Delivery of simulation based training instruction and assessment through Distributed Learning Management System – Instructional design considerations

Dr. Jacqueline Haynes
Mr. Ohad Bukai
Dr. Robert Pokorny
Dr. Kevin C. Ruess

15400 Calhoun Dr. Suite 400
Rockville, MD 20855, USA
jhaynes@i-a-i.com

Abstract

In this paper we present the view of Intelligent Automation Inc. (IAI) as to the future development of Instructional Systems Design (ISD) when applied to Simulation Based Instruction (SBI). We call for the formalization of the SBI ISD process in a way that will achieve the separation of the ISD process from a specific simulation implementation, and address the complexity and special needs of the process. Additionally we describe a new model for instructional design that can be applied to SBI. Unlike traditional models such as ADDIE that focus on the process, our model focuses on the integration of four design elements: process, ontology, collaboration model and product description. We show how the proposed model integrates with current trends in Simulation-SCORM interoperability, such as incorporating simulation, SCORM and the S1000D technical documentation standard, as well as streamlining SBI ISD with the ISD of traditional Web-based training (WBT) used in a similar context. The proposed model streamlines both SBI ISD and the IDS of traditional web based training. Our suggested approach contains a supporting business model, as well as a new class of automated ISD editors to help instructional designers plan effective simulation instruction interactions.

1. Problem description

While the Sharable Content Object Reference Model (SCORM) was originally designed as a standard to deliver training that is static in nature to a single user on demand, it is increasingly pressured to provide for richer user experiences to both individuals and teams with training solutions that are more dynamic and software intensive in nature. As part of this trend the need arises to support the delivery and management of simulation within the SCORM infrastructure. There are two obstacles to supporting that objective: (1) There is a need to standardize flexible support for a wide array of simulation technical solutions, and (2) There is a need to better address the user experience from an instructional point of view.

In this article we address the issue of Instructional System Design for simulation. Instructional System Design (ISD) was introduced as a formal model of making decisions that support effective instruction [12]. A frequently used methodology for performing ISD is the *ADDIE* model (Analysis --> Design --> Development --> Implementation --> Evaluation) [13]. With the growing technological abilities of simulation technology, and growing complexity of the systems they model, there is an emerging need to enhance instruction using traditional delivery media with simulation. It is our opinion that current models of instructional design are inappropriate for the development of simulation-based instruction and therefore there is a need for a new approach that will capture four aspects relating to simulation usage: technological, organizational / infrastructure, business setting, and instruction delivery platform. We will refer to this new methodology as SISD (Simulation Instructional System Design). Three independent aspects prevent the current ADDIE model from being well suited or practiced in simulation-based instruction:

(1) Organizational considerations – Organizational considerations lead to situations in which instructional design is introduced late in a simulation development project when simulation decisions important for instruction have already been made. In other cases legacy simulation systems are transformed to be used for instruction, limiting the flexibility of any instructional design process.

(2) The process itself cannot address all simulation instruction needs – Current instructional design process models, such as the ADDIE model, cannot be directly applied to simulation as many of their stages are already bound by the decision to use simulation. In the same way, the separation between analysis design and implementation stages may not be as clear cut as in traditional instruction. Simulation Based Instruction (SBI) also has special requirements due to simulation being a complex system, using nonlinear advancement, and being a nondeterministic experience, thus making current ISD process models inapplicable to simulation.

(3) Linkage between the designs to specific implementations – Frequently instructional design is linked to a specific implementation in a way that diminishes the independence of the instructional design process. In these situations, the ISD process is tailored to support special characteristics of a specific simulation and may not be applicable to others.

The combination of the above three difficulties suggests that there is no common forum or organizational / business infrastructure for the development of a unified SISD doctrine, or to map the various practices of SISD available. However, at the infrastructure interoperability level, solutions are currently being developed to deliver simulation-based instruction over the web through learning management systems (LMS). The lack of a standard for SISD will become more apparent, and more problematic, as infrastructure interoperability evolves without SISD support.

We expect that as simulation based instruction will be used more through LMS, its ISD outcome and pedagogic reasoning will be explicit within the ISD package in a way that will enable instructional designers to position simulation in the context of other instruction in a pedagogically sound manner. Moreover, SBI will have some basic constructs that are shared and understood across the industry.

Our discussion here focuses on the online delivery of instruction through simulation. An important standard for online delivery of instruction and a focus of this paper is SCORM [2]. Nevertheless, many of the concepts discussed are relevant beyond consideration of SBI, and apply to standards other than SCORM. It has been demonstrated before that simulation instruction can be delivered within SCORM using a SCORM-compliant LMS [3]. It has also been demonstrated how to pedagogically integrate simulation into a rich instructional environment combining SCORM-based didactic instruction, interactive instruction, and simulation used for practice and performance assessment [4]. These and other demonstrations show the potential value of a combined solution offering Simulation-SCORM interoperability, tied together with a comprehensive pedagogical approach. The IEEE Learning Technology Standards Committee (LTSC) Simulation Interface Standards and Simulation Interoperability Standards Organization (SISO) study group has started working on scoping and addressing the standardization of a technical solution. Still, there has been little community discussion to date addressing standardization and automation in the area of ISD and the pedagogy of SBI.

We begin by asserting that developing a standardized process of instructional design for SBI should be addressed independently from infrastructure interoperability issues. We will describe how instructional design should be embedded with a future infrastructure solution of online delivery and reusability within SBI, and we will explain the need for the industry to formalize the instructional design process and to treat SISD as a special case of ISD. We will then describe the main building blocks that a formal SISD process should contain. We will call for separation of the instructional design process from a specific simulation implementation. Finally, given the above considerations, we will demonstrate how automation can enhance rapid delivery of quality instruction for simulation, address the business potential of the described model, and, lastly, call for the instructional design community to take action to address the issues of instructional design and simulation.

2. Distinction between an instructional view and an infrastructure view

When a standard that deals with instruction is introduced, it has to account for both infrastructure technical considerations, and pedagogic instructional issues to reach its objectives. The infrastructure considerations are technical enablers to perform the target tasks, and the pedagogic ones ensure the quality and the usefulness of the target standard from an instructional point of view, and delivery of instruction in a pedagogically sound manner. In the case of SCORM, for example, Advanced Distributed Learning (ADL) chose to focus on a pedagogy -neutral solution that accounts only for infrastructure requirements. The results are that learning objects do not have embedded pedagogy or ISD metadata that is required for reuse, sequencing, and discovery of learning objects. This lack has resulted in a variety of methodologies to externally 'inject' pedagogy into SCORM [11]. To avoid this problem in future related developments we suggest that the effort to standardize the online delivery of instruction through simulation should include parallel technical and instructional efforts that are independent in nature, but maintain parallel links. Although the two efforts may exist in the same context, they serve different communities, have different target objectives, and are controlled by different professionals. To address the technical view, the IEEE LTSC and SISO, are hosting a study group, open to interested parties, in an effort to initiate a process that eventually will lead to a Simulation-SCORM-interoperability standard. The three primary areas of the group's focus on an infrastructure solution to address SCORM-Simulation interoperability are:

- Addressing an infrastructure framework that will enable a simulation to be used in the context of SCORM.
- Mapping use cases, in which simulation is (or might be) used through the LMS.
- Defining taxonomies in the context of simulation.

These focus areas explicitly address technical / infrastructure needs to streamline an industry-wide standard to extend simulation to be used through the LMS. These three focus areas do not affect the internal pedagogy and instructional design of the underlying simulations.

In parallel to the important infrastructure questions, there should be an independent process to address standardization and best practices in how instructional design processes relate to SBI. The following are some questions to be addressed in this regard:

What are the building blocks of SBI?

What are the specifics of the instructional design process for simulation (SISD)?

How can automation increase the quality and rapid automation of SISD?

How can we streamline the development and delivery process of SBI to be aligned instructionally with other traditional web based instruction?

How can standardized SISD help instructional designers to author better instruction using simulation?

How can a SISD standard help in the discoverability of simulation instruction, and the authoring of courses, leveraging pre – authored simulation instruction objects?

How can instructional products that relate to simulation be reused the same way as other software components?

The outcome of this effort should be a common base for instructional designers to develop, communicate, and exchange products of SBI, across different organizations, and in different stages of the lifecycle of the SBI product.

3. Our view of the infrastructure solution (and delivery in the context of SCORM)

In order to achieve an effective instructional setting for simulation and instruction through SCORM, instructional and infrastructure efforts have to work cooperatively to ensure that control methods for the web based instruction and for the simulation will fulfil both infrastructure requirements set by simulation vendors as well as instructional pedagogical standards set by the clients and content authors. Here are some of the base considerations for fulfilment of setting infrastructure and instructional solutions in the same context.

3.1 Encapsulation in a SCO-like object

It is expected that training interactions will be encapsulated in an autonomous object. This type of object will: (1) have all the technical information to launch the simulation and access and use needed resources; (2) make a communication portal available for the hosting client to send commands and retrieve reports from the underlying simulation; (3) maintain enough metadata for instructional designers to use it in a larger context; and (4) maintain an API to maintain control by the launching LMS.

3.2 Availability of instructional metadata

As of now, SCORM objects do not contain significant instructional metadata to explain the instructional planning involved in the design of the underlying interaction. We expect this issue to be addressed in the future for SCORM, but meanwhile instructional metadata will support a few objectives. (1) It will help instructional designers use simulation objects in a way that will be harmonic with other sequenced instruction. (2) It will enable instructional designers to discover simulation instruction objects using emerging storage and discovery repositories. The discovery will include pedagogical attributes as well as content. (3) If instructional information for an object is stored in a standardized manner, its attributes can be accessed automatically at runtime and manipulate interactions automatically.

3.3 Breakdown of experience into atomic "stages" analogous to SCO

Most current publications regarding the integration of SCORM and simulation create a relationship in which the simulation itself equals one SCO [5]. This approach is technically reasonable as it requires only one load of the simulation. Nevertheless, instructionally it forms problems: (1) The interaction cannot be "broken" into stages (2) SCORM cannot intervene and select a course of action if only one SCO is used. These limitations call for enabling technology and standards that will enable a simulation runtime to stay persistent as new concepts, events and layers are introduced.

3.4 New SCO definition

The definition of a SCO as an atomic unit of instruction may not be appropriate to represent the concept of a single unit of instruction in the context of SBI. The issue with SCO definition in the context of SBI is the inter relationships in the learning experience between content and context. A traditional WBT SCO is aligned within some type of sequencing, but presented independently, and always the same. A simulation atomic object may appear different depending on the context of a specific scene, and may appear together with other interacting objects with multiple permutations. Furthermore, and perhaps most importantly, it generally will not have meaning without an encapsulating context. As such, there is a need for a new definition of atomic instructional objects and their inter-relationships as they relate to SBI.

3.5 Conflicting controller's issue

In a SCORM based environment, a simulation learning experience, has two "competing" controllers: (1) static scripted scene information set by the instructional designer; and (2) dynamic influences from external sources such as live coaches, other participants, other dynamic entities accessible through HLA or other simulation infrastructure. This situation is problematic as it causes ambiguity: With dynamic external agents, a single SCO could deliver different experiences to different users, based on a diversified dynamic environment. To address this problem we propose the use of a "task grid" that will electronically serve as an "arbiter" of responsibilities regarding control and presentation during the simulation. In general, this grid will specify which aspects of the simulation must be presented to deliver the desired instructional experience of the SCO. Other aspects of the simulation are free to vary under the control of programmed or human agents involved in the simulation. This specification is critical so that the SCO presents an equivalent instructional setting for the learner even with uncontrolled agents, and to ensure an appropriate experience for different learners.

3.6 Interaction between external content and simulation state

There can be various models of transition of control between SCORM content and simulation. The difference between the potential models lays in the level to which control is shared, from full control by the LMS to full transfer of control to the simulation. Our preference is shared control. In a shared control setting, the calling instruction (SCORM) maintains control of some areas of the screen / console, and the simulation controls dedicated parts of the screen / console. As such, together with state information that the simulation shares, the host instruction can support the user with coaching / external information that changes in way that is synchronized with the simulation state.

3.7 Repository compatibility

Simulation objects should be accessible through repositories the same way SCORM objects are. After a simulation instruction object has been authored, it can be made widely available, it can exist and be re-used in pieces smaller than a single course, it can be discovered, and it can be retrieved in standard ways.

3.8 Parallel execution of SCOs

Current SCORM infrastructure assumes SCO exclusivity of execution, that is, as one SCO is running, no other SCO can be launched prior to termination of the execution of the other, and no two SCOs can operate, deliver content, and report to the LMS at the same time. This exclusivity is enforced by the various sequencing rules, as well as by LMS implementations. Parallelism can enable teaching of various aspects, or layers of the learning material at the same time, and make possible the modelling of simulation interaction in a way that is exposable to LMS and manageable through it. It will also enable parallel control of training and simulation (for example where training delivers auxiliary information while a simulation is running). As exclusivity of execution will be a limiting factor for delivery of simulation through LMS, we consider two different solutions: (1) We may want to maintain parallel streams of instruction running at the same time: one for the simulation and one for the hosting course, which provides satellite information, coaching and reference to the running simulation. This cannot be effectively achieved without parallelism. (2) Unlike a traditional WBT SCO that has a single focus, a simulation by definition has processes, events and concepts running in parallel. We would like to be able to add and subtract events, layers, difficulties, and attributes, where each of these may have a defined instructional goal but, as a training object, has no instructional meaning without additional context. An additional consideration would be cases in simulation where it would be desirable to introduce various unrelated events in parallel in order to assess or train users for complex, multitasking skills and complex environments. Accordingly, it would be beneficial for an infrastructure solution to address the capability of parallelism.

4. The need for formalizing instructional design for simulation

Instructional design models such as ISD, formalize the process of defining, developing, and analyzing training. If the training adheres to a model, it most likely has gone through user analysis, training analysis, technical analysis, a rigorous design process, development, and a final post-implementation evaluation. The effectiveness of a model is also subject to its use in the context where it was originally designed to be used. Such a formal model streamlines and enhances the rapid development of instruction, and enables participants from different organizations to cooperate. Several initiatives traditionally formalize standards for technical interoperability of online training. Among these organizations are ADL, IEEE, AICC, IMS, and several others. There is no ongoing effort to address the instructional design process for SBI.

4.1 Formal instructional design as a mean of communication across participating organizations

To represent the need to formalize the instructional design progress for SBI, we first describe the lifecycle of a (SCORM-based) simulation instructional package:

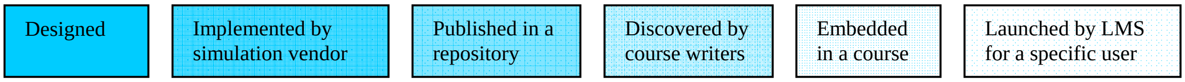


Figure 1: various usages of simulation instruction object

Figure 1 illustrates some of the various contexts in which a simulation instruction package is used. This requires the package to *potentially* be used by six distinct organizations, starting with its design and ending in its delivery to a learner. In any of these six stages, instructional design is either created or used in order to make decisions. In some contexts, the decision is made by software components at runtime rather than manually.

1. Design – Instructional designer employs instructional design methodology to create the interaction.

2. Implementation – Simulation engineering team uses the design as a formal requirement for developing activities for the simulation to deliver.

3. Publishing – When a simulation package is published, attributes of the instructional design and pedagogy are used as metadata attributes required for cataloguing purposes.

4. Discovery – Attributes that relate to the instructional design and underlying pedagogy are used when an instructional designer searches for a simulation package in a repository, to match the specific needs of a learning experience.

5. Embedding in course – When embedding in a course, the simulation must be aligned with the instruction and pedagogy embedded within the course.

6. Launching through the LMS – The end result should run the instruction and simulation in harmony. Using the same formal language, automatic query and reporting can be performed between the LMS and the simulation using formal agreed-upon constructs as objectives and skills rather than implementation specific attributes.

Training assessment and training revision – When a simulation package is assessed, its effectiveness can be assessed in comparison to the instructional design it represents. Parts of it can be edited based on changes in objectives or required skills.

As described above, in the emerging world of distributed learning, a situation in which instructional design practices are local to a specific publisher is unlikely, as a single simulation instruction package goes through many hands and across unconnected organizations. Any user of such simulation objects should have easy access to the instructional design and pedagogy embedded in the SBI package without having to know the specifics of the instructional design practices of the authoring organization.

4.2 Formal instructional design as a mean of increasing design quality

Without an alternative instructional design process that is more appropriate for simulation, instructional designers tend to apply various versions of the ADDIE, or other ISD methodologies, as much as it can apply in their design. It has been suggested before that other models may be more appropriate for simulation. Taylor for example suggested the Dick and Cary Design Model [6]. The sparsity of developers applying focused SISD process makes the unification of the process unlikely. As instruction by simulation will increase and be more accessible over public shared infrastructure, this sparsity will be more visible and will affect the community more as instructional designers will attempt to enhance instruction with simulation, or to embed simulation within instruction. There is a little written about the specifics of an appropriate instructional design process for SBI.

Unfortunately, the ADDIE model is not sufficient for SBI for the following reasons:

1. Much of the *Analysis* stage in the ADDIE model is geared towards the selection of interaction type With SBI, the simulation itself has required interactions, or these interactions are built into the organizational decision making process, making the analysis stage miss its objective.
2. The separation between the *Design Development* and *Implementation* is quite clear with traditional WBT, while in simulation the boundaries between these stages are less clear. One of the reasons may be the fact that with simulation instruction some of the states of design are delegated, or performed in collaboration with engineering groups (who may be designing a functional system upon which the simulation is based) rather than solely within instructional design groups.
3. Simulations are typically designed by a larger number of people who perform different tasks than conventional WBT. The integration of their work becomes embedded as part of the instructional design process used.
4. While for WBT the design process can be seen as a sequence of stages starting from *Analysis*, and ending with *Implementation* and *Evaluation*, for simulation, the design process is phased across many development stages that include multiple level of abstraction, beginning with the abstract and ending with the concrete. An instructional design process for SBI should address this view of simulation design.

The gap between the ADDIE model and the requirements for designing SBI suggests that a new process should be launched to refresh and formalize the instructional design process when the instruction may benefit from using simulation.

4.3 Automation of simulation instructional design process

The process of developing instructional design for simulation is unique in the sense that it requires different skills and types of collaboration to be effective: (1) The number of participants is generally larger than the number of participants in other types of ISD. (2) Tasks that designers do are more inter-related with the tasks of others, such that work packages cannot be encapsulated and sent to developers as in other types of instruction development. (3) There is a large variety of roles among the people who perform SISD tasks. Each role has a different view of the project. (4) The work is performed at different levels of abstraction.

In order to manage and enable collaboration of the participants of this process, there is a need for solutions that will automate the shared effort. General purpose software solutions such as Microsoft Project can assist, but they are too general to give a comprehensive effective tool for instructional designers. In order to enable the development of this class of development and collaboration tools, the process of developing SBI should be formalized in a way that will enable these groups to collaborate in ways that are specific to their tasks, and editors to describe work products in a machine readable / writable way.

4.4 Sharing knowledge about effective learning design

An explicit description of instructional design constructs formally embedded in training packages will prove to be important information. This can be used in conjunction with performance data of learners, collected and analyzed across many applications, to assess the efficacy of specific designs. This will enable the industry to incrementally learn more about the effectiveness of various designs, and about their applicability to various learning contexts. Potentially, this information can aggregate into catalogues of effective learning and teaching patterns that can be communicated precisely and adapted to other contexts, problems or content [10].

4.5 Adaptation and learning support

Machine readable descriptions of the instructional content can be *interpreted* by software elements during runtime and give learners, teachers, and managers a useful tool to help manage and control the flow of the learning process. In the same way, automated software elements can *change the flow and attributes* of the learning experience to adapt to the specific abilities, performance and learning style of each learner. These attributes are at a higher conceptual level than the underlying simulation attributes, and therefore can be used more directly to derive information from an instructional simulation and control it.

4.6 Reuse and upgrade

One of the main objectives in distributed e-learning is reuse of instructional objects. Through reuse, organizations can reduce the cost of developing instruction, and clients achieve a higher return on investment. A key requirement of reuse is that an instructional object will maintain the instructional strategy and design that was initially planned by its author. The instruction-explicit design metadata supplies future users with information as to how specific content components are related to the design, to the strategy, and to the learning objectives. Potential future developers can use this information to evaluate whether an existing instructional object under consideration is suitable for a specific purpose. This formal metadata also enables rapid forward evolution of instruction. By having annotated instruction packages, a new one can be incrementally built, relying on the content of a previous version, without requiring the full instructional design cycle from scratch.

5. Overview of the building blocks of instructional design for simulation.

Traditional models of instructional design focus on describing the process as composed of a series of stages. The growing complexity of target domains, together with the richness and complexity of a simulation as a delivery platform (a simulation system) requires additional features to describe how the instructional design is done. We suggest the following four components: Ontology, Process, Collaboration model, Product representation.

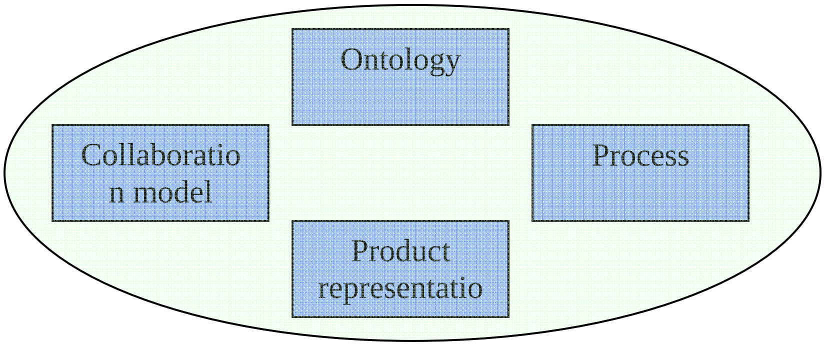


Figure 2: Four components of instructional design for simulation

These four components (which we will describe in more detail) follow the realization that the complexity of the domain, the simulation and the instruction call for intensive use of automation in the instructional design process, and that the automation should not be limited to editing, but should also encapsulate instructional / pedagogy ontology. The automation takes the form of dedicated knowledge management tools and authoring tools that build on a known ontology and manage collaboration of various types of professionals through a process, in which simulation-based instructional packages are created.

5.1 Ontology

This component of SISD deals with the basic objects SISD refers to and their relationships. It is important to distinguish here between the ontology or taxonomy of the underlying simulation and the ontology used to describe the SISD process. It has been recognized that interoperability of simulation systems requires composability of conceptual models [14]. While simulation ontology deals with events, processes, entities software and models related to the modelling that is relevant to the ***simulation***, SISD ontology deals with the instructional process, knowledge objects, objectives, performance, and student model related to the ***instruction*** and the relationships between them. Ontology has been recognized time and time again as an important ingredient of any ISD authoring tool [8]. As authoring tools are not 'theory aware', they have no capacity to declaratively represent instructional design theory, nor help their users implement any theory while authoring. Without an agreed upon ontology, instructional designers do not have the tools to communicate the design or theory they want to implement to other users, or to a potential authoring tool. As authoring tools do not have ontologies built into them, they do not support reuse of content that was developed for another purpose or system. Without a shared ontology, instructional designers find it difficult to communicate among themselves in a manner supporting effective collaboration. The definition of a shared ontology within the simulation-instruction community will facilitate collaboration in a way that will ensure that a common meaning will be given to objects and attributes, and that the same types of objects and attributes will be used in the design and development of instruction.

5.2 Process

The second component of SISD is a definition of the instructional design process. Unlike the ADDIE model in which each stage is defined by activities, here we expect stages to be defined by the output that they produce. Some of the ADDIE-like activities may shift or be shared across stages. The stages of the process themselves are geared to describing the instruction at increasing levels of detail, relying on abstract guidelines derived from the preceding stage in the process.

1. Task analysis – This step allocates all objectives of the instruction. It is performed through a cognitive task analysis or other task analysis methods. This step should describe entry and exit behaviours and skills, knowledge items, required tasks, objectives, etc. Using the result of the task analysis, a model of the expert and models of students at different stages of the learning process will be created. The models will represent abstractions of the performance and knowledge states with respect to target attributes at any stage of the learning. The representation of the models can take various forms such as a state machine, mathematic equation, case selection, decision tree, biometric attributes, some AI components, or other known methods used to describe the state of a model.

2. Drills selection – In this step, the specific tasks and objectives are broken down into drills. Each drill will have entry and exit requirements, and attributes such as time and resources and performance requirements that will be allocated to it. The specific tasks, knowledge items and requirements will be mapped to specific drills.

3. Setting design – This step deals with the specific details of the simulation setting. It contains scene requirements for the whole set of drills, as well as technical requirements for the

underlying simulation. In this step the designer will describe how specific requirements of each drill will be translated into information in the scene.

4. Scene population - Once the requirements for the setting are available, instructional designers can populate the scene with specific information. We see this process mostly as editing a rich script with various layers, although, for specific simulations (for example, ones that are not interactive in runtime), there may be other ways to populate scenes with details. Again, it is important to distinguish between the instructional view of scene population, and the technical (simulation) view of scene population. For our purpose, we mean a description of object attributes and environment values that have instructional meaning. These values are independent and may be different from underlying objects that are presented in the simulation. For example, an instructional designer may break a scene down to three time blocks, each unique in its objectives or level of difficulty. These time blocks do not have any presentable meaning in the native simulation objects that the underlying simulation manages, except for the pedagogical meaning the instructional designer assigns them.
5. Assessment – In this step the instructional designers define the means for the simulation to perform assessment. This is done by applying products from previous steps: (1) The student model from the task analysis is the base for defining performance. (2) Inputs of state of objects from the design and scene population stages are linked to the model, in a way that describes a learner's performance state in any point of time. (3) Entry and exit behaviours described in stages 1 and 2 are defined as pass / fail standards related to the state of the student model.
6. Evaluation – As in the ADDIE model, the result of using the instruction is evaluated. We expect the effectiveness of the instruction to be evaluated at two levels (1) Reviewing the simulation apart from the context of other instruction (2) Reviewing a simulation that is embedded within a course, as a whole instructional experience.

5.3 Collaboration model

The collaboration model describes how an instructional design team works together to develop and deliver SBI. A collaboration model is a component that is not part of traditional instructional design as the variety of skills needed for standard WBT is more standard. Furthermore, the inter-relationships of the collaborators performing WBT ISD are more hierarchical, while in SBI they are more networked. A collaboration model describes the types of collaborators, their skills, and their inputs and outputs related to other collaborators. Having a grid of collaborators, skills, inputs and outputs, work packages can be arranged in order. Of course, there is a direct link between the collaboration model, process, and ontology. The inputs and outputs the collaborators exchange are described using the ontology, while the ordering of tasks follows the above described process. The relationship between these bases of SISD calls for automation of the process using a collaboration tool that manages the users, process, and product. General purpose knowledge management (KM) tools help collaborators communicate, share tasks, collaborate on a shared work space and perform quality control. In the context of instructional design, KM tools that follow a defined collaboration model can bring together instructional designers, software engineers, policy makers, and subject matter experts to address the task of transferring the model to instruction in a precise, effective manner. A collaboration model can address a fine granularity of learning objects and objectives, it can facilitate the appropriate communication to address specific tasks, and it can create constructive collaboration environments to support effective and efficient development.

5.4 Product representation

The outcome of the SISD process is a simulation instruction object. As described in Section 4, this instruction object is not only a design document, but also a formal technical description that is used in many contexts. Some previous work has been done in the field to formally describe performance requirements for simulation alone [7], however we expect efforts in a similar direction to address the full scope of SISD products. Some of the common languages used for modelling are UML, RDF schema, OWL (Web Ontology Language), and XML. The use of a standard machine readable format enables automation in which collaborators can use an online editor to work together in the development of SBI. The output is readable for simulation implementers, instructional designers that embed the simulation objects in a course, and LMSs and SCOs that use attributes of the instruction product automatically. It enables some discoverability capabilities, and enables researchers and organizations acquiring instruction to evaluate the effectiveness of instructional strategies.

6. Separation between simulation instructional design and the underlying simulation implementation.

In this section we examine the relationships between instructional design and a specific simulation implementation. In the classic view of the ISD process, instruction is designed prior to the selection or design of the target platform. It is expected that the development of simulation based instruction will be performed before an actual implemented simulation solution will be selected, or even developed. In this way, the instruction product becomes a digital requirement document for simulation vendors to implement using existing, or newly developed simulations. In practice, however, instruction is often designed late in the stage of developing the target simulation, or even when the simulation is completely built. Even in the best case, instruction is developed in parallel to the simulation and not before. Two voids in SISD contribute to this situation:

(1) Lack of general constructs and formal industry process in SISD – Although the simulation experience is well defined from the engineering / modelling / software point of view, no constructs, entities, stages, etc, are defined in a general manner from the instructional point of view. This situation forces the instructional designers to describe the instructional interaction in terms of the model / software of the underlying simulation implementation, rather than separate from the simulation. Take the model away, and the designers are left with little or nothing to describe the interaction. Describing the interaction in term of the simulation model is a process that results in the loss of instructional planning: The instructional designers “author” their design to the simulation using editors. These editors are pedagogically neutral, and as such, they may not guarantee that a pedagogically sound instructional design is maintained while transformed from hard copies to a simulation ISD documents. They also do not maintain any instructional metadata content that explains the interaction in terms of the instructional design (in terms of objectives, level of performance, prerequisites, etc).

(2) Lack of a formal way to communicate about instructional design between instructional designers and simulation implementers - In order to develop simulations, instructional designers have to use implementation specific proprietary tools to edit and author the simulation interactions. These tools use constructs that are local in nature and that are specific to the model of the underlying implementation.

Both the above causes demonstrate that there is no general purpose model process or standard to describe SBI at different levels of abstraction. It is our view that the SBI industry experience is mature enough to close this gap, which is an outcome of the distributed nature of the industry. This gap shows the importance of the concept we refer to as "*instruction separation*." Instruction separation is a concept stating that SISD should be conducted separately from any specific simulation implementation, and that the linkage between the instruction and a target simulation should be done (at least potentially) at a later stage of the design. We expect the *instruction separation* to take SBI to another level in the following ways:

(1) As the instruction is independent from any specific implementation of simulation, it can be reused as a new simulation implementation or technology is introduced. It can also be used with different simulations having different level of fidelity and different costs.

(2) *Instruction separation* can help to promote outsourcing of the instructional development separately from the producer/vendor of the simulation. It will support inter-organization cooperation enabling organizations with instructional expertise to develop instruction in collaboration with organizations having expertise in simulation.

(3) *Instruction separation* promotes the development of industry standards and shared constructs for SISD. It enables the emergence of general purpose practices in designing SBI with varying level of abstraction, independently from a target simulation implementation or model.

(4) The availability of such general purpose standards enables the development and use of editors to assist designers in developing instructional design for simulation.

(5) The outcome of this design can be formalized in a machine readable way to promote automatic or semi-automatic implementation of designs by simulation vendors.

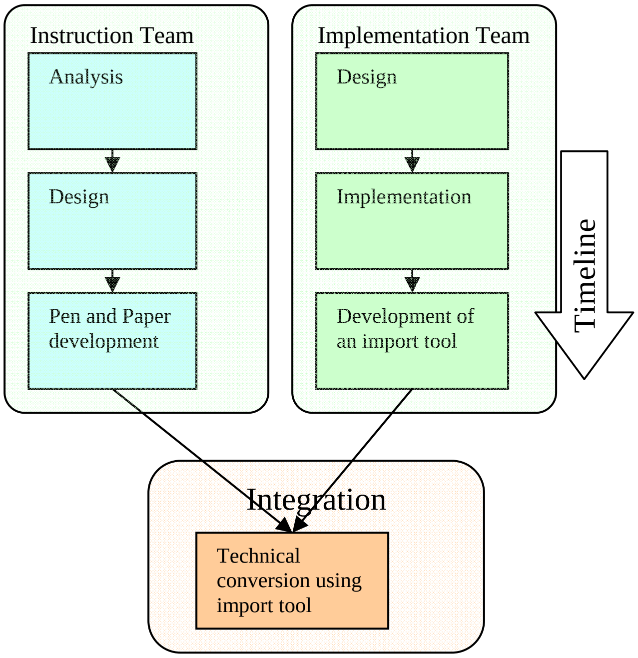


Figure 3 Traditional SBI design process

The two processes are independent and not time constrained

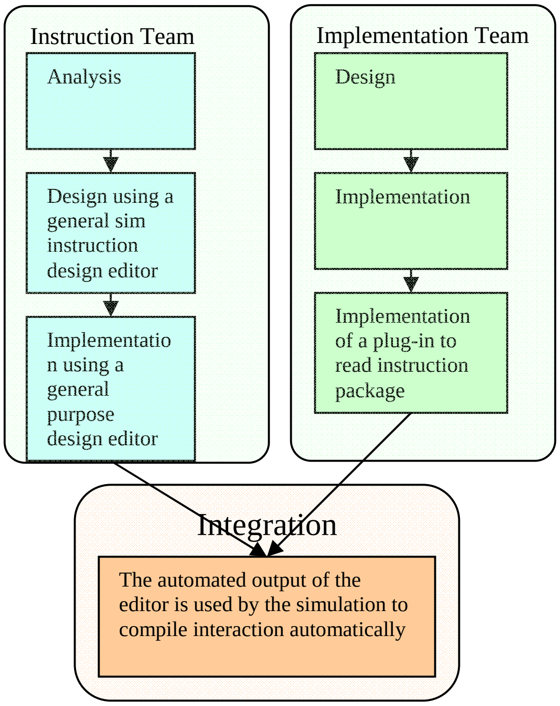


Figure 4 Instruction separation design

process

In Figures 3 and 4 we compare the process of developing instruction for simulation with and without instruction separation.

(1) Developing instruction without instruction separation requires time synchronization between the engineering process and the instructional design process. In an optimized setting they would work in parallel and enhance the design as they go along, linking the development of instruction to the development of the simulation model. As the instruction team is relying on the model of the underlying simulation, much of the instructional design is done without actual tools to conceptualize the design at a higher level, so only work intensive methods are available. As the process progresses from design to implementation, there is a higher reliance on the specific implementation model. Although design tools and methodologies can be borrowed from the engineering world, (such as the use of UML and automated UML tools) these tools are geared to support engineering and systems needs and not instructional needs which are different and independent. At a late stage of development, the instructional team receives an import tool from the engineering team, which enables the instructional team to hardcode their design into the simulation. These tools are effective to the extent of executing the design, but they maintain only a runnable version of the design without the instruction associated with it. Accordingly, the instruction embedded in the simulation cannot be extracted out of it back to an abstract representation for reuse or revaluation. The instructional design product cannot be reused in this case, and may be lost in the transition to the simulation.

(2) Using the instruction separation methodology, the engineering and instructional development can be independent processes, both in time and in hosting organization. In this scenario, the instructional design team develops instruction using instruction-specific methodologies. As these methodologies are general in nature and can apply to any instruction, automated tools can be used to author this design, whether or not it includes simulation. These tools support different levels of abstraction in the description of the target interactions to be used at different stages of the design. These tools are ontology-based and pedagogically rich. The output of these tools is a package, written according to an industry standard that describes the interaction. The engineering team is involved at the later stages of simulation development. The engineering team can use a “plug-in” technology to support the automatic import of the instruction package, and the automatic or semi-automatic conversion to a run time package.

7. The use of editors to assist instructional designers in developing simulation activities.

As described above, one of the outcomes of instruction separation is the availability of general purpose SISD editors.

The emerging complexity of both simulation models and simulation software that implements these models reflects the instructional design itself, which has to account for more attributes and target systems that are non linear, and sometimes less deterministic than in other forms of instruction. Accordingly, the use of “manual” means to perform SISD becomes more difficult and less effective. An automated editor is very useful in guiding the design team through the process and maintaining required attributes, objectives, knowledge items etc.

The editors have two parallel objectives. From the instructional side they implement a specific SISD model and assist instructional designers who develop SBI to follow the model and to develop the instruction accordingly. From the infrastructure point of view, the editors are a common formal way of communicating between the instructional design team and the engineering team. As such, it ensures that the instructional design is developed in a way that is supported by simulation, and that engineering team implements the design without disturbing the instructional features to be embedded in it.

These editors will leverage the SISD model, as the one described in section 5 that will be developed as a community effort. The editor is a design tool to assist instructional designers to develop SBI. As such, this is an instructional tool at its core and not an engineering tool. Its output is not a simulation but a description of the learning experience to be represented in the design and implementation of a SBI instance. It is different than other editors (e.g. Macromedia) as it caters to a specialized instruction delivery platform (Simulation) that is fundamentally different from other more standard delivery platforms (e.g. Web media). One of the main objectives of the editor is to maintain the pedagogic instructional design information embedded in the target training packages. Current editors that edit regular web based instruction, and specifically SCORM instruction do not maintain pedagogic information in the target training package. This leaves the target training package with no metadata regarding the instructional or pedagogic context in which it should be used, and severely limits the ability of instructional designers to search for this package in repositories or to make a sound decision about whether or how to embed specific content within a course of study. In SBI, the need to maintain the pedagogic content is more important, because compared to other WBT, it is more difficult to extract the pedagogic data by "seeing" the interaction (See Figure 5).

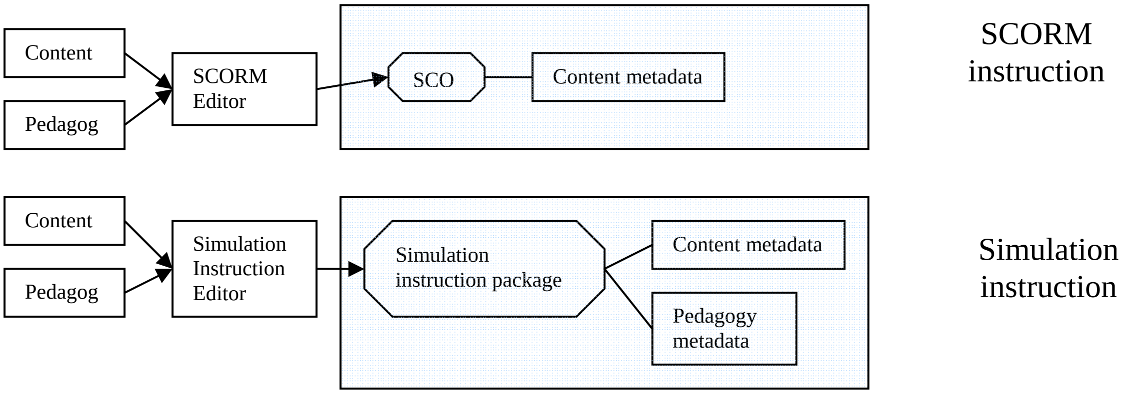


Figure 5: Pedagogic metadata embedded in simulation training

The following are some of the characteristics that this type of editor should support:

(1) Multiple levels of abstraction – As the design is generally done by many people at different levels with different responsibilities, the editor should promote consistency as work is performed, sometimes simultaneously, at different levels. This consistency is difficult to achieve in a static, environment where requirements and specifications exist only as documents. With an editor, however, changes at one level of abstraction can be reflected throughout the design and implementation process. Furthermore, the process can be treated as one with a continuum of activities rather than as a process with discrete steps.

(2) Breakdown of the instruction to discrete “drills” – The editor should enable users to break down the instruction to be delivered into discrete interactions and to map objectives as well as target knowledge and skills to each interaction.

(3) Separation and management of formative “drills”, assessment drills, and remedial drills. – The editor should be able to help the instructor subtype interactions from core interactions to ones that focus on practice, assessment remediation, etc. It should support the tools to assist the instructional designer to describe the interactions as well as how to assemble them when an activity relies on previous performance.

(4) Accountability for product delivery in the context of online instruction and SCORM – As described in section 3 of this paper, we focus here on SBI in the context of SCORM or its successors as a delivery platform. The editor should provide assurance that the SBI package conforms to the delivery standard (SCORM) and that its design will enable SCORM-conforming instruction that also has other types of media which may also have some control over the simulation, and the ability to query for the state of the simulation or some performance measures.

(5) Visual Script utility for design of a time breakdown of drill stages and events – At a later stage of development, a visual editor should enable the instructor to plan the simulation on a semantic timeline. In this editor the instructional designers can edit and link simulation events to instruction requirements and knowledge items and align them over a timeline scale of the simulation run. Alternatively, editors can use a visual state machine as a visual metaphor for the semantic stages of a simulation the learner covers through the interaction. These and other visual editing views of the semantic design of the interaction can serve as a visual metaphor to help instructional designers conceptualize the interaction in their own terms, rather than in technical terms borrowed from an underlying simulation.

(6) Development of performance models that can easily correspond to target implementation models – As the assessment and performance models are independent from the underlying simulation model, they should explicitly be expressed using the editor, and linked to event attributes and values, in a target underlying model.

(7) Development of templates for “standard” reoccurring designs for fast development – The development process using this editor is comprehensive and time consuming. It is expected that for specific classes of simulations, there will emerge some repeating patterns of SBI that will enable rapid development of instruction. The availability of these classes can facilitate developing templates for design instead of developing each instance from scratch.

8. Business view of the PROPOSED MODEL

One of the lessons learned from open standard attempts such as SCORM is that these standards have to reside in the context of a business model that will create incentives for both vendors and consumers to participate within the shared infrastructure. The following are components that support this goal:

Instruction separation – As described above, instruction separation separates the instructional design of simulation from a specific implementation. This separation shuffles the business setting for development of instruction, as it facilitates taking instructional design out of the host organization, the simulation vendor. It also creates a balance in which good instruction does not rely on a single simulation, as it can be implemented using more than one simulation instance. We envision new business relationships and dynamics in the design of instruction: (1) DoD (for example) can develop specifications for instruction, and call for both instructional design vendors and simulation vendors to bid for implementing the two different components. (2) A vendor of instructional content can write simulation instruction and have a target client choose which one of many simulations to use. (3) An engineering company and a content provider can form an alliance for developing simulation and instruction, using the previously described infrastructure as a communication / collaboration medium. Accordingly, a trend in specialization may follow in which simulation vendors will focus on the engineering component, and content and instructional design firms will focus on the instruction.

Reuse – As the cost of developing instruction is significantly lower than the cost of building simulation systems, there is a strong need to reuse instruction for multiple uses and over different simulation platforms. This design is a way to extend the utility of existing or multi-purpose simulations without extending the development of simulation systems specifically for instruction. We see reuse in different context here: The same instruction can be used with different simulation implementations; the instruction can be reused when a simulation is upgraded to a later version, or when instruction or objectives are revised; and existing instruction packages can be changed easily. Instruction can be shared, used, and launched through common sharing and delivery infrastructures as a SCORM LMS and a CORDRA or other repository.

Open standard – The use of an open standard for both the SISD process as well as for the format of the output package introduces more players to compete on providing excellent instruction using current simulations, or excellent simulations within existing instruction. The availability of a standard which describes how to package instruction electronically enables an easy separation between authoring organizations and simulation vendors. The joint development of such a standard across interested parties in the military, academia and private sector will enhance the potential quality, robustness and effectiveness of the target model.

Web based delivery system – The recent advancement in repository standards, and more specifically the SCORM – CORDRA effort enables publishers to expose their instruction products in a searchable and retrievable manner. These repositories are not only a place for clients to find and use simulation training packages; they are also a provide a marketplace for simulation vendors and instruction developers to join. In this marketplace, organizations can “implement” instructional packages for their platform and create instruction or extend instruction for existing implementations.

Putting it all together – As described in Figure 6, the integration of the above features creates a flexible market, in which both suppliers of SBI and suppliers of simulations load their products (instruction package or stub to the simulation) on public repositories. Both simulation stubs and training packages are searchable using the standard CORDRA search and retrieve mechanism. This availability enables both instruction and simulation vendors to initiate a linkage of instruction and simulation. Potential users can access linked / unlinked content for use and acquisition (purchase or license, or whatever business rules are applied). We see this basic infrastructure as an opportunity for the development of other services and products that will enhance the mobility and interoperability of both simulation implementations and training packages.

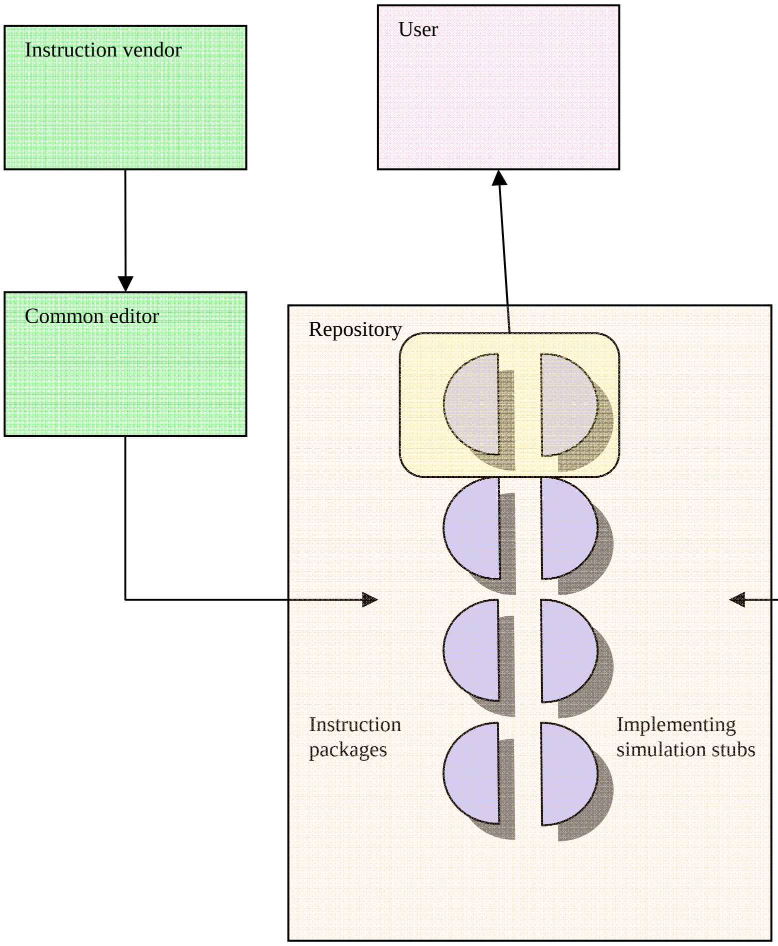


Figure 6: Business view, putting it all together

9. S1000D and its affects on SBI

9.1Applications of S1000D and simulation

S1000D is a standard to manage and describe interactive technical manuals, developed by Technical Publications Specification Maintenance Group (TPSMG). Although originated in Europe, S1000D is gaining popularity throughout DoD, and in expended domains that reach far beyond the original Aviation industry in which it originated. The Advanced Distributed Learning Co-Laboratory has taken the initiative to explore ways to integrate both S1000D and SCORM standards and to allow the same content and processes to be shared among both the SCORM and S1000D community. It is expected that sooner rather than later the need for use of simulation in conjunction with S1000D will become inevitable. We can imagine the following use cases:

- Simulation of collaboration models within a maintenance unit.
- Simulation of equipment failure and the relevant process performed by the maintainer.
- Simulation that focuses on the skills of using S1000D as a basic maintenance skills tutor rather than training for the maintenance of a specific hardware.
- Integration of S1000D simulation within other federated simulations.

- Attempt to describe and maintain simulations using S1000D as the base model.
- A link to a simulator can be embedded inside a technical document replacing a CAD illustration, giving a clearer demonstration of the behaviour of equipment.

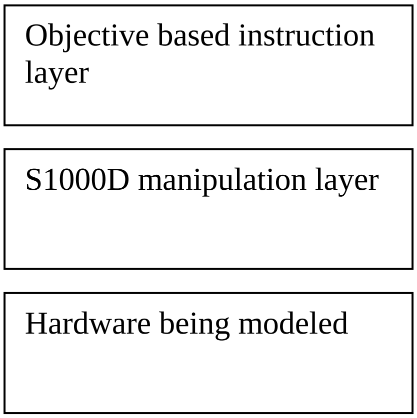
9.2 S1000D and instructional design

The integration of S1000D with instruction raises some concerns about maintaining instructional design standards while complying with the strict S1000D processes and requirements. As instruction has different lifecycle, objectives and target audience than technical documentation, some creative mitigating process must take place to combine the two without losing functionality.

Some examples of issues that we have encountered in the integration process were:

- Items from both standards have different granularity.
- Flow of user experience is strict and follows maintenance practices in S1000D while better flexibility is needed in SCORM to achieve instructional goals.
- When there is an update to a technical publication, what is the right process to ensure update of instruction?

Despite these disadvantages, in the scope of SCORM and Simulation, the introduction of S1000D brings some advantages, very simple user interface that can easily deliver S1000D base simulation in networked environments. S1000D is a great way to simplify the access points to complex systems without compromising the functionality of the maintenance process, enabling some complex simulation to simulate equipment while supporting low signature front end interactivity, which is important in we b based training where computing and bandwidth resources are limited.



In order to overcome the shortcoming above, and the rigidness of S1000D as an interface, an instructional layer should wrap the interface and give some supporting information and scripting. Accordingly, a Simulation based instruction that adheres to both SCORM and S1000D will have 3 theoretical layers: (1) the underlying hardware or equipment being simulated (2) S1000D layer that determines processes and means of interaction/ inspection of the equipment. (3) Instructional layer that positions the S1000D like interfaces into instructional framework.

9.3 S1000D and models for SBI

It has been argued time and time again that the various standards that are used to deliver and present knowledge are in the process of conversion [15]. This refers largely to standards as S1000D and SCORM. Many government officials see redundancy in multiple standards and would like to see content written once rather than twice.

If we return to the 4 components of SBI, as described in part 5 of this paper, it becomes clear that these components do not enforce or require the details of any specific standard, and can easily accommodate the S1000D integration cases described above. This way, using the same ontology, process, and collaboration models, users can address instructional challenges that contain the complexity of S1000D.

10. Activities that have to be done to move this model from theory to practice.

As described above, we see the standards community generating two cooperating parallel efforts: simulation infrastructure, and instructional design. As an infrastructure effort is already taking place, there is an emerging need to spawn the instructional design standardization effort. Although this effort will be open to the public, it is expected that the main participants will be potential clients and authors of content. We expect interest to come from military clients, vendors of instructional design services and content, simulation developers, and organizations such as ADL. The tasks of such a community effort would be:

1. Perform a field study and literature review of current practices.
2. Perform requirement elicitation for the objectives and outcomes of a new SISD process in light of current and future simulation systems capabilities.
3. Perform a needs assessment for the instructional integration of simulation experiences in the context of SCORM delivery.
4. Formalize the instructional design process, ontology, collaboration model and product representation format of SISD.
5. Describe inter relationships between other emerging standards such as SCORM, LOM, technical SCORM-Simulation interoperability standards, repositories and others.
6. Perform pilot tests and demonstrations using the developed standard in a variety of use cases.
7. Encourage private sector development of tools to support automation in implementing the standard.

11. Conclusions

Intelligent Automation, Inc. has been actively involved in various activities related to both simulation and emerging standards that support automation of developing and delivering instruction both in the context of online delivery of instruction, and delivery of innovative instruction through simulation. We have participated in the integration of joint efforts between government organizations such as ADL, and private industry, which have proven to be effective in promoting innovation and testing of new instructional standard initiatives. Such initiatives combine three resources: The clients and their requirements are represented by the government and its branches, vendors of instruction and content are represented and share their view of current practices in development. Research companies and academics provide a rich mix of relevant theory, innovative approaches and development environments to explore and experiment with them. Although currently related standardization efforts are performed by clearly differentiated groups such as SCORM, S1000D, HLA and CORDRA, there is an underlying trend toward integrating these standards under some unified concepts and practices.

References

[1] Chan T, Lin C, Lin S and Kuo H (1993) OCTR: A model of learning stages. In Brna P, Ohlsson S and Pain H (eds) *Proceedings of AI-ED 93* . Scotland, August 1993. ACCE publication.

[2] http://www.adlnet.gov/index.cfm

[3] Manikonda, V., Maloor, P.; Haynes, J.; Marshall, S., "An Architecture for Integrating SCORM-Compliant Instruction with HLA-Compliant Simulation", SISO Fall Simulation Interoperability Workshop, 2004.

[4] J. Haynes, S. Marshall, V. Manikonda, P.Maloor. Enriching ADL: Integrating HLA Simulation and SCORM Instruction using SITA (Simulation-based Intelligent Tutoring System). The Interservice/Industry Training, Simulation and Education Conference (IITSEC), Orlando Florida, Dec 6 -9, 2004

[5]http://www.academiccolab.org/resources/presentations/adlss/Hicks%20Summer%20Institute.ppt#271,14,Slide 14

[6] Lyn Taylor, Monash Medical Centre - Southern Health Nursing Education and Research - Perioperative Services - Educational Theories and Instructional Design Models: Application to Simulation, Health and Medical Simulation Symposium 2004 National Convention Centre, Canberra.

[7] Webb Stacy, Ph.D., Jared Freeman, Ph.D, Emily Wiese, Cullen Jackson, Ph.D, Aptima, Inc. A Language for Rapidly Creating Performance Measures in Simulators, 2005 I/ITSEC CD ROM.

[8] Bourdeau, J., Mizoguchi, R. (2000). Collaborative Ontological Engineering of Instructional Design Knowledge for an ITS Authoring Environment, In Proc.of ITS'00 Conference.

[9] Rachid Benlamri, Jawad Berria, A Framework for Ontology Instructional Design and Planning Journal of e-Learning and Knowledge Society, The Italian e-Learning Association Journal, Vol. 2, No. 1 (2006).

[10] Koper, R. (2004). Use of the Semantic Web to Solve Some Basic Problems in Education: Increase Flexible, Distributed Lifelong Learning, Decrease Teacher's Workload. Journal of Interactive Media in Education, 2004

[11] Tyde Richards, Robby Robson, Charlie Brewer, Virginia Mesenbrink, Sharable Content Object Reuse (SCORE) Methodology Guidelines and Procedures, http://www.jointadlcolab.org/research/2004/score/SCORE_Methodology_Guidance_Procedures.doc

[12] Leshin, C. B., Pollock, J., & Reigeluth, C. M. (1992). Instructional Design Strategies and Tactics. Englewood Cliffs, NJ: Education Technology Publications.

[13] Strickland, A.W. (2006). ADDIE. Idaho State University College of Education Science, Math & Technology Education.

[14] Andreas Tolk (2006) What Comes After the Semantic Web - PADS Implications for the Dynamic Web, Workshop on Parallel and Distributed Simulation archive, Proceedings of the 20th Workshop on Principles of Advanced and Distributed Simulation table of contents, Computer Society Washington, DC, USA

[15] Richard Boyd, Interactive 3D, Converging standards and platforms, Military simulation and training magazine, 6/1005.