods or services – by improving interoperability, quality, or convenience.*

As examples:

- Technical specifications enable parts to be manufactured, and related services to be offered, by anyone in possession of the specifications. This supports (and often creates) a supply chain for parts and services and, if the specifications are public and not proprietary, enables competition and consequently leads to market efficiencies.
- Process standards such as ISO 9001, provide quality assurance that engenders trusts among trading partners and reduces the inefficiencies caused by poor parts, missed deadlines, and cost overruns.
- Environmental, occupational, and other safety standards increase consumer trust in products – which is necessary for sales – and reduces the human, societal, and other costs of the manufacturing process. Standards concerning data privacy and ethical applications of artificial intelligence (AI) [1] have similar effects.
- Standardized labels serve as shorthand for features and functionality that consumers can understand. For example, when a consumer buys a "USB 3.0 Bluetooth Mouse" they do not need to understand USB or Bluetooth to judge whether and how the mouse will work with their computer.
- The Open System Interconnection (OSI) model, developed as the ISO 35.100 series of standards in the 1970s, defines network layers (physical, data link, network, transport, session, presentation, and application) that are widely used in product manuals, purchasing requirements, engineering courses, and elsewhere [2]. Standards like OSI provide manufacturers, consumers, researchers and users with shared conceptual model, guiding system designs and improving communication in the market.

1.2 What is a Successful Standard?

If we accept (1), it follows that

> *(2) Standards are successful when they facilitate expansion, remove friction, lower costs, ensure safety, add stability, or otherwise positively impact their target market or supply chain.*

It also follows that a standard must achieve a reasonable level of adoption in order to succeed, and since adoption is not only a necessary condition for impact but (presumably) a strong indication that a standard provides value, we can conclude that:

> *(3) The key measure of the success of a standard is adoption in a relevant market.*

Market adoption may be a good measure of success, but it does not imply that a standard is "good." Standardization can have negative as well as positive effects, including reducing the variety and scope of products and services available and stifling innovation. The most egregious instances occur when a single company or organization uses standards to impose their approach or product on a market in an anti-competitive fashion, but even well-intentioned standards that have been adopted through a fair and open process can lead to inferior or limiting solutions and have unintended consequences. A good example from learning technology is the Advanced Distributed Learning (ADL) Initiative's Shareable Content Object Reference Model (SCORM) [3] and the IMS Global Learning Consortium (IMS) [4] and IEEE Learning Technology Standards Committee (IEEE LTSC) standards it incorporates [5].

At the time SCORM was developed, it was assumed that learning would be managed and delivered by an LMS. Moreover, the core IEEE LTSC standard adopted by SCORM, derived from earlier work of the Aviation Industry CBT Committee (AICC), assumed that training would be delivered to one learner at a time in a sequential set of "chunks" (called assignable units in the AICC specifications and Shareable Content Objects, or SCOs, in SCORM). Finally, reflecting the way courseware was typically developed by instructors (in academia) and instructional system designers (ISDs, in industry and the military), the IMS Simple Sequencing specification used in SCORM assumed that each content object would define its own learning objectives and transmit them to an LMS.

On one hand, it is fair to say that SCORM and related standards were incredibly successful. They achieved global adoption, accelerated the proliferation of eLearning across education and training, and spawned a multi-billion-dollar industry [6]. On the other hand, SCORM's assumptions about managed learning and local delivery became out of date over the years. Evolving technology caused technical problems and served to inhibit the adoption of innovations like competency-based education and training, which require data about the learner's current competence and learning objectives be shared across learning activity providers, both local and cloud-based.

1.3 How are Standards Developed?

Finally, before discussing the IEEE and AIS standards, we address the question of how standards are developed.

In this regard, we are particularly interested in formal standards that are developed by SDOs using a transparent process rather than de facto standards that are created by individuals, companies, or other organizations and that become standards through adoption. Formal standards – and especially "de jure" standards that are developed by accredited SDOs – have two advantages over "de facto" standards. First and foremost, formal (de jure) standards are developed using a process that is transparent, open, consensus driven, and designed to be fair to all relevant stakeholder groups. This does not guarantee market relevance, technical merit, or adoption, but it helps facilitate all three and it prevents misuse of the standards for the benefit of a single organization or stakeholder group. The second advantage to de jure standards is that, as the term suggests, they have more legal weight than de facto standards, and they often have more commercial weight. When a standard is essentially proprietary and has no governance structure, it is difficult to use as the basis of regulatory requirements and it can be risky to require conformance in a purchasing process.

To summarize, the standards on which we wish to focus are AIS standards that are:

(4) *Developed using an open, consensus process that involves and is fair to all stakeholder groups, including designers, producers, buyers, and users of adaptive (and other) instructional systems, as well as the broader learning science and learning engineering communities;*

(5) *Likely to be adopted and to facilitate the growth and application of AIS; and*
(6) *Unlikely to stifle innovation or place unnecessary (and foreseeable) limitations on AIS or their dissemination and use.*

1.4 Guiding Principles

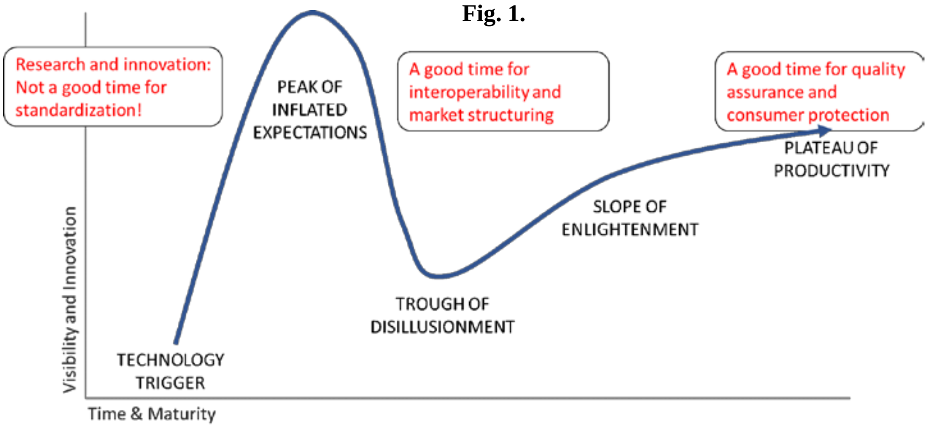
Among the above three criteria, (4) is relatively easy to meet. As will be described later, the IEEE provides the process and structure for it, and is well-suited to do so. Ensuring (5) and (6) is more difficult, but there are observations about standardization that can be used as guiding principles.

Hype Cycle Guidance. The first observation is that standardization is often tied to and spurred by innovation. As argued by Robson [7], the innovations relevant to AIS include the cloud (and SaaS) computing and business models; the ascendance of mobile, AR/VR and other platforms that produce more varied and voluminous data; and the maturation of artificial intelligence (AI) and its applications in the form of educational data mining / learning analytics.

The next observation is that innovations typically follow the Gartner hype cycle [8]. As expressed in Fig. 1 (next page), it is generally a poor choice to produce standards before an innovation stabilizes, i.e. before the hype cycle is on an upward swing towards the “slope of enlightenment” and “plateau of productivity.” Prior to then, markets are dominated by research and innovation and are attempting to discover which of the many possible applications of an innovation will produce the most value. SDOs that attempt to develop standards at this point run a high risk of favoring the wrong approach and creating an anti-competitive atmosphere that discourages the development of new products and services with “secret sauces” based on the innovation in question.

As an innovation emerges from the “trough of disillusionment,” high value applications start to emerge, and market structure starts to crystalize. At this point, it makes sense for SDOs to explore interoperability and to start to develop standards that help define product categories. Later, when the innovation is mainstream, and after a healthy

Fig. 1.



Research and innovation: Not a good time for standardization!

A good time for interoperability and market structuring

A good time for quality assurance and consumer protection supply chain has developed, SDOs may wish to consider quality standards and standards that protect consumers, both as a benefit to humanity and to establish the trust needed for further market expansion.

Scientific Research and Standardization. One strength of the IEEE is its large community of researchers who can help standards developers avoid unforeseen consequences, provide test data, and contribute to the technical underpinnings of standards. Standardizing the results research projects, how- ever, is not usually a good idea because there are fundamental incompatibilities be- tween research and standards. In particular, real-world markets are based on the mundane repetition of the same solution and have little appetite for abstraction or generalization. Research, in contrast, is focused on the continual discovery of new solutions, new knowledge, new ideas, and new approaches. Moreover, research seeks to discover the ideal, whereas standards operate on the principle that the "perfect is the enemy of the good" [9]. In the experience of the authors, these differences in mindsets – and the failure to recognize them – have caused significant delays in the development of learning technology standards and have led to standards that have great intellectual merit but have not achieved any significant adoption.

Avoiding Dominance. In standardization, "dominance" refers to domination of the standards process by a single organization or stakeholder group. The policies and procedures of SDOs like the IEEE include provisions to prevent dominance, but there are naturally occurring forms of dominance that process alone cannot prevent. In the learning technology field, natural dominance results from the divisions between education and training, and between military and civilian applications. These divisions have resulted in remarkably non- intersecting supply chains and hence in standards that are extensively used in one com- munity and hardly at all in another. There is nothing wrong with developing standards that, for example, apply only to military training or to academic LMSs. However, if the goal of AIS standards is to be crosscutting, since the same technologies are used in all segments, it is important that all intended communities participate and contribute to their development.

2 Potential AIS Standards

In this section we review four of the proposed AIS standardization ideas: common AIS conceptual model, common learner record features, component interoperability and reuse, and validation standards.

2.1 Common AIS Conceptual Model

AISs are computer-based systems that guide learning experiences by tailoring instruction and recommendations based on the goals, needs, and preferences of each learner in the context of domain learning objectives [10]. Adaptive instructional systems imply intelligence. Personal assistants, personalized drill and practice products, and intelligent tutoring systems are accepted subgroups of AISs. It is generally accepted that an ITS has four major components [11-17]: The domain model, the learner model, the instructional or tutoring model, and the user-interface model.

Terms used in the AIS and ITS lexicons tend to have more than one meaning, and several terms are often used to describe the same thing. For example, the word *context* is often used loosely to describe the state of an AIS at a particular instant in time, but depending on who is using the term, may include or not include specific descriptions of the state of instruction. Before we can support interoperability between the four common components mentioned above, we need to understand their scope, define them, and identify their functions as part of a reference or concept model, i.e. as part of "a domain-specific ontology consisting of an interlinked set of clearly defined concepts produced by an expert or body of experts in order to encourage clear communication" [18].

A hierarchical common understanding of the composition of AISs would help researchers and engineers communicate designs and ideas in lectures, presentations, and technical papers as well as in system specifications. While we expect some debate, it seems that a common conceptual model of AISs is low hanging fruit and is needed prior to defining component interoperability, which is our next potential area for standardization.

2.2 AIS Component Interoperability and Reuse

If an AIS conceptual model can be adopted as a standard, the ability to create a model of interoperability between AIS component might also be realizable. Sottilare and Brawner [10, 19-20] suggest that a set of common messages could be developed to facilitate communication and interoperability among the four common components mentioned previously. This is more feasible than attempting to standardize the internal functions of each of the four AIS components, which are often proprietary and contain intellectual property their developers do not wish to share, but it is likely to be more difficult than developing a common conceptual model.

A set of messages that enable data sharing, and specifications that drive component processes, can expand as needed over time to cover the various functions in the four components without impacting systems that previously adopted the standard (backward compatibility). A common set of messages also allows for replacement of components with new ones that satisfy the messaging standard. The benefit to the marketplace is the increased potential for reuse and the opportunity to select specialized components that are especially suited to the domain of instruction. For example, the Generalized Intelligent Framework for Tutoring (GIFT) [21-22] uses Merrill's component display theory [23] as a basis for authoring and pedagogy. Standardized messaging would al- low authors to easily switch to pedagogy based on Gagne's nine instructional events [24] if they felt that was a better match to the AIS's domain of instruction.

While the measures for assessment of learning and performance are likely to differ among domains, there are attributes of the learner that might be consistent in their structure and use. This brings us to our next potential area for standardization: common learner record features.

2.3 Common AIS Learner Record Features

An idea discussed during recent AIS standards workshops and meetings has been the idea that learner records contain a set of common features that could function as the basis for a default learner model [25]. Learner record features could include demographic data, historical records of experience and achievement, and a model of domain competency along with associated models of skill decay. Standard learner record features would allow systems other than the originating system to read in and interpret learner data in support of new instructional experiences. The opportunity for common learner record features including measures of current and predicted learning and performance forecasts the need for a method to validate AIS functions and their effectiveness which is our next opportunity for AIS standards.

2.4 AIS Validation Standards

Once standards have been adopted for common conceptual models, component interoperability, and learner record features, we will not only want to validate AIS compliance to those standards but will also want to test their effectiveness, their fit for purpose, and their compatibility with other learning systems. Support for other standards, such as the experience API (xAPI) must be considered, and authors of AISs may desire to evaluate the effectiveness of their systems as a whole or in part to understand how their product stacks up against marketplace expectations for performance and learning effectiveness. Early on, GIFT adopted a testbed function to support experimental evaluation of AISs to determine whether all components met validation criteria, e.g., whether a given instance of GIFT operated as a functional tutor. Our experience with GIFT may serve as a model for how we might approach validation, and therefore serve as a guide to normative language in a broader series of standards that address the quality of AISs and their compatibility with other learning systems.

3 IEEE Standards

In this next section we provide an overview of the IEEE standardization process and discuss how it can best be used to develop successful AIS standards.

3.1 Organizational Structure

The IEEE is an international professional society with more than 400,000 members in more than 160 countries. Its mission is advancing technology for the benefit of humanity. Within the IEEE, as its own organizational unit, is the IEEE Standards Association (IEEE-SA). The IEEE-SA is an accredited standards development organization that produces voluntary open consensus standards with participation from over 7,000 individual and 200 corporate members and with offices in Austria, Belgium, China, India and the United States. The IEEE-SA and its volunteers has produced over 2,000 standards, generally in areas related to the interests and expertise of IEEE members. These include many high-profile areas such as Wi-Fi, 5G, power engineering, nuclear engineering, smart cities, autonomous vehicles, and – important for us – learning technology. IEEE standards in learning technology are produced by the IEEE Learning Technology Standards Committee (IEEE LTSC).

3.2 The IEEE Standards Process

A wealth of information about the IEEE standards process is available online from IEEE-SA web sites and from the IEEE-hosted site www.standardsuniversity.org. As a brief overview:

- Development of an IEEE standard takes place within a Working Group (WG) that operates within a committee such as the IEEE LTSC.
- The WG follows a set of approved policies and procedures to develop a specification that is proposed as a draft standard. The policies and procedures must be compatible with IEEE-SA policies and procedures [26] and incorporate the principles of openness, due process, balance, right of appeal, and consensus.
- A balanced and fair "ballot group" is formed, allowing interested IEEE-SA members to comment and vote on the draft standard. Drafts are often revised in response and sent out for another round of comments and votes.
- Once enough interested members approve, and all comments are considered and addressed, the IEEE-SA conducts a formal review of the process used. Assuming that process was correctly followed the approved draft becomes an IEEE standard.

The details of this process involve many more steps than listed above, and it has features that many participants find extremely useful. For example, when making a comment on a draft, each commenter is required to suggest a revision, or else the comment is rejected. This prevents wanton criticisms with no proposed solutions. This process is what the IEEE-SA provides and monitors, while all technical decisions are the purview of the working groups. This means that it is incumbent on any AIS-related WGs to ensure the applicability, suitability, and adoptability of their standards and to solicit participation that is sufficiently broad, knowledgeable, and balanced.

3.3 Standards, Recommended Practices, and Guides

In general, standards have at least some normative portions, i.e. portions that define verifiable conformance criteria. In standards, conformance criteria are identified by their use of the verb *shall*, as in the sentence:

`A password shall be at least eight characters in length and shall contain at least one capital letter, one lower-case letter, one number, and one punctuation mark.`

Standards may also have informative language that explains the context of a standard or, in some cases, provides direction and choices. Formal standards identify directional statements by their use of the verb "*should*" and choices by their use of the verb "*may*." Changing the first "*shall*" in the sentence above to "*should*" would, for example, relax the requirement that a password be at least eight characters in length, although it would recommend it. Changing the first "shall" to "may" would say that a password is permitted to be at least eight characters in length, without requiring or recommending it.

The reasons for discussing the above is that IEEE standards do not have to be "standards" in the sense that they specify a set of requirements that must be met. They can also be *recommended practices*, which are characterized by the verb "should" or *guides* that provide several alternative choices and make heavy use of the verb "may." An AIS guide might, for example, recommend that assessments of learner competencies *should* be reported and provide several alternative protocols that may be used to report them.

4 Conclusions

This paper reviewed the IEEE process for the adoption of standards along with marketplace drivers for developing standards. We also discussed the potential of four AIS standards proposals: common AIS conceptual model, common learner record features, component interoperability and reuse, and validation standards. Based on discussions that have been held in AIS workshops, and on the marketplace rationale for putting forth standards (e.g., cost, affordability, usability, return-on-investment), we believe that these four proposals have the greatest chance of positively affecting the authoring, delivery, management, and evaluation of AISs.

5 References

1. IEEE. (2018). Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems. Version 2 (for Public Discussion) Retrieved 30 April 2018 from https://ethicsinaction.ieee.org/
2. Wikipedia. (2018). OSI model. Retrieved 29 April 2018 from https://en.wikipedia.org/wiki/OSI_model.
3. Advanced Distributed Learning (ADL) Initiative. (2004). The Shareable Content Object Reference Model (SCORM) web site. Retrieved 30 April 2018 from http://adlnet.gov/scorm.
4. IMS. (2003). IMS Simple Sequencing Specification. Retrieved 30 April 2018 from https://www.imsglobal.org/simplesequencing/index.html.
5. Robson, R., & Barr, A. (2013). Lowering the Barrier to Adoption of Intelligent Tutoring Systems through Standardization. Design Recommendations for Intelligent Tutoring Systems: Volume 1-Learner Modeling, 7 - 13.
6. Reuters. (2017). Global E-Learning Market 2017 to Boom $275.10 Billion Value by 2022 at a CAGR of 7.5% – Orbis Research. Retrieved 29 April, 2018 from https://www.reuters.com/brandfeatures/venture-capital/article?id=11353.
7. Robson (2018). Learning Technology Standards – A New Awakening. To appear in
8. *GIFTSYM6,* edited by Sottilare, R.
9. Gartner. (2018). Gartner Hype Cycle. Retrieved 30 April 2018 from https://www.gartner.com/technology/research/methodologies/hype-cycle.jsp.
10. Wikipedia. (2018). Perfect is the Enemy of the Good. Retrieved 30 April 2018 from:https://en.wikipedia.org/wiki/Perfect_is_the_enemy_of_good.

11. Sottilare, R. & Brawner, K. (2018, March, *in press*). Exploring Standardization Opportunities by Examining Interaction between Common Adaptive Instructional System Components. In Proceedings of the *First Adaptive Instructional Systems (AIS) Standards Workshop*, Orlando, Florida.
12. Elson-Cook, M. (1993). Student modeling in intelligent tutoring systems. *Artificial Intelligence Review*, 7, 227-240.
13. Nkambou, R., Mizoguchi, R. & Bourdeau, J. (2010). *Advances in intelligent tutoring systems*. Heidelberg: Springer.
14. Graesser, A.C., Conley, M. & Olney, A. (2012). Intelligent tutoring systems. In K.R. Harris,
15. S. Graham & T. Urdan (Eds.), *APA Educational Psychology Handbook: Vol. 3. Applications to Learning and Teaching* (pp. 451-473). Washington, DC: American Psychological Association.
16. Psotka, J. & Mutter, S.A. (1988). *Intelligent Tutoring Systems: Lessons Learned*. Hillsdale, NJ: Lawrence Erlbaum Associates.
17. Sleeman D. & J. S. Brown (Eds.) (1982). *Intelligent Tutoring Systems*. Orlando, Florida: Academic Press, Inc.
18. VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems and other tutoring systems. *Educational Psychologist*, 46(4), 197-221.
19. Woolf, B.P. (2009). *Building intelligent interactive tutors*. Burlington, MA: Morgan Kaufmann Publishers.
20. Wikipedia. (2018). Reference Model. Retrieved 1 May 2018 from: https://en.wikipedia.org/wiki/Reference_model.
21. Brawner, K. & Sottilare, R. (2018, June, *in press*). Proposing Module-level Interoperability for Intelligent Tutoring Systems (ITSs). In the Exploring Opportunities to Standardize Adaptive Instructional Systems (AISs) Workshop of the *19th International Conference of the Artificial Intelligence in Education (AIED) Conference*, London, England, United Kingdom, June 2018.
22. Sottilare, R. & Brawner, K. (2018, June, *in press*). Component Interaction within the Generalized Intelligent Framework for Tutoring (GIFT) as a Model for Adaptive Instructional System Standards. In the Adaptive Instructional System (AIS) Standards Workshop of the *14th International Conference of the Intelligent Tutoring Systems (ITS) Conference*, Montreal, Quebec, Canada, June 2018.
23. Sottilare, R.A., Brawner, K.W., Goldberg, B.S. & Holden, H.K. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). Concept paper released as part of GIFT software documentation. Orlando, FL: US Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED). Retrieved from: https://gifttutoring.org/attachments/152/GIFTDescription_0.pdf
24. Sottilare, R., Brawner, K., Sinatra, A. & Johnston, J. (2017). An Updated Concept for a Generalized Intelligent Framework for Tutoring (GIFT). Orlando, FL: *US Army Research Laboratory*. May 2017. DOI: 10.13140/RG.2.2.12941.54244.
25. Merrill, M. D. (1983). *Component display theory*. Instructional-design theories and models: An overview of their current status, 1, 282-333.
26. Gagne, R. M. (1985). *The conditions of learning and theory of instruction* (4th ed.). New York: Holt, Rinehart & Winston.
27. Sottilare, R. & Brawner, K. (2014, May). A Persistent Learner Model to Drive Optimal Macro-Adaptive Decisions by Intelligent Tutoring Systems. Florida Artificial Intelligence Research Society, Pensacola, Florida, May, 2014.
28. IEEE. (2017). IEEE-SA Standards Board Operations Manual. Retrieved 30 April 2018 from https://standards.ieee.org/develop/policies/opman/sb_om.pdf.

Proposing Module-level Interoperability for Adaptive Instructional Systems

Keith Brawner and Robert Sottilare

Army Research Laboratory
`{keith.w.brawner.civ;robert.a.sottilare.civ}@mail.mil`

Abstract. The core components of a tutoring systems appear to be widely agreed upon. As an example, Murray's seminal review work from 1999 included the components of the student interface, domain model, teaching model, and student model [1]. Woolf's 2011 review of ITS divides the space into models of student knowledge, domain knowledge, tutoring knowledge and the communication of knowledge [2]. VanLehn's review work describing the behavior of tutoring systems mentions "that although tutoring systems differ widely in their task domains, user interfaces, software structures, knowledge bases, etc., their behaviors are in fact quite similar", clearly viewed the world through a lens of finite modeling paradigms [3]. The author has worked significantly on the Generalized Intelligent Framework for Tutoring (GIFT) system, which includes models of a domain, pedagogical, learner, and external interface explicitly. It seems clear that each Intelligent Tutoring System (ITS) must, at a minimum, include a model of the domain of instruction, a model of the learner that it is instructing, and a model of instruction to deliver. This paper discusses component-level standardization for data interchange as well as proposing that the base components are the one which have been widely agreed upon.

Keywords: Standards, Interoperability, Adaptive Instruction, Intelligent Tutoring Systems

6 The interchangeable parts of an ~~ITS~~ AIS

The abstract discusses the basic, widely-known, parts of a standard ITS; models of the domain, leaner, pedagogy, and interface. Naturally, these models can be as simple of a set as "one algebra problem" (domain model), "Right/Wrong" (learner model), and "Do it until you get it right" (instructional model). This can be as complicated as an interactive scenario with varying levels of difficulty and feedback (domain model), a lifelong history of all learner activities with assessment of competencies (learner model), and a system of feedback timing and difficulty adjustment (instructional model). While these core components of an ITS are somewhat agreed upon, few organizations outside of the Army Research Laboratory have constructed greater than 50 ITSs from a common core of components [4]. This would be a great engineering feat if not for the use of interchangeable components.

Interchangeable components allow for a few significant goals to be accomplished. The first of these is the fair evaluation of one component against another. For individual learners, is a system of Socratic dialogue (ala AutoTutor) or direct immediate feedback (ala Cognitive Tutor) the most appropriate action to take? For low motivated learners? For prior domain experts? These interesting research questions are enabled via interchangeable components. It is not possible to answer these types of research questions without crafting a whole system, at the moment. The second of these goals is to provide a common basis on which to construct authoring tools. A set of authoring tools which can construct many tutors obviates the need for both tutor construction and component construction.

6.1 Adaptive Instructional Systems (AISs)

The term "Adaptive Instructional System" (AIS) encompasses many of the different types of systems which exist within the field. It encompasses, but is not limited to, the ITS family of systems. The authors are proposing the use of the AIS term to help to define standards among systems which interchange the same types of information - a pedagogical engine is not an ITS, but is a part of an AIS; an adaptive test is not an ITS, but can conform to AIS standards; a indexed content repository with various metadata is a component of an AIS and should be compliant with standards. The AIS term is broad enough to encompass all systems with a stated goal of enabling both a) adaption, and b) instruction, based on the learner.

Previous work has discussed basing design decisions around the idea of the Lowest Replaceable Unit (LRU), the lowest level of compliance [5]. This idea is roughly equivalent to ideas in other fields, such as the "Unit" from software engineering, an item which can be independently tested at the input/output level for conformance to desired behavior, in the software standards community [6]. Similarly, in the early business stage software community, they describe the Minimum Viable Product (MVP) as the smallest unit which produces a market-acceptable service (i.e., "scratches an itch"). AISs are composed of many of these base-level components, and although there is debate among what the base level is, its description is not – it is the minimum testable and viable software module or service.

7 Case Study: Interchangeable Instructional Models and Modules

As the GIFT system has been developed, it has started with simplistic models and experimented with more complex models in each of the model categories. As an example, the initial models served merely as "pass-through" models – the Learner Module forwarded all of its information to the Pedagogical Module without meaningful processing. Later, the learner model was reaching out to external Learner Record Stores (LRS) [7] systems in order to aggregate learner performance, and recommending content in an instructionally simplistic manner [8]. In the current GIFT release, this information is used in more complex manners, including the injection of content mid-lesson [9].

The GIFT production system has additionally changed over time. The initial instructional model, like the Learner Module, consisted of "pass-through" functionality. This was quickly exchanged for a model which used information about the learner in order to drive content selections, while still operating in an error-sensitive feedback manner within simulated environments [10]. This 2012 model, based on Component Display Theory[11], has since been replaced with a model which performs similar actions, but tracks its past actions over time in order to learn which content types of an individual learner/course make the largest difference [12], based on the Interactive/Constructive/Active/Passive remediation framework [13]. Because of the nature of interchangeable parts, each of these models is still usable in the current system – a system designer can choose among these models to use without making *any* changes to the models of the learner, models of the content, or models of delivery. The internal evidence is suggesting that the reinforcement-based models are having the highest performance *in situ*, and have been set to the default model types for newly created content.

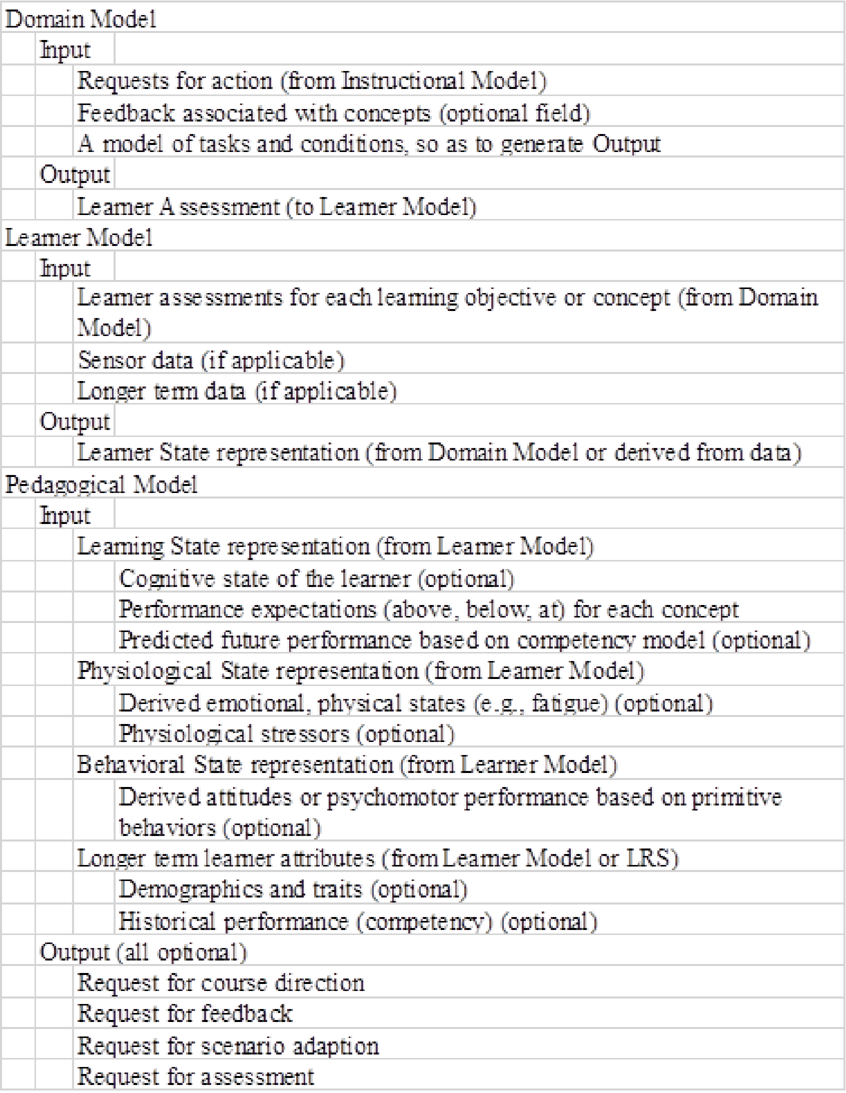
The ability to interchange models has significant advantages. It allows for the direct comparison of models. As an example, such a system allows for the answering of the question: "What would the effect be if the Cognitive Tutor were to ask introspective questions in the manner of AutoTutor instead of selecting problems?" This allows for the selection of a "best of breed" model for each of the tasks. Next, it enables significant levels of reuse – one model can be used for many systems. Finally, it allows for the creation of authoring tools which can author content, independent of the other models - an instructional model can be constructed without knowledge of the domain model and hence be useful for a wide variety of training applications; a domain model can be created with standard tools without particular concern about the motivation levels or prior experience of the learners (handled by the learner model and instructional model, respectively). The abilities to compare, reuse, author, and encapsulate complexity drive down overall cost and development time.

8 Where to Start?

GIFT adopted the design principle of separating content from executable code [14]. This is more than simply the separation of domain content from the instructional content – but multiple layers of separation of logic. Modules (operational software processes) are separated from the data they process in either through the use of configuration information (i.e., domain content, sensor configuration settings), restrictions on the types of data input and output (i.e., interface specifications), or software library calls (i.e., an outside call to a library to provide assessment of student information). This provided a starting point of the GIFT system and the authors believe that the evolving messages and standards provide a natural starting point for standardization.

We recommend an analysis of the information required by AIS common components to begin formulating an ontology as a basis for selecting candidates for standard messages. Given that there are a number of relatively finite components, the natural question is "where do we start?" Based on experience with GIFT and in reviewing the literature, Table 1 lists some of the most frequently recurring examples, but is intended to be illustrative, not exhaustive. Table 1 provides a starting place for component-level standardization discussion, and invites comment. This initial table of items provides the starting point from which GIFT has built upon.

Table 1. Table of proposed messages for initial module-level interoperability



Domain Model			
	Input		
		Requests for action (from Instructional Model)	
		Feedback associated with concepts (optional field)	
		A model of tasks and conditions, so as to generate Output	
	Output		
		Learner Assessment (to Learner Model)	
Learner Model			
	Input		
		Learner assessments for each learning objective or concept (from Domain Model)	
		Sensor data (if applicable)	
		Longer term data (if applicable)	
	Output		
		Learner State representation (from Domain Model or derived from data)	
Pedagogical Model			
	Input		
		Learning State representation (from Learner Model)	
			Cognitive state of the learner (optional)
			Performance expectations (above, below, at) for each concept
			Predicted future performance based on competency model (optional)
		Physiological State representation (from Learner Model)	
			Derived emotional, physical states (e.g., fatigue) (optional)
			Physiological stressors (optional)
		Behavioral State representation (from Learner Model)	
			Derived attitudes or psychomotor performance based on primitive behaviors (optional)
		Longer term learner attributes (from Learner Model or LRS)	
			Demographics and traits (optional)
			Historical performance (competency) (optional)
	Output (all optional)		
		Request for course direction	
		Request for feedback	
		Request for scenario adaption	
		Request for assessment	

8.1 Case Study Example: Hinting amongst AutoTutor, Cognitive Tutor, and GIFT

As an example, consider the behavior of somewhat disparate systems – AutoTutor, Cognitive Tutor, and GIFT. Each of them provides very different models of the student, instruction, domain, and other items. However, at the base level, each of the systems has a "last piece of feedback" item built into the system. In Cognitive Tutor, this is called the "bottom out" hint within its hinting model [15]. In AutoTutor, if the student has not responded to Socratic question asking over the course of the tutorial session, the student is given a hint, prompt, pump, and eventually the "assertion" [16]. In GIFT, its default behavior is to repeatedly give the final hint within a hint sequence [17]. The standard representation, presented above, allows for each of the systems AutoTutor to output the final level of feedback (bottom out, assertion, repeated hinting, respectively) as part of a unified standard. While these systems are different in their design, intent, and domains of instruction, their actual behavior can be well-represented within standardized communication protocols.

9 Discussions

A typical "shell tutor" can be used to create a significant amount of content which conforms to a relatively finite number of templates, such as "the vast majority of mathematics problems" in the case of the Cognitive Tutor. GIFT, however, is a tutoring architecture that supports the authoring of ITSs, the delivery and management of adaptive instruction, and the evaluation of ITSs and adaptive instructional capabilities for nearly any task domain (e.g., cognitive, affective, psychomotor, social). It can be, and has been, used to tutor or adjust on the above described uses cases – simulations, content, and experiments. It has done so through the engineering of interchangeable parts and standardized interfaces. The authors believe that these interfaces can provide a starting point for other ITSs, and have previously worked to communicate domain information with AutoTutor, Cognitive Tutor, and Betty's Brain systems; there is reasonable confidence that these interfaces are sufficient to enable relatively advanced tutoring techniques.

9.1 Market Opportunities Created

The current business practice, with a lack of standards, forces the monolithic creation of a single AIS system. While this single AIS system may make use of a variety of standards (i.e., JSON, XML, TCP/IP, LTI reporting standards, etc.), the end result is that its components are not interchangeable. The creation of standards enables new business practices and market opportunities.

The basic market opportunities that standardization at this level creates are, roughly, one business model per module of standardization. In this manner, a business case for a "vendor-supplied" instructional engine is created. The business for the learner model is mostly as an aggregation service with an emphasis on traits relevant to learning, rather than the traditional model of consumer preferences. The business for the domain model is in the sale of content which can train individuals for their tasks. Further, the creation of individual models creates new business models which service the needs of the businesses based on the previous ones – content aggregators for learner model data provisioning, system-creators for selection of the appropriate types of models relevant to a task domain, certification businesses for certification that the models interchange as specified, analytic services for existing data reprocessing, etc. If the reader is curious on the kinds of educational businesses which may come into existence around a standard, they need only consider the number of vendors which make use of SCORM [18] for content, or xAPI for learner modeling information logging [7].

References

1. Murray, T., *Authoring intelligent tutoring systems: An analysis of the state of the art.* International Journal of Artificial Intelligence in Education (IJAIED), 1999. **10**: p. 98-129.
2. Woolf, B., *Intelligent Tutors: Past, Present, and Future.* Keynote address at the Advanced Distributed Learning ImplementationFest, 2011.
3. VanLehn, K., *The behavior of tutoring systems.* International Journal of Artificial Intelligence in Education, 2006. **16**.
4. Brawner, K., A.M. Sinatra, and R. Sottilare, *Motivation and research in architectural intelligent tutoring.* International Journal of Simulation and Process Modelling, 2017. **12**(3-4): p. 300-312.
5. Sottilare, B. and K.W. Brawner, *Exploring Standardization Opportunities by Examining Interaction between Common Adaptive Instructional System Components*, in *Adaptive Instructional Systems Workshop*, B. Sottilare, Editor. 2018, IEEE: Orlando, FL.
6. Committee, I.S., *Standard Glossary of Software Engineering Terminology*, in *IEEE STD 610.12-1990 IEEE*. 1990, IEEE: IEEE.
7. Regan, D.A. *The Training and Learning Architecture: Infrastructure for the Future of Learning*. in *Invited Keynote International Symposium on Information Technology and Communication in Education (SINTICE), Madrid, Spain*. 2013.
8. Brawner, K. and S. Ososky. *The GIFT 2015 Report Card and the State of the Project.* in *Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym3)*. 2015. Orlando, FL.
9. Folsom-Kovarik, J.T. and M.W. Boyce. *Developing a Pattern Recognition Structure to Tailor Mid-Lesson Feedback.* in *Proceedings of the 5th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym5)*. 2017. Robert Sottilare.
10. Goldberg, B., et al. *Use of Evidence-based Strategies to Enhance the Extensibility of Adaptive Tutoring Technologies*. in *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*. 2012. Orlando, FL: NTSA.

11. Merrill, M.D., *Component display theory*. Instructional-design theories and models: An overview of their current status, 1983. **1**: p. 282-333.
12. Rowe, J., et al. *Extending GIFT with a Reinforcement Learning-Based Framework for Generalized Tutorial Planning*. in *Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym4)*. 2016.
13. Chi, M.T. and R. Wylie, *The ICAP framework: Linking cognitive engagement to active learning outcomes.* Educational Psychologist, 2014. **49**(4): p. 219-243.
14. Patil, A.S. and A. Abraham, *Intelligent and Interactive Web-Based Tutoring System in Engineering Education: Reviews, Perspectives and Development*, in *Computational Intelligence for Technology Enhanced Learning. Studies in Computational Intelligence*, F. Xhafa, et al., Editors. 2010, Springer-Verlag.: Berlin. p. 79-97.
15. Aleven, V., et al., *Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor.* International Journal of Artificial Intelligence in Education, 2006. **16**(2): p. 101-128.
16. Graesser, A.C., et al., *AutoTutor.* Applied natural language processing: Identification, investigation, and resolution. Hershey, PA: IGI Global, 2012.
17. Sottilare, R.A., et al., *The Generalized Intelligent Framework for Tutoring (GIFT).* 2012.
18. Advanced Distributed Learning Initiative, *Sharable Content Object Reference Model (SCORM™).* Advanced Distributed Learning, http://www. adlnet. org, 2001.

A Computational Perspective of Adaptive Instructional Systems for Standards Design

Vasile Rus, Arthur C. Graesser, Xiangen Hu, Jody L. Cockroft

The University of Memphis
vrus@memphis.edu

Abstract. In this paper, we propose a novel perspective of the architecture of adaptive instructional systems (AISs) with the goal of identifying common components and data exchange protocols among those components. The novel view is inspired by open architectures of computational systems. We hope this novel view will inform the adoption of a common architecture that promotes innovation and standardization, plug-and-play architectures, and reduced entry barriers for newcomers while protecting the intellectual property of the various players. Our design makes no ontological commitment which brands the design generally applicable to changes in semantics of core elements of adaptive instructional systems such as knowledge components.

1 Introduction

Adaptive instructional systems (AISs) are advanced education technologies that provide tailored instruction to learners of all ages. Standardizing various aspects of AISs would be an important step towards more widespread deployment and use of such systems. This paper intends to contribute to this conversation on AISs standardization. To this end, we propose a novel perspective of the architecture of AISs with the goal of identifying common components and data exchange protocols among those components. Our perspective is inspired from open architectures of computational systems, namely Personal Computer systems. We hope this novel view will inform the adoption of a common architecture that promotes innovation and standardization, plug-and-play architectures, and lowers barriers for entry for newcomers while protecting the intellectual property of the various players.

We will address the following two core issues when it comes to standardization: (i) adopting a common architecture and (ii) designing interface protocols to facilitate communication among the main components of the common architecture as well as with external components and systems or users. Identifying a common, widely accepted architecture is essential for any thriving area/industry as it promotes healthy growth and progress by enabling innovation while also protecting the Intellectual Property (IP) of commercial enterprises. For instance, newcomers proposing novel ideas and products could easily enter the market by developing solutions for just a component or key function of AISs. As long as their product meets the standardized interface protocol for that component, users could easily use the novel solution in their existing AIS - all they need to do is switch the old component with the new one and their AIS should run smoothly, eventually better due to the novel solution in the replaced component.

Our approach to identify common components in AISs is based on a functional analysis as well as a market analysis, i.e., we explore what market players focus on in terms of commercial products. If particular market players focus on one task/function, such as assessment, that might be an indication that there should be an independent module or component dedicated to that task in the standardized, widely accepted AIS architecture.

As mentioned, we adopt in our analysis a computational perspective that considers the major computational steps involved in an AIS, as explained in detail later. Additionally, we draw inspiration from previously developed open architectures of computational systems for reasons which we briefly present next. For instance, the open, standardized architecture of modern personal computers (PCs) led to the creation of a healthy ecosystem of various developers that offer a variety of alternative products for components or full systems resulting in a wide selection of choices available to users, i.e., businesses and individuals. This in turn led to the widespread adoption of PCs which enabled other major breakthroughs such as the Internet and the World Wide Web and so on. Indeed, open, standardized PC architectures made it possible for various players to focus on various aspects of a PC. A hard drive manufacturer could focus on just that, making hard drives for PCs, as long as their hard drives complied with one of the major external device interface standards such as the Integrated Device Electronics (IDE), Serial Advanced Technology Attachment (or Serial ATA), or Small Computer System Interface (SCSI). A PC vendor or end user could buy a hard drive of her choice, e.g., a larger or faster one from a vendor of their choice, and then install it on their PC using a simple plug-and-play procedure. The PC would be fully functional without any further adjustments. We envision a similar future for AISs where an open architecture of major components are widely recognized and accepted by the various stakeholders and interfaces among those components are standardized to a level where a healthy ecosystem of players evolves for the benefit of all learners, the entire society, and the economy.

Another key aspect we rely on from the PC architecture is the existence of a central processing unit (CPU). We will assume the existence of such a core component in AISs because there needs to be a component that orchestrates the overall operation of the system. Furthermore, the existence of such a main or “conductor” component allows us to better describe the functionality, core components, and related operations needed in all AISs. Based on our analysis, the pedagogical module in the GIFT architecture (Sottilare, Brawner, Goldberg, & Holden, 2012) could play this role. However, if this option is adopted, then the pedagogical module will have to carry tasks such as authenticating the learner which might be outside the scope of such a module. Alternatively, the proposed main or “conductor” component may correspond to a component whose role is to just run continuously the two loops as described in VanLehn’s two-loop architecture for ITSs (VanLehn, 2006) extended with the additional functionality of user authentication and other session management tasks.

Besides the PC world, our work was also informed by existing research efforts in AISs and related areas. Indeed, we drew inspiration from previously proposed AIS architectures, such a VanLehn's two-loop architecture (VanLehn, 2006), Sottilare and colleagues' GIFT architecture (Sottilare, Brawner, Goldberg, & Holden, 2012), and our own AISs' architectures (Rus, D'Mello, Hu, & Graesser, 2013). We also considered the recent suggestions for AISs standardization in Sottilare and Brawner (2018).

The other core issue of standardizing systems is defining interface protocols among the main components in the common architecture. A key challenge for this task in the area of AISs is that many components and the underlying representations they use, e.g. a set of knowledge components (KCs) to represent students' knowledge level with respect to a domain, are domain and to some degree vendor specific. Indeed, a vendor may use a set of KCs while another vendor may use a very different set of KCs for the same target domain. The set of KCs could even be proprietary, being part of the vendors' IP or "secret sauce" and therefore less likely to be disclosed. On one hand, there is a need for openness while on the other hand there is a need for IP protection. The answer to this challenge is to develop interface protocols that are general enough such that various vendors adopt them, support proprietary elements, and are flexible enough to allow changes, refinements, and extensions without the need for vendors to invest massively in implementing updated interface protocols too often. That is, the interface protocols should be general enough to allow components to work together while at the same time be specific enough to make components from various vendors able to talk to each other in meaningful ways. Our solution, described later, is to adopt a hierarchical approach to specify units of data exchange, e.g., KCs, and use numeric ids for identifying ontologies and the KCs in those ontologies. The use of ids helps avoiding ontological commitments in the interface itself. This design results in a decoupling of the interface specification from the actual ontology specification, which is accomplished separately. Therefore, the interface protocols will focus on specifying typical operations such as querying a component for a particular unit of information such as a particular KC, references (in the form of ids) to externally specified ontologies and to individual elements in those ontologies. Other elements of the interface standards can be added such as error and status codes.

It is important to note that for the interface standardization suggestions we drew inspiration from related efforts such as Advanced Distributed Learning's Total Learning Architecture (TLA; Regan, Raybourn, & Durlach, 2013) whose goal is to promote sharing data about learners and content, mixing media and delivery methods as context changes, and sequencing recommendations, as well as IMS Global Learning Consortium's efforts to develop learning technology interoperability standards.

It is beyond to scope of this paper to offer a complete solution to the very challenging task of standardizing AISs, their components, the representations that form the logical basis of the functionality of those systems and components, and related artifacts. The ideas here are meant to provide a novel and useful perspective and some suggestions that could inform the process of standardization. The process will require significant and sustained efforts from various stakeholders including researchers, practitioners, policy makers, research and development groups, commercial entities, and standardizing bodies.

The rest of the paper is organized in three parts. We first discuss briefly the core issue of representing the learner in AISs. We then present a functional analysis of an AISs in order to arrive at a common architecture by identifying key building blocks or components (or modules - different previous reports use these various terms to identify key functional units in AISs) that are not too big and not too small so that a particular player may be willing to focus on building such components. We end the paper with Conclusions.

2 Affective, Behavioral, Knowledge, Motivational, Social, And Atomic Components

Central to any AIS is the learner representation or learner model. The AIS must have ways or at least access to ways of representing what the learner knows and feels, how they behave, what motivates them, and so on. Given all this personal information, any AIS standard discussion should address privacy and security issues. However, it is beyond the scope of this paper to address these issues. We also do not discuss the important issue of trusting the stored data about a learner due to space reasons. To tackle this issue, for instance, the use of trust methods such as blockchain should be considered and discussed during standardization processes. Instead, we focus on the core issue of modelling the learner in the best and most comprehensive way possible in order to provide them the best learning experience and outcomes.

We start with modelling what the learner knows. A widely used approach is to rely on the notion of knowledge component (KC), an atomic piece of knowledge (concept or skill) that a learner must acquire while learning to master a domain (Koedinger, Corbett, & Perfetti, 2012). KCs are typically the result of cognitive task analyses followed by validation and refinement cycles based on actual student performance data.

The set of KCs for a domain, or domain model, can be organized in more complex structures, such as parameterized prerequisite knowledge structures, which are used for, among other things, shaping students' learning trajectories, i.e., the sequence in which students explore the concepts and skills to be mastered in the target domain. The set of KCs constitutes the basis of the student model and could be as simple as specifying the mastery level on each KC at one moment in time. In the literature, this type of learner model is called the overlay model, i.e. the learner model can be viewed as an overlay of the domain model and covering the parts of the domain model the student has explored so far together with performance information, i.e., whether the student mastered or not the explored topics.

We suggest to generalize the notion of KC to account for other aspects of learning such as social, motivational, behavioral, emotional, psychomotor, and physiological. For instance, learners' emotional states can be modelled as a set of emotion components (ECs). An example of an emotion component would be frustration reflecting students' level of frustration relative to the current learning goal or instructional task at one particular moment in time. A learner could be frustrated for reasons not related to the current task which should probably be accounted for separately. Similarly, we work under the assumption that there are social components (SCs) capturing learners' social skills at one particular moment in time, motivational components (MCs), behavioral components (BCs), and so on.

A key challenge with standardizing the sets of KCs that form the basis of the domain and student models is that they must be agreed upon by the various parties willing to adhere to the standard. As indicated before, there is a challenge spawning from the conflicting goals of standards in general which, on one hand, should promote open specifications to encourage collaboration and innovation, and, on the other hand, major (commercial) developers tend to protect their IP or "secret sauces". The latter is not surprising as IP is a critical aspect of commercial enterprises that gives them a competitive advantage and ultimately assures their survival. Given these opposing forces, there are at least two solutions worth exploring: (1) a bridging mechanism or (2) an "arm's length" solution. The bridging mechanism is about developing mappings from one ontology of KCs from one vendor, to another ontology from another vendor. The bridging mechanism has been used in the Semantic Web movement to enable semantic level mapping among various proprietary data representations.

The "arm's length" solution requires significant negotiations among major stakeholders to find the right balance between what is visible and what stays invisible or proprietary. For instance, the stakeholders must decide what KCs should be reported by all players; what is not specified in this set of open KCs is by default proprietary. A more flexible solution would be to have a hierarchical specification of the various KCs (cognitive, emotional, motivational, etc.) with each level in the hierarchy providing more details. Stakeholders then can decide which level of details in this hierarchical representation they are comfortable to adopt. The visible part of the hierarchical model could play the role of a common upper ontology, which could also be used as the basis of a bridging mechanism. For instance, various vendors can automatically map their internal ontologies to the common upper ontology for interfacing purposes.

A yet another solution is for the student model structure to be fully specified and what remains proprietary is the process by which the parameters of the models are derived. To illustrate this idea, we use the prerequisite structure used by ALEKS (Assessment and Learning in Knowledge Space; Doignon & Falmagne, 1999). The actual elements and structure for a domain, e.g. Algebra, could be totally open and standardized, e.g., similar to curriculum standards adopted by various states. It is the trained parameters of the model over the prerequisite structure, e.g., the parameters of the underlying stochastic process that guides assessment and the learning trajectories of individual students, which are proprietary.

Finally, in order to keep the interface model agnostic and increase the chances of being standardized and adopted by the various players, we propose a solution that is both general and flexible and avoids any ontological commitment in the interface itself, thus enabling components to talk to each other without being bound to a particular set of KCs, for instance, through the interface. The solution relies on just specifying the core operations in the interface together with numerical ids for KC ontologies and the ids of specific KCs in those ontologies. That is, the interface will only specify an id for an ontology to be used whereas the actual specification of the ontology is separate. This separation of the interface specification and the ontology specification enables the adoption of a simple, stable, and flexible interface that doesn't require updates every time the ontology is being refined. Furthermore, this approach frees the development and refinement of the ontology from the specification of the interfaces relying on that particular ontology. The ontology could be completely changed while the interface can remain the same. The interface specification should be about standardizing operations, numerical ids for identifying the ontology in a repository of ontologies, and error and status codes (and other elements as necessary).

To illustrate the basic idea of using just references to externally-specified KCs ontologies in standardized interfaces, we show in Figure 1 a learner model component and its interface. There should be an operation available through the interface that could be used to retrieve the student performance on a particular KC. When, say, the “conductor” module of an AIS attempts to retrieve such a KC, it will send a “retrieve” command through the interface together with the learner id, the KC id, and the ontology id. The learner module would respond with a corresponding performance value, say normalized value between 0 and 1, if the operation was successful. If not, an error code will be transmitted back. Indeed, error and status codes must be defined for each standardized interface in order to indicate various outcomes for an operation. An error code may be sent when performance for a particular KC is not available in the learner model.

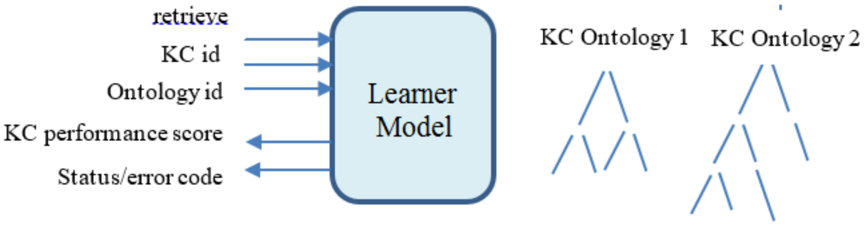


Fig. 1. Illustration of an interface standard that is ontology independent. A new ontology could be use with this interface by simply switching the ontology id when, for, instance, querying the learner model component.

The proposed ontological agnostic interface design enables vendors who cannot adhere to an ontology, for whatever reasons (e.g., legacy systems which would be expensive to change), to participate in the market and offer solutions. Such vendors can simply publish their own ontology of KCs which can then be referenced in the standardized interface as any other ontology of KCs, standardized or self-published.

As mentioned, this ontological agnostic interface design shifts the burden of specifying the KCs to the ontology designers resulting in increased flexibility, e.g., sets of KCs can be refined or new ontologies can be supported by the same interface design, and stability of the interface design, i.e., when a better set of KCs is defined or a set of KCs for a new domain is proposed, there is no need to change the interface design.

Another principle to be considered when standardizing KCs and other aspects of AISs is backward compatibility. That is, when an ontology is being refined, it should not make existing systems dysfunctional. Deployed learner models should operate as before when a new version of an ontology is being released whereas those users who want to take full advantage of the features of the refined ontology can update the learner module accordingly. Backward compatibility is another important lesson from the PC world that should be kept in mind.

3 A Functional View of Adaptive Instructional Systems

We present in this section a functional view of AISs for standardization purposes. The functional analysis and decomposition into modules or components also considers what the major players, in particular in the industry, currently offer. Industry foci are very important in defining standards because the market clearly indicates what functional aspects of AISs companies are willing to tackle. For instance, ALEKS' main focus is the development of knowledge spaces, directed graph-like structures of KCs, which serve as the domain module in ALEKS. The knowledge spaces are also used to infer learners' knowledge state based on which the next learning goal is selected. The next learning goal is suggested from the fringe, i.e. the set of concepts that students are ready (or more likely) to master next given the concepts they already mastered and the parameterized prerequisite knowledge space of the domain. It should be noted that the Cognitive Tutor AIS from Carnegie Learning, Inc., also relies on a similar concept called cognitive or skill model of a domain (Koedinger et al., 2012). This focus on modelling the domain suggest the need for a domain model component in AISs as noted by Sottilare and colleagues (2012) and Sottilare and Brawner (2018). One difference between their domain model component and ours is that we propose a decomposition of their domain module into several modules. For instance, we suggest to have a separate assessment component in the standardized AISs architecture whereas Sottilare and Brawner include assessment in the domain module or component.

Our analysis starts with a discussion of two known proposed architectures, Sottilare and colleagues' GIFT model and VanLehn's two-loop model. We then present two important interface standardization principles: finding the adequate level of component granularity and focusing first on standardizing interfaces at a logical level.

The proposed GIFT architecture includes three core modules: student model, domain model, and pedagogical model. There are other "local modules" such as the sensor module and the user interface as well as the authoring module. In our view, the authoring module is not part of the main function of any AIS. Rather, the artifacts produced by the authoring module are. We focus here primarily on the main AIS and therefore do not include the authoring module in our discussion. Nevertheless, there must be a discussion about the standardization of the authoring module or at least of the products of the authoring module, e.g., the instructional tasks, so that they could be easily imported in various AISs.

VanLehn's two-loop architecture for intelligent tutoring identifies the following main functions of such systems: instructional task selection and providing a set of instructional tasks (for the outer loop) and feedback, assessment, and next-step hints (for the inner loop). There are other functionalities mentioned such as error-specific feedback and solution review. Nevertheless, we believe these latter functionalities fit under the former categories. For instance, error-specific feedback is just a form of feedback for the special case of students making errors. Furthermore, solution review is a strategy that could be handled by the pedagogical module.

Some of the local modules in, for instance, the GIFT architecture are too coarse-grain, at least at the level of specification presented above. Industry foci indicate a finer grain level of module or component specification. To illustrate our point, we use the assessment function of AISs and note that there is a whole assessment industry that focuses on developing assessment theories, methods, and technologies. For instance, ETS (Educational Testing Services) is an educational testing and assessment organization that provides assessment methods that could be used in assessment components in AISs. Indeed, our point is that there should be an independent and visible module of the open, widely accepted AISs architecture which is dedicated to assessment such that various assessment modules produced by various vendors could be easily interchanged in a plug-and-play manner. Similarly, publishers can provide banks of instructional tasks, as described later, from which AISs can select the most appropriate task for a given instructional goal and learner at a given moment in time. Therefore, a module that acts like a database or bank of instructional tasks may be needed together with a task selection module. The task bank may be an external or non-local component that is accessed by AISs as needed. In isolated environments, e.g., a learner in a remote place with no internet connection, a local task bank would be their only option.

We believe the first phase of standardization efforts of AISs should focus on specifications at what we call the logical level. That is, the level of specification should focus on the basic logical primitives that are needed to make an AIS functional and that ignore details that are used to infer these primitives. For instance, we suggest KCs be the finest-grain size of information being considered, e.g., the affect state of a student should be specified as a set of emotional components (EC) according to some emotional ontology whereas the signals from which those ECs are derived should not be visible at this logical interface level. Indeed, at this level, there is no need to specify what facial features, voice, or physiological features were observed to infer those emotions. We should simply assume there is an external “sensor” (or complex of sensors) that connects to the AIS through a standardized external connectivity interface, described next, and sends in streams of EC data while a learner interacts with a target AISs. For this reason, we do not include in our discussion a Sensors module, which currently is specified in GIFT. Again, a Sensors module is an external component or “device” that should not be visible at the logical interface level. It is only the inferred relevant markers, e.g., the ECs that are relevant from a functional logic perspective.

All AISs need to be connected to external components (called non-local modules in GIFT) or systems such as the Learning Record Store (LRSs), which provides persistent storage for the learner model. A standardized interface to connect to external components, similar to the role of the SCSI interface for PCs, would permit easy connectivity with external devices or components and other systems. Such as external interface standard would encourage the development of augmenting components that would basically expand the capabilities of the main AIS.

We present next a brief operational view of AISs to understand the main functionality and components of such systems. This view emphasizes functions and components which we believe should be the main focus in a first discussion of standards for AISs.

Brief Operational View of AISs. When a student starts using an AIS, the system needs to authenticate and retrieve information about the student. The role of authentication is, among other things, to retrieve a unique student id which could then be used in all subsequent operations of the AIS for that particular student such as retrieving the current student model. All communication between modules related to a particular learner will eventually involve the learners' unique id. Given its central role in the functionality of AISs, user id specifications must be standardized.

Once the learner id is obtained, the AIS must retrieve the learner model, a critical step that enables tailored instruction for each learner. A related issue to be addressed is whether the "conductor" component should retrieve this information from a Learning Record Store (LRS) or simply query the learner module which in turn may retrieve the learner model or parts of the learner model from the persistent storage in the LRS. If no learner model is available, then the AIS may decide to give the learner a (pre-)test. An assessment instrument or set of assessment items could then be retrieved for this purpose from the assessment items pool, described later.

A current instructional goal could be stored in the learner model as well, which could be useful when, for instance, a learner interrupts a session and returns later. If no instructional goal is currently set, the "conductor" will call upon the pedagogical module to set the next instructional goal (see later). Once an instructional goal has been identified, the "conductor" calls upon the pedagogical module to select an instructional task or a set of instructional tasks. In fact, the "conductor" may call upon the pedagogical module to select a learner resource first. A learning resource could be a monolithic, coherent instructional resource such as a collaborative problem solving dialogue-based tutoring system. Once the learning resource has been selected only then the instructional task or set of tasks should be selected. There is the other possibility of selecting the tasks before the learning resources. That is, the instructional task or tasks are selected first before deciding which learning resource or learning paradigm to be used for delivering the task for the learner. These are all operations that must be standardized.

Once a task (and learning resource) has been selected, students will be shown the task on the interface and asked to work on it. The AIS will monitor students' work on the task, assess their performance at each step, give feedback which may include correcting misconceptions, and providing hints. In some cases, like ALEKS, there is not much within-task monitoring involved. Students are simply assessed as solving or not solving the given task. In the latter case, a worked-out solution is shown to them. An 'update the learner model' operation will need to be initiated and sent to the learner model after each task. The "conductor" component or the learner model component may call upon the assessment module to do a new performance assessment which is then recorded in the student model. The question is when an 'update the learner model' operation should be initiated and sent to the LRS. After each task or each session or an entire section/topic is being covered? There are trade-offs among the various options that need be discussed.

Also, there is a need for micro-assessment, discussed later, when there is close monitoring and feedback at each step within a task. The type and frequency of feedback and hints will be guided by the micro-strategies that the pedagogical module selects. A bank of micro-strategies may be available similar to the task bank. Alternatively, the micro-strategies can be selected by default when a learning resource is chosen.

Ideally, the within task instructional strategies should be dynamically selected for each particular learner. However, specifying effective micro-strategies is both challenging and a less mature and understood area. We are not aware of any major player focusing on developing effective strategies and then offering them, for a price, similar to how publishers and teachers design instructional tasks and assessment items. This may also indicate the sensitive nature of sharing strategies. The instructional strategies are at the very core of various educational services such as CognitiveTutor, AutoTutor, DeepTutor, or ALEKS, and often are “the secret sauce” of those products, which makes instructional strategies less transparent and less separable, if not impossible to share. All these issues suggest that selecting a learning resource for within task tailored instruction could be an adequate level of standardization at this moment as opposed to standardizing the selection and specification of instructional micro-strategies. CognitiveTutor, AutoTutor, and DeepTutor share some common micro-strategy elements such as the use of hints and assertions. When instructional tasks are authored their standardization may require the specification of hints and assertions which the AISs can use as they wish.

Setting the next instructional goal. This implies the involvement of several key components such as the learner model, the pedagogical module that handles instructional strategies (account for prerequisite structures, spacing effects, type of task, modality, etc.), the domain module, and curricula standards (policymaker recommendations such as state K-12 education standards). For instance, the next instructional goal could simply be pulled from a state curriculum standard or inferred through a complex mechanism that accounts for many variables. It should be noted that by instructional goals we mean coarse grain goals such as linear equations or projectile motions or common base amplifiers as opposed to micro-goals that could be used within a task, e.g., a micro-goal could be working on the next step of the solution to a problem solving task. Should there be a separate component dedicated to this function?

Selecting the next instructional task. While addressed this functionality earlier, we reiterate it again to make a case for considering it for a component of its own. This corresponds to VanLehn’s outer loop functionality. Given the next instructional goal, this component is supposed to select from a pool of instructional tasks (see next more on this topic) the next best task. The instructional task selection could be a complex process involving various aspects such as spacing effects where a particular task targeting an already mastered KCs could be selected in order to optimize long-term learning. Should there be a separate component dedicated to this function?

Instructional Task Database. Selecting the next task from a pool of tasks implies the availability of such task pool or database. The tasks can be developed by content developers such as publishers who typically publish textbooks or various instructors who may create their own tasks. When adding tasks to the pool of tasks, their representation must clearly specify what instructional goal they may serve such that an AIS can query the database tasks that can serve the current instructional goal for the current learner. Since content developers may choose to focus exclusively on content development and just provide such content to commercial or research AISs, we recommend having such a component with the corresponding functionality of storing and retrieving tasks that can serve various instructional goals.

The tasks could vary from regular textbook sections to be read to short video lectures to simulations to problems to be solved. It should be noted that in some cases the tasks in the database need to be developed in collaboration with AIS developers. For instance, a problem to be solved with ALEKS will have to have associated with it just a solution, i.e., a worked out example, whereas the same problem to be solved interactively via a dialogue based intelligent tutoring system will need a tutorial dialogue script. In fact, one can imagine a query operation to the task database requesting a task related to a particular instructional goal for a particular type of AIS, e.g., dialogue-based interactive problem solving. When there is such a task but no interactive version, i.e., no tutoring script available, a not-available status code should be the response.

Assessment. The assessment module should return at least a holistic score for a particular subject, e.g., Algebra, and also for the set of KCs that the student has mastered and attempted. How comprehensive and fine-grained the set of KCs is could vary as discussed earlier in the learner model section.

Assessment can be macro-level assessment, which we just described, or micro-level assessment, which is step-level assessment needed to inform within task feedback. Micro-level assessment may not need to constantly update the learner model after each student response within a task. The micro-level assessment component (or the "conductor" component) may send an update request to the learner model only when a student finishes working a particular task.

Assessment Item and Instrument Database. Similar to the instructional task pool, there should be an assessment instrument and item pool. The specification of the instruments and items must be standardized. Such assessment items must be aligned with the various KCs and learning goals such as they could be used when needed for the desired purpose, e.g. assessing learners' knowledge related to a particular learning goal.

Generally speaking, AISs rely on various artifacts for their operations such as instructional tasks, assessment items and instruments, instructional strategies, and cognitive/skill/domain models. The specification of these artifacts should be standardized in order to be easily shareable by various players.

Operations Standardization, including Error Flags. For each of the modules identified for a common architecture, a set of operations must be defined, as we already pointed out earlier. For instance, the assessment module should accept "assess" requests for assessing a learners' knowledge state. The input and output needed for these operations should be specified. Such commands sometimes cannot be successfully executed in which case an error code needs to be sent back to the inquiring module. A standardization of such error codes is thus needed.

Extensions of AISs. External connections interface standards should be at logical level, e.g., KCs, as opposed to lower level as indicated earlier. This is similar to the logical level in the SCSI interface for PCs where details about the electrical and mechanical level are not addressed in the logical interface.

4 Conclusion and Next Steps

We presented a novel view of the architecture of AISs based on a functional analysis, previous research efforts, as well as some industry analysis. Furthermore, we favor an emphasis on the logical level standardization first.

We advanced the idea of an interface design that avoids ontological commitments with respect to learner model specifications, thus decoupling the interface specification from the specifications of those ontologies. We strongly believe that by avoiding any ontological commitments it is possible to define a flexible and stable standard, i.e. one that does not need to be changed or updated often, while at the same time encouraging innovation and refinement.

Some parts of the AISs are harder to specify, less understood, and sensitive such as the specification and sharing of instructional strategies. In those cases, we suggested to use a monolithic solution that treats the strategies as embedded in a learning resource.

We have not addressed other aspects of learning such as team learning and related topics such as instructional tasks for team. The ideas presented could ideally be extended to team learning.

5 Acknowledgements.

This work was supported by the Army Research Lab (W911NF-12-2-0030) and the Office of Naval Research (N00014-12-C-0643; N00014-16-C-3027). Any opinions, findings, or recommendations are solely the authors'.

6 REFERENCES

1. Doignon, J.-P. & Falmagne, J.-C. (1999). *Knowledge Spaces.* Springer-Verlag.
2. Koedinger, K. R., Corbett, A. C., & Perfetti, C. (2012). The Knowledge-Learning-Instruction (KLI) framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36 (5), 757-798.
3. 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: Volume 1-Learner Modeling*. Orlando, FL: U.S. Army Research Laboratory. ISBN 978-0-9893923-0-3. Available at https://gifttutoring.org/documents/42. (pp. 217-226).
4. Rus, V., D'Mello, S., Hu, X., & Graesser, A.C. (2013). Recent Advances in Conversational Intelligent Tutoring Systems, *AI Magazine*, 34(3):42-54.
5. Sottilare, R.A., Brawner, K.W., Goldberg, B.S. & Holden, H.K. (2012). *The generalized intelligent framework for tutoring (GIFT).* Concept paper released as part of GIFT software documentation. Orlando, FL: U.S. Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED). Retrieved from: https://gifttutoring.org/attachments/152/GIFTDescription_0.pdf
6. Sottilare, R.A. & Brawner, K.W. (2018). Exploring Standardization Opportunities by Examining Interaction between Common Adaptive Instructional System Components, Adaptive Instructional Systems (AIS) Standards Workshop, March 7-8, Orlando.

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

Knowledge Components as a Unifying Standard of Intelligent Tutoring Systems

Andrew C. Tackett[1], Zhiqiang Cai[1], Andrew J. Hampton[1], Art Graesser[1], Xiangen Hu[1,2], Ruben Ramirez-Padron[3], Jeremiah Folsom-Kovarik[3], Cameron Copland[3]

[1] University of Memphis, Memphis TN 38152, USA
[2] Central China Normal University, Wuhan, Hubei, P.R.China 430079
[3] SOARTech, Orlando FL 32826, USA

Abstract. ElectronixTutor constitutes the culmination of an effort to bring together many disparate Intelligent Tutoring Systems in a unified platform. This required several key developments. Learner mastery and understanding of the various concepts addressed by each learning resource, topic, and item needed to be measured and quantified via some forms of common ontological unit. Difficulty levels of individual learning items, topics, and learning resources needed to be constructed in a transferrable manner. Finally, curriculum cohesion and completeness needed to be assessed. To address all of these problems, we employed knowledge components as a central unit within ElectronixTutor. We demonstrate how the creation and implementation of knowledge components solved these problems and we propose that their adoption as a standard in Intelligent Tutoring Systems would greatly benefit the learning community as a whole.

Keywords: Intelligent tutoring systems, ElectronixTutor, Knowledge Components

1 Introduction and Motivation

Intelligent Tutoring Systems (ITSs) are learning environments that help learner learn through adaptive interactions. In each step in a learning process, an ITS assesses a learner’s knowledge states and other psychological characteristics from the learner’s inputs and present the learner with the most suitable proximal content interaction.

Knowledge components are used as the basic units of learning state, affording intelligent assessment and recommendation by the system. ElectronixTutor is a complex system that integrated together multiple intelligent learning resources, including AutoTutor, Dragoon, LearnForm, ASSISTments BEETLE-II and NEETS [1]. One of the major challenges in integrating such learning resources was passing a learner’s learning state among the learning resources. Knowledge components [2] have played the role of representing learners’ learning states in ElectronixTutor. Due to the lack of standardization, we had to come up with our own way to define knowledge components for electronics. However, with the help of the knowledge components, ElectronixTutor could adaptively provide content and learning resource to learners and accurately track learner’s progress. In this paper we will first present an overview of ElectronixTutor, then we will describe how knowledge components were used in ElectronixTutor for fine-grained tutoring and high level recommendation. At the end we will discuss some ideas about standardization of knowledge components in general domains.

1.1 Overview of ElectronixTutor

ElectronixTutor [1] is a meta-Intelligent Tutoring System whose purpose is to help teach learners about the use and function of electronics and electronic circuits. It combines several different existing ITSs as Learning Resources. These include AutoTutor from the University of Memphis [3], Dragoon from Arizona State University [4], ASSISTments from Worcester Polytechnic Institute [5], BEETLE-II from the Office of Naval Research [6], Learnform from Raytheon [7], as well as the Navy Electricity and Electronics Training Series (NEETS) from the US Navy [8]. All of these resources are integrated together within the open source learning management system of Moodle. To track learner progress and performance, ElectronixTutor populates and references a Learning Record Store (LRS) running the xAPI standard. Specifically the LRS implementation is Learning Locker, a performance-oriented LRS which makes use of MongoDB's powerful aggregation pipeline.

Learners can interact with ElectronixTutor in three ways (see Fig. 1):

(7) Topic of the Day—learners are assigned a series of topics to learn from an instructor beforehand and then proceed through these topics in a bundle loop which assesses their understanding of each topic by presenting Learning Resources in an adaptive manner; depending on performance, more complex and advanced resources or less complex and simpler resources are loaded for learners to work on.

(8) Recommendations—learners receive an average of three recommended topic-by-learning resource pairs from which to choose. ElectronixTutor produces these recommendations to promote cognitive variability and maximal learning via an algorithm described later.

(9) Self-regulated learning—learners navigate to various items directly via the menu on the left hand side of the screen. This is useful for advanced learners or those who do well with self-directed learning.

Both Topic of the Day and Recommendations utilize knowledge components in formulating suggested next items for learners to work on, as described later in this paper.

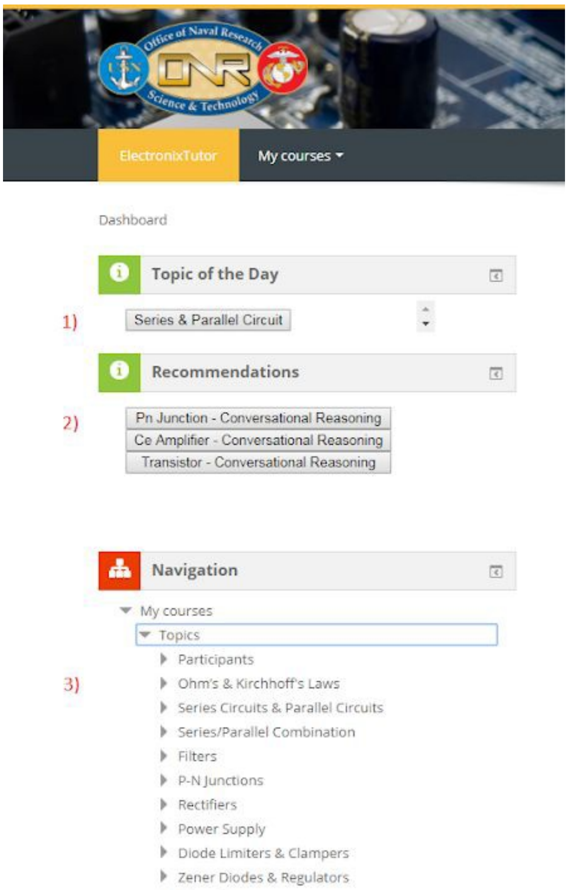


Fig. 1. The ElectronixTutor content selection interface, indicating the three methods of interacting with learning content in red numerals (enumeration not visible to learners).

2 Knowledge Components in ElectronixTutor

A Knowledge component is defined by Koedinger et al [2] as “an acquired unit of cognitive function or structure that can be inferred from performance on a set of related tasks”. They are usually pieces of knowledge and skills a learner has acquired through instruction. Whether or not a learner possesses a knowledge component can be assessed by performance on items associated with that knowledge component. However, there is no standard way to specify knowledge components.

Each knowledge component in ElectronixTutor is specified as a topic-frame pair, where each topic is a concrete component in the curriculum (e.g., transistors, filters, PN junction), and each frame is the epistemic frame or schema specified by a domain expert (e.g., structure, function, parameter, behavior). Within the epistemic frames, structure represents mastery of the components and terminals that a device has. Parameter represents the influence of quantities and values of variables on the device operations. Function is an understanding of the purpose of device components on a successful on successful device activities. Behavior represents the specific impact on how particular device states influence specific device activities.

For example, we can examine the knowledge component Diode_Behavior_Forward. Its topic is Diode and its epistemic frame is Behavior. It represents the following ideas:

*The diode only lets current flow in one direction through it, namely the direction that its triangle points. When current is flowing through it, the diode is said to be forward biased. When the diode is forward biased, then the diode has a very low resistance, about 100 ohms. Most resistors have a much higher resistance, so when a forward-biased diode and a resistor are in series the total resistance is pretty much the same as the resistor's resistance. Thus, the amount of current that flows is determined by Ohm's law (V =I*R) where R is the resistance of the resistor. The bigger the resistor, the less current that flows.*

Other knowledge components represent ideas in a similar manner.

2.1 Knowledge Component Uses in ElectronixTutor

In ElectronixTutor, knowledge components are used for multiple purposes, including (1) checking completeness of curriculum; (2) linking various learning resources; (3) identifying difficulty level of learning items; and (4) tracking learners' learning progress.

Curriculum Completeness. In ElectronixTutor, there are hundreds of learning items. Learning items are mapped to curricula through knowledge components. The way we specify knowledge components provides a structural organization of learning curriculum. The mapping from learning items to knowledge components can clearly show the completeness of the learning curriculum and the repetition of knowledge components in items (see Table 1).

Table 1. Knowledge components to item/topic/learning resource mapping snippet. (1 indicates the knowledge component is associated with the item)

Learning_Resource_Name	Topic	TYPE	Diode_ Physics	Diode_Behavior_ Avalanche	Diode_Behavior_ Forward
PN_DR_ Q1	PN Junction	AutoTutorDRQ	1	0	0
PN_DR_ Q2	PN Junction	AutoTutorDRQ	0	1	1
PN_KC_E_Q1	PN Junction	AutoTutorKC	0	1	0
PN_KC_E_Q2	PN Junction	AutoTutorKC	0	0	1
PN_KC_Q1	PN Junction	AutoTutorKC	0	0	1

Linking various learning resources. ElectronixTutor incorporates multiple independently-developed learning resources, with the capacity to incorporate new learning resources as they become available. This integration requires that a learning resource satisfies two requirements. The first one is simply a mapping from an item to knowledge components. This is usually easy for a content expert to do even after the development of the learning resource. The second requirement is that when a learner finishes an item, the learning resource needs to be able to report a performance score on each associated knowledge component.

This often involves the creation of wrapping code because the independently developed learning resources often do not have the standardized assessment based on knowledge components. In ElectronixTutor, the performance score was always a value from 0 to 1, a common metric for all learning resources.

Item difficulty level identification. Every item within ElectronixTutor has a calculated theoretical difficulty value. This starts by assigning difficulty values to each of the four categories of knowledge components. Structure, function, behavior, and parameter knowledge components are assigned respectively a 0.25, 0.5, 0.75, and 1.0 difficulty levels. Learning resources are also assigned difficulty values of 0.25, 0.5, 0.75, and 1.0 for BEETLE & ASSISTments, AutoTutor Knowledge Check questions, AutoTutor Deep Reasoning questions & Learnform, and Dragoon respectively. Each of the topics has an assigned difficulty of 1-6 which is scaled to be between 0-1. Finally, the item's difficulty is calculated as an average of its learning resource difficulty, the number of knowledge components associated with the item, and the difficulties of each knowledge component associated with the item.

These theoretical difficulty values are used in lieu of empirical difficulty values which are planned to be incorporated into ElectronixTutor after user data has been collected. They will be assessed and calculated using machine learning techniques.

Learner modelling with knowledge components. In ElectronixTutor, learners are modelled using several key metrics including: knowledge component scores, item scores, learning resource scores, topic scores, item completion, and item success among others. Each of these metrics makes use of knowledge components in various ways.

Knowledge component scores are the mean of all individual knowledge component scores for a knowledge component. Item scores are the proportion of knowledge components with scores greater than the knowledge component threshold for that item. Learning resource scores are knowledge component scores for the learning resource associated with an item. Topic scores are the mean of the knowledge component scores associated with each topic. Item completion is tracked via checking whether there is an associated knowledge component score for each item attempted. Item success is where item scores are higher than an item threshold.

2.2 Knowledge Components in Topic of the Day and Recommended Items

Knowledge components are used extensively in the Topic of the Day and recommended items paths in ElectronixTutor. Below we briefly describe the algorithms for item selection in both.

In Topic of the Day, learners proceed through a bundle structure through a variety of steps. Learners first receive a topic summary containing an overview of the current topic. They then attempt an AutoTutor deep reasoning conversation, a natural language dialog to probe learner knowledge, to assess understanding of the current topic on several dimensions using knowledge component scores. Afterward, high performing learners proceed to the most difficult learning resource, a Dragoon item. Low performance on the deep reasoning conversation triggers a move to one of the more limited and paced learning resources, including topic summary, BEETLE-II, or ASSISTments. Intermediate performance on deep reasoning conversations directs learners to LearnForm items. After learners achieve a high enough performance on a variety of learning resources, as measured with knowledge component scores, they proceed to the next topic. If learners are unable to achieve the requisite level of mastery, they loop back through the bundle starting with the Topic of the Day again.

Recommended items are generated based on learner knowledge component scores as well as several rules. First topics are repeated if a learner’s topic performance score falls below a threshold. Next there is a focus on underperforming knowledge components. Topics with medium performance scores and individual knowledge component scores below a threshold are recommended. In addition, there is pushing the envelope where learners who perform above a threshold but have not seen certain learning resources have those resources recommended. Finally, we have motivated and unmotivated bottom dwellers as defined by topic performance scores and their processing time in completing items.

2.3 Knowledge Component Implementation Details

In ElectronixTutor, instances of individual knowledge component scores are implemented and saved into the learning record store as xAPI compliant statements. Note in the Figure 3, BEETLE and BEETLEQ1. These specify the learning resource associated with the item just completed as well as the problem id of that individual item. Each knowledge component score is a separate statement, allowing for aggregation at the statement level using MongoDB’s aggregation pipeline to produce the average knowledge component score. Note also resistor_series_behavior under statement → result → response. This is the title of the actual knowledge component whose score we are recording. Under actor → mbox we have the email address of the learner associated with the knowledge component score. Finally, under result → score→ *scaled* we have the actual value that is used for calculations involving knowledge component scores.

```
   "statement": {
     "authority": {
       "mbox": "mailto:hello@learninglocker.net",
       "name": "New Client",
       "objectType": "Agent"
     },
     "stored": "2018-03-29T20:47:35.677Z",
     "context": {
       "extensions": {
         "http://autotutor.x-in-y.com/AT": {

         "SiteHome": "ElectronixTutor",
         "MID": "x",
         "timeNow": "2018-03-29T20:47:36.725Z",
         "correct": "true",
         "user": "andrew.tackett@electronixtutor.com",
         "UserStudent": "Andrew",
         "timeStart": "2018-03-29T20:47:34.209Z",
         "fullname": "Andrew Tackett",
         "timeTaken": 2516
       }
     }
   },

   "actor": {
     "objectType": "Agent",
     "name": "Andrew Tackett",
     "mbox": "mailto:andrew.tackett@electronixtutor.com"
   },
   "timestamp": "2018-03-29T20:47:35.677Z",
   "version": "1.0.0",
   "id": "859677d1-895e-40df-918a-100405fce68b",
   "result": {
     "score": {
       "scaled": 0.5,
       "min": 0,
       "max": 1,
       "raw": 0.5
     },
     "response": "resistor_series_behavior"
   },
   "verb": {
     "id":
"https://umiis.github.io/ITSProfile/verbs/SaveKCScore",
     "display": {
       "en-US": "SaveKCScore"
     }
   },
   "object": {
     "id": "xxx",
     "objectType": "StatementRef"
   }
 }
```

Fig. 2. A statement storing a knowledge component score. This shows our xAPI compliant method of storing knowledge component scores along with relevant data including problem id, learning resource name, knowledge component name, and learner email.

3 Standardization of Knowledge Components in ITSs

We have presented the important role knowledge components play in ElectronixTutor. However, due to the lack of standardization, we had to be creative in devising a system to construct and organize knowledge components and to integrate independently developed resources from many different institutions. In general, knowledge components could be obtained by decomposing subject matter topics and learning goals into small sets of concepts and skills [2]. A central question is how we can set a standard for ITS authors to follow in creating such concept and skill sets?

We propose the following simple rules for ITS knowledge component specification.

- A knowledge component has a unique name for identification;
- A knowledge component contains a unique set of concepts and skills;
- There exist ways to assess learners mastery of the component;
- The union of the knowledge components should complete cover the target domain; and
- ITSs provide multiple opportunities for a learner to experience each knowledge component.

4 Conclusion

We have shown that knowledge components were integral in implementing ElectronixTutor and useful in several different ways for unifying the various intelligent tutoring systems that it integrates together. As basic units of learning states, they are instrumental in knowledge assessment and recommendation. Additionally, their common specification provides a useful way to determine difficulty levels of learning items, topics, and learning resources as well as link various ITSs. Finally, when mapped to curriculum items, topics, and learning resources, the knowledge components can be used to assess curriculum completeness. We encourage others to adopt knowledge components as part of their ITSs in the future to enable greater standardization and interoperability.

5 References

8. Graesser, A. C., Hu, X., Nye, B. D., VanLehn, K., Kumar, R., Heffernan, C., Heffernan, N., Woolf, B., Olney, A.M., Rus, V., Pavlik, P., Cai, Z., Wetzel, J., Morgan, B., Hampton, A.J., Lippert, A.M., Wang, L., Chen, Q., Vinson, IV, J.E., Kelly, C.N., McGlown, C., Majmudar, C.A., Morshed, B., Baer, W., & Andrasik, F. ElectronixTutor: an intelligent tutoring system with multiple learning resources for electronics. In: International Journal of STEM Education: Innovations and Research. (in press).
9. Koedinger, K.R., Corbett, A.C., & Perfetti, C. The Knowledge-Learning-Instruction (KLI) framework: Bridging the science-practice chasm to enhance robust student learning. In: Cognitive Science, vol. 36 (5), pp. 757-798. (2012)
10. Graesser, A.C. Conversations with AutoTutor help students learn. International Journal of Artificial Intelligence in Education, vol. 26, pp. 124-132. (2016).

11. VanLehn, K., Wetzel, J., Grover, S., & van de Sande, B. Learning how to construct models of dynamic systems: the effectiveness of the Dragoon intelligent tutoring system. In: IEEE Transactions on Learning Technologies. (2016).
12. Heffernan, N., & Heffernan, C. The ASSISTments ecosystem: building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. In: Int J Artif Intell Educ, vol. 24, pp. 470–497. (2014).
13. Dzikovska, M., Steinhauser, N., Farrow, E. et al. In: Int J Artif Intell Educ vo. 24, pp. 284. (2014).
14. Kumar, Rohit & Chung, Gregory & Madni, Ayesha & Roberts, Bruce. First Evaluation of the Physics Instantiation of a Problem-Solving-Based Online Learning Platform. pp. 686-689. 10.1007/978-3-319-19773-9_92. (2015).
15. Navy, U. S. Navy electricity and electronics training series (Vol. 1–24). Naval Education and Training Professional Development and Technology Center, Pensacola (1998).

Proposed Standard for an AIS LOM model using Pedagogical Identifiers

Jeanine A. DeFalco [1]

[1] Oak Ridge Assoc. Universities/Army Research Laboratory, West Point, NY & Orlando, FL

jeanine.a.defalco.ctr@mail.mil

Abstract. The history of instructional design and technology has historically included not only the analysis of learning and performance problems, but more centrally the design, implementation, and management of instructional processes and resources to improve learning and performance [1]. Within the burgeoning field of adaptive instructional systems (AISs), exploring opportunities to standardize processes within the field of AISs should include clarifying and revising interfaces that have already proved effective [2]. Metadata tagging and metadata models for learning objects are established, effective, and central elements in the authoring process of instructional design for AISs, and one that merits establishing a standardized protocol for AISs. This paper will advocate for adopting an AIS Learning Object Metadata (LOM) model standard to include pedagogical identifiers based on the learning framework of Bloom's Revised Taxonomy [3]. Further, this paper will argue that this revised taxonomy – from an historical perspective and current domain practices perspective – is the most comprehensive framework of learning to employ in this standardization effort. Lastly, this paper will propose a framework of pedagogical identifiers that instructional designers and curriculum developers can use in future implementation of this standard.

Keywords: Standards, Adaptive Instructions Systems, Metadata tagging, Pedagogical identifiers, Bloom's Revised Taxonomy

1 Introduction

The history of instructional design and technology has historically included not only the analysis of learning and performance problems, but more centrally the design, implementation, and management of instructional processes and resources to improve learning and performance [1]. However, given the accessibility and flexibility of some adaptive tutoring systems (AIS) that allows for the role of the instructional designer to shift from deciding merely how to present materials to determining what materials should be used for instructional objectives, the process to tag and retrieve learning objects that identify objects associated with complex cognitive processes should be standardized.

Within the burgeoning field of AISs, exploring opportunities to standardize processes within this field should begin with clarifying and revising interfaces that have already proved effective [2]. Metadata tagging and metadata models for learning objects are established, effective, and central elements in the authoring process of instructional design for AISs, and one that merits establishing a standardized protocol. This paper argues that adopting a standard to include pedagogical identifiers based on Bloom’s Revised Taxonomy [3] will help fuel compatibility and interoperability for instructional designers and AIS developers, making it easier to search, retrieve, and compare learning objects more effectively in the effort to maximize learning and performance across a range of domains. Following this discussion, there will be a more detailed analysis of scope of the learning framework that constitutes Bloom’s Revised Taxonomy as well as a suggestion on how this framework can be implemented.

1.1 The problem with standards

It is widely acknowledged that the establishment of specification and procedure standards in a domain is important to ensure reliability of development and use of materials, products, methods, and/or services [2]. Standards fuel compatibility and interoperability, making it easier to understand and compare products [2]. However, not all efforts in establishing standards are successful. The failure to establish standards is often the result of “proactive” attempts to design new products or methodologies before the new efforts are given an opportunity to be used and tested [2]. With that in mind, the advocacy for establishing a new standard should be one that is not only grounded in a historical context of products and/or processes that have been used and existed with in a domain for some time, but Sowa [2] argues that the adoption of a new standard should be more of an iteration, clarification, and revision of a product or process already established as effective and useful. Accordingly, the proposed standard for AISs consists of revising the IEEE Learning Object Metadata (LOM) model to incorporate the use of pedagogical identifiers based on Bloom’s Revised Taxonomy [3].

1.2 Critiquing IEEE Learning Object Metadata model

Metadata tags provide information not only to authors of AISs, but are key in driving learning object searches. Metadata tagging represents key micro-communications between systems and search engines, and without it, the ability to meaningfully consume and exchange information with learning management systems is limited.

The IEEE Learning Object Metadata (LOM) is a metadata model consisting of conceptual data schema designed in a way to allow for extensions to the established schema, such as new vocabularies or taxonomies [4]. These taxonomies act as qualifiers for nine established categories: general, lifecycle, metametadata, technical, educational, rights, relation, annotation, and classification [4]. The IEEE LOM standard specifies the semantics and syntax via attribute elements so learning objects can be used, re-used, or referenced in a learning platform.

Within the educational category, there are eleven elements with accompanying descriptors: interactivity type; learning resource type; interactivity level; semantic density; intended end user role; context; typical age range; difficulty; typical learning time; description; and language [4]. It is this first element, the interactivity type,that is flawed in its current iteration, and most specifically warrants a reexamination and revision for this metadata model standard as well as standardizing metadata tagging for AIS's.

Currently, the IEEE LOM description for the interactivity type element within the Educational Category is allows for descriptors that include (1) active learning, (2) expositive learning, (3) mixed learning, which is a blend of active and expositive interactivity types [4]. From an educational psychology perspective, this simplistic model of learning not only excludes decades of work in curriculum and instructional design theory and research, but it also does not allow for the tagging of elements that covers more complex cognitive processes nor does it adequately support tagging learning objectives beyond the cognitive domain, namely, the affective and sensorimotor domains. As such, the IEEE LOM data model should be revised to change this limited definition of learning types. In its place, the revision should replace the definition of the interactivity element with pedagogical identifiers based on the established vocabulary and definitions established in Bloom's Revised Taxonomy [3], and this revision would serve as the basis for a new AIS LOM metadata model and metadata tagging process standard.

What follows is a review of the history of instructional design and theory that will contextualize the justification for the adoption of Bloom's Revised Taxonomy [3] as the definitive learning framework by which to guide the proposed AIS LOM standard.

2 Instructional design and Curriculum development

2.1 Historical context

Historically, instructional design and curriculum development inhabited two distinct domains and approaches to learning [1, 5]. Curriculum developers focused primarily on what should be learned in schools whereas instructional designers focused on how content should be organized for industry and military [5]. Curriculum developers can follow a theoretical path back to Ralph Tyler (1949) who consolidated a framework of curriculum design for educators to address what should be taught, how learning experiences should be selected, how learning experiences should be organized, and how these experiences could be assessed [5].

In the 1950s, Tyler's approach was simplified and adapted as instructional design was established by media specialist, educational psychologist, industrial and military instructors. In the 1970s, instructional designers expanded the landscape of instructional systems to include additional considerations such as resources, constraints, and alternate delivery systems [5]. What followed over subsequent years were similar yet distinct theoretical learning models both within the domains of the instructional designer and the curriculum developer both of whom sought to provide a procedural framework for designing successful learning environments. The salient distinction between these two groups, however, rested in differing views on what constituted "learning" and how that learning was to be achieved.

Instructional designers of the 1950s were heavily influenced by Skinner's 1954 behavioralist model [6], where content was broken down into behavioral objectives and steps were devised to achieve those objectives. These behavioral objectives were further bolstered within the field when Bloom's original taxonomy of learning was published in 1956 [7]. This behavioralist approach was further solidified by Gagné's work in 1965 [8], and ultimately became the cornerstone for instructional practice through the 1980s [1].

Curriculum developers were also influenced by Skinner and behavioralism. However, where the aim of instructional designers was focused on industrial implications of designs that employed engineering process models [9] (e.g., Gagné), curriculum developers focused on learning aims in schools [5]. These learning aims and accompanying curriculum designs went through a shift in the 1980s when theories of constructivism became mainstream, and when the basic orientations of curriculum were distinguished [5].

2.2 Constructivism

Constructivism is the theoretical position that knowledge and learning arises from a process of active construction [10]. Constructivism has its roots in the philosophy of John Dewey (1933) [11], and follows an intellectual history from Bruner (1960) [12], to Piaget (1972) [13], to Vygotsky (1978) [14]. Constructivism includes the ideas that learning occurs when students are placed in problem solving situations where they have to draw on past experiences and prior knowledge to discover new facts, relationships and information [12]. Fundamentally, constructivism challenged the transmission approach to teaching by demonstrating that long term learning and transfer does not occur when students passively engage with content, but rather learn through guided discovery, simulation-based learning, and problem based learning, amongst others. Further, constructivism prompted a reevaluation as to the fundamental orientations and purposes of teaching and learning [15].

Importantly, the ideas of constructivism influenced not only curriculum designers and instructional design principles [1], but the principles of constructivism influenced subsequent researchers and educational theorists. It is out of this landscape that Bloom's Revised Taxonomy (2001) [16] emerged – where the *what* of teaching was included in a framework of *how,* thus providing a theoretical meeting place upon which this paper argues to devise a standardized approach to employing pedagogical identifiers. For the purposes of the field AISs, arguably the role of authoring content for tutoring systems requires not just an understanding of *how* to design instructional content, but needs must also answer *what* should be taught and *why*. As such, the blurring of lines between instructional designers and curriculum developers is in part one of the rationales for adopting a learning framework that comprehensively addresses the how and what that needs consideration in developing a learning environment.

3 Bloom’s Revised Taxonomy examined

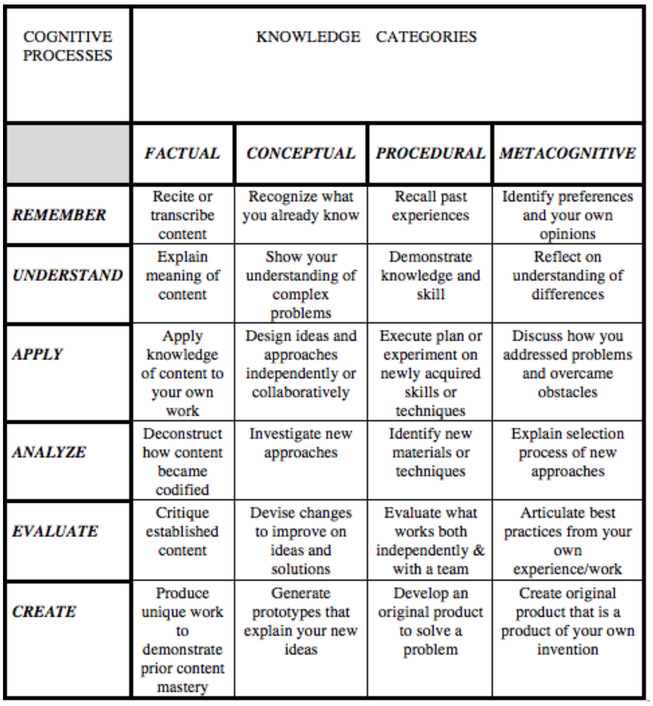
3.1 Evolution of the taxonomy

The 1956 publication of the *Taxonomy of Educational Objectives: The Classification of Educational Goals* yielded the classification system of cognitive skills, commonly referred to as Bloom’s taxonomy (after one of the editors of the text) [7]. This text has had a profound and lasting effect on all levels and across virtually all domains of education [17]. This original learning framework was a one-dimensional framework consisting of a continuum of organizing objectives to help educators clarify and communicate what they wanted students to learn: *Knowledge, Comprehension, Application, Analysis, Synthesis, Evaluation* [16]. Over the years, though, changes in the aims of instruction as well as new understandings about how people think and learn, necessitated a revision of this taxonomy.

Bloom’s Revised Taxonomy [16] was the result of a group of instructional researchers, cognitive psychologists, and curriculum theorists’ efforts to emphasize the interrelationship of cognitive processes and knowledge, changing Bloom’s Taxonomy from a one-dimensional framework to a two-dimensional one (see Figure

1) [16]. The cognitive process dimension consists of six categories that lie on a hierarchical continuum, and slightly revised from the original taxonomy. In order of complexity, the categories are as follows: *Remember, Understand, Apply, Analyze, Evaluate, Create* [16]. In this revision, the taxonomy uses verbs instead of nouns, and swaps the highest of the thinking skills “evaluation” with “creation” [16]. The second dimension, the knowledge dimension contains four categories of increasing complexity: *Factual, Conceptual, Procedural,* and *Metacognitive* [16]. Similar to the cognitive processes, knowledge categories lie on a hierarchical continuum, beginning with concrete (*Factual*) and spanning to abstract (*Metacognitive*), with *Conceptual and Procedural* in-between and overlapping into the concrete and abstract categories [16].

While the taxonomy does not proscribe exactly how fill in the categories between cognitive processes and knowledge categories, Anderson et al. [16] does provide examples in their definitive volume of how to use the taxonomy table to determine how to devise objectives, how to use the table to make decisions on their curriculum/instructional design, and how to determine the alignment of objectives, assessments, and activities. Figure 1 (see below) demonstrates a sample taxonomy table with fixed labels for cognitive processes and knowledge categories, and examples of the kinds of learning objectives that could be filled in across and between these processes and categories. These boxes could also be elaborated upon, by including specific learning objects that address the two-dimensional nature of cognitive processes and knowledge categories – objects that could include activities and assessments that align with the identified learning objectives.



COGNITIVE PROCESSES	KNOWLEDGE CATEGORIES			
	FACTUAL	***CONCEPTUAL***	***PROCEDURAL***	***METACOGNITIVE***
REMEMBER	Recite or transcribe content	Recognize what you already know	Recall past experiences	Identify preferences and your own opinions
UNDERSTAND	Explain meaning of content	Show your understanding of complex problems	Demonstrate knowledge and skill	Reflect on understanding of differences
APPLY	Apply knowledge of content to your own work	Design ideas and approaches independently or collaboratively	Execute plan or experiment on newly acquired skills or techniques	Discuss how you addressed problems and overcame obstacles
ANALYZE	Deconstruct how content became codified	Investigate new approaches	Identify new materials or techniques	Explain selection process of new approaches
EVALUATE	Critique established content	Devise changes to improve on ideas and solutions	Evaluate what works both independently & with a team	Articulate best practices from your own experience/work
CREATE	Produce unique work to demonstrate prior content mastery	Generate prototypes that explain your new ideas	Develop an original product to solve a problem	Create original product that is a product of your own invention

Fig. 3. An example of a complete taxonomy table with learning objectives.

3.2 Comparison of Bloom's Revised Taxonomy to other learning frameworks

At this point, there is an anticipated criticism that perhaps there are other established taxonomies, or frameworks of learning, that would be better suited or easier to use than Bloom's Revised Taxonomy. Anderson et al. [16] provides an analysis with comparisons on eleven alternative unidimensional frameworks and eight alternative multidimensional frameworks that identify how Bloom's Revised Taxonomy either parallels or supersedes the following frameworks in terms of comprehensiveness, similarities and differences. These frameworks include the following:

Unidimensional:

1. Gerlach and Sullivan (1967) Taxonomy of Commonly Taught Behaviors
2. Ausubel and Robinson (1969) Six Hierarchically Ordered
3. Metfessel, Michael, and Kirsner's Synonyms (1969)
4. Gagne (1972, 1977) & Gagne and Briggs (1979) Hierarchy of Learning
5. Stahl & Murphy (1981) Domain of Learning
6. Bruce's (1981) Integration of Knowledge with the Other Categories
7. Romizowki's (1981) Analysis of Knowledge and Skills
8. Biggs and Collis (1982) Structure of the Observed Learning Outcome (SOLO)

9. Quellmalz (1987) Taxonomy of Cognitive Processes
10. Hauenstein's (1998) Conceptual Framework for Educational Objectives
11. Riegeluth and Moore (1999) Comparison Framework

Multidimensional:

1. DeBlock's (1972) Three-Dimensional Framework
2. DeCorte (1967) Modification of Guilford's Structure of Intellect Model
3. Ormell (1974-1975) Modification of the Taxonomy
4. Hannah and Michaelis (1977) Comprehensive Framework for Instructional Objectives
5. Williams (1977) Behavioral Typology of Educational Objectives
6. Marzan (1992) Dimensions of Learning
7. Merrill (1994) Component Display Theory
8. Haladyna (1997) and Williams and Haladyna's (1982) Typology for Higher-Level Test Items

The only other taxonomy that could be identified for comparison that was not included in Anderson et al.'s [16] review was Finks Taxonomy (2003) [18]. Finks Taxonomy is distinguished from Bloom's Revised Taxonomy in that Finks is not a hierarchical taxonomy, but rather covers a cross section of domains and excludes the psychomotor domain [18].

Overall, the comparison of the eleven unidimensional frameworks and nine (including Finks) multidimensional frameworks can be summarized as efforts to improve on the original Bloom's Taxonomy or create an easier one to use. Within that context, it is not beyond the scope of reason to choose Bloom's Revised Taxonomy as the framework of choice for the AIS standard as Bloom's Revised Taxonomy offers a greater breadth of possible pedagogical identifiers with an established set of definitions and terms to assist in that alignment.

3.3 Criticism of validity of hierarchical categorization of Bloom's Revised Taxonomy

Perhaps one of the most significant criticisms of Bloom's Taxonomy is the assumption of employing a hierarchical framework. At the time of the publication of the original Taxonomy, 1956, there was no empirical evidence to support assumption of employing a hierarchy framework, and as such, that criticism was largely warranted [16]. Since that time, though, there has emerged some empirical evidence as to the validity of a hierarchical framework that derived from research on the original Taxonomy. This empirical research to validate the use of a cumulative hierarchy was established through a range of investigative methods on the original taxonomy, including path analysis, factor analysis, and structural linear equation modeling [16].

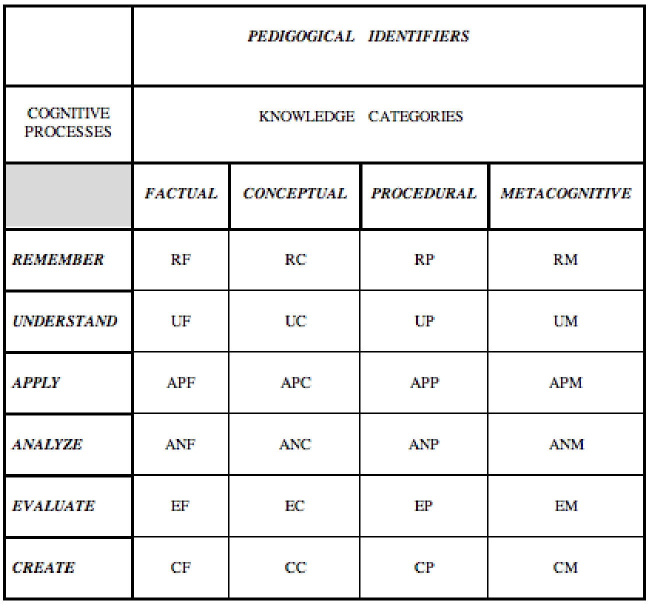
Further, a meta-analysis conducted by Kreitzer and Madaus (1994) [19] concluded there was supporting evidence for the ordering of less complex categories of Comprehension, Application, and Analysis, but less for placement of most complex ones of Synthesis and Evaluation, and Knowledge placement in the structure was identified as problematic. While this is not an exhaustive review of the empirical validation of Bloom's Revised Taxonomy, a more thorough analysis is beyond the scope of this paper. Suffice to say that in combination with the aforementioned empirical evidence, the validity of Bloom's Revised Taxonomy is also grounded in the scope and longevity of active use of this framework across a diverse range of domains that spans decades.

4 Using Bloom's Revised Taxonomy for pedagogical identifiers

Turning from establishing a justification for the use of Bloom's Revised Taxonomy as the most comprehensive of learning frameworks, we can now address how this taxonomy can be employed for the purposes of establishing an AIS standard by revising the IEEE Learning Object Metadata (LOM) and advocating for the use of pedagogical identifiers based on Bloom's Revised Taxonomy.

The example taxonomy chart provided in Figure 1 is an example of a filled in taxonomy chart, where the cognitive processes and knowledge categories are fixed, but the learning objectives are determined by the designer/instructor. However, the taxonomy table exists as a mostly blank template but for the cognitive process identifiers and the knowledge categories. Within this template, there is room for instructors and developers to fill in the template according to their specific instructional needs. For example, the template could be used to fill in learning objectives, as is demonstrated in Figure 1, but it can also be used to fill in learning objects. To guide designers and developers in filling in their template, Anderson et al.'s [16] volume includes definitions of knowledge types and subtypes, process categories, and specific cognitive processes with verbal descriptions regarding sample objectives, sample activities, and sample assessment tasks, providing a tool kit and structured guide to align their objectives, activities, and tasks.

As such, the importance of the template and tool kit includes a standardized vocabulary and taxonomy of definitions for use in establishing a standard for a metadata model that incorporates pedagogical identifiers in AISs. Further, this established vocabulary and taxonomy of definitions have already been recognized and practiced within the field of education for a sustained period of time, thus meeting one of the benchmarks of establishing a successful standard as identified by Sowa [2]. Therefore, to translate Bloom's Revised Taxonomy into a set of useful pedagogical tagging identifiers, one could adopt the following methodology as demonstrated in figure 2:



	PEDIGOGICAL IDENTIFIERS			
COGNITIVE PROCESSES	KNOWLEDGE CATEGORIES			
	FACTUAL	***CONCEPTUAL***	***PROCEDURAL***	***METACOGNITIVE***
REMEMBER	RF	RC	RP	RM
UNDERSTAND	UF	UC	UP	UM
APPLY	APF	APC	APP	APM
ANALYZE	ANF	ANC	ANP	ANM
EVALUATE	EF	EC	EP	EM
CREATE	CF	CC	CP	CM

Fig. 2. Proposed pedagogical identifier tags

In sum, by adopting a standard that employs Bloom's Revised Taxonomy, a simple set of metadata tags can be standardized and subsequently used that directly correlates previously established product and processes of categorizing learning objects, easing the process of categorizing, classifying, searching, retrieving, and re-using content for instructional designers and curriculum developers to use when designing courses in AISs.

5 Conclusion

As the blurring of lines between instructional designer and curriculum developers continue to evolve, the tasks in authoring learning environments in AISs requires not only understanding of what to teach, but how to teach. Content alone does not produce positive learning outcomes; rather it is interaction of knowledge and cognitive processes, as well as aligning learning activities and assessments with learning objectives that creates a robust learning environment. To aid in the authoring of AIS courses, instructional designers and curriculum developers need to search, identify, and retrieve learning objects that are relevant to their learning objectives. By revising the IEEE LOM metadata model to change the interactivity definition to one that uses the definitions provided in the two-dimensional Bloom's Revised Taxonomy framework, and by standardizing the process to include using metadata tagging with the incorporation of pedagogical identifiers described above, an industry standard can be achieved that is both effective and useful for AIS developers and designers.

6 Acknowledgements

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number **W911NF-17-2-0152**. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

7 References

1. Reiser, R. A history of instructional design and technology: Part II: A history of instructional design. *Educational technology research and development*, 49(2), 57-67 (2001).
2. Sowa, J. The Law of Standards, http://www.jfsowa.com/computer/standard.htm accessed on 2018/4/3 (2004).
3. Anderson, L., Krathwohl, D. A Taxonomy for Learning, Teaching, and Assessing. Addison Wesley Longman, Inc., New York (2001).
4. IMS Global Learning Consortium. IMS Metadata Best Practice Guide for IEEE 1484.12. 1-2002 Standard for Learning Object Metadata. http://www.imsglobal.org/metadata/mdv1p3/imsmd_bestv1p3.html accessed 2018/03/27 (2002).
5. Petrina, S. Curriculum and instruction for technology teachers. *http://www.cust.educ.ubc.ca/programs/tsed/research/books* accessed 2018/04/04 (2004).
6. Skinner, B.F. Teaching machines. *Science*, 128, 969-977 (1958).
7. Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. Taxonomy of educational objectives: The classification of educational objectives. *Handbook I: The Cognitive Domain, David McKay*, (1956).
8. Gagné, R.M. (1965b). *The conditions of learning*. Holt, Rinehart, and Winston, New York (1965).
9. Carr-Chellman, A. A., Rowland, G. (Eds.). Issues in technology, learning, and instructional design: Classic and contemporary dialogues. Taylor & Francis, New York (2016).
10. Mascolo, M. F., Fischer, K. W. Constructivist theories. *Cambridge encyclopedia of child development*, 49-63 (2005).
11. Dewey, J. Experience and education (1933).
12. Bruner, J. *The Process of Education*. Harvard University Press, Cambridge, MA (1960)
13. Piaget, J. *The psychology of the child*. New York: Basic Books (1972).
14. Vygotsky, L. *Thought and language*. MIT Press, Boston, MA (1986).
15. Loyens, S. M., Gijbels, D. Understanding the effects of constructivist learning environments: Introducing a multi-directional approach. *Instructional science*, *36*(5-6), 351-357, (2008).

Anderson, L. Krathwohl, D., Airiasian, P., Cruikshank, K., Mayer, R., Raths, J., Wittrock, M. P. *A taxonomy for learning, teaching and assessing: A revision of Bloom's Taxonomy of educational outcomes*: *Complete edition*. Longman, New York (2001).

16. Adams, N. Bloom’s taxonomy of cognitive learning objectives. *Journal of the Medical Library Association: JMLA*, *103*(3), 152 (2015).
17. Fink, L. Creating significant learning experiences: An integrated approach to designing college courses. John Wiley & Sons, San Francisco, CA (2013).

18. Kreitzer, A., Madaus, G. Empirical investigations of the hierarchical structure of the taxonomy. Bloom’s taxonomy: A forty-year retrospective. Ninety-third yearbook for the National Society for the Study of Education: Part II, 64-81 (1994).