Position Paper for ITS'96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs, Montreal, June 10th 1996.

Increasing the Effectiveness of ITS Research and Development

Kurt Rowley, Ph.D.

Armstrong Laboratory
Intelligent Training Branch
7909 Lindbergh Dr.
Brooks AFB, Texas 78235-5352

June 6, 1996

Interoperability and reusability are powerful and enabling capabilities that have helped improve effectiveness of computer-based environments in many domains, and show promise for increasing the effectiveness of ITS research and development. Interoperability in particular, is much more than simply an approach to achieving cost-effectiveness and reusability. As I have discussed elsewhere, interoperability facilitates the creation of large systems by defining the requirements for interaction transactions between many small systems, with the Internet (TCP/IP) and Musical Instrument Digital Interface (MIDI) standards as premier examples (Rowley, 1995). In the case of the Internet, interoperability is evident in the ability of network communication software and hardware from multiple suppliers to follow standardized protocols, and facilitate the reliable transfer of files across their interconnected networks. The interoperability of the components of the Internet has facilitated the proliferation of a new level of interconnectedness in the global economy, through e-mail, FTP and the World-Wide Web. In the case of MIDI, interoperability is evident in the ability of electronic musical equipment from multiple suppliers to interface to common network and data standards, and work harmoniously together to create an integrated, technology-supported music studio. The MIDI music studio supports new types of music composition and management software, allowing a higher level of musician productivity than was possible before the advent of interoperability.

Based on these and other past interoperability experiences, the establishment of interoperability and reusability standards for intelligent tutoring system (ITS) research and development can help bring about an inflection point in the influence of intelligent tutoring systems on instructional outcomes. Interoperability can help each ITS developer concentrate efforts on one part of the ITS, rather than having to constantly `re-invent the wheel' of an entire system for each implementation. In order to illustrate the potential that interoperability and reusability standards can offer ITS researchers and developers, it is useful to consider a vision of possibilities, the levels at which interoperability standards can be applied, possible components of an interoperable ITS, and necessary tasks for the implementation of interoperability and reusability standards among intelligent tutoring systems.

Visions of Interoperable, Reusable Intelligent Tutoring Systems

There are numerous new capabilities that interoperability and reusability can bring to ITS research and development. Some of these capabilities include allowing ITS systems to take advantage of compatible commercial products, improving the cost-effectiveness and reusability of the components of intelligent tutoring systems in diverse domains, and increasing the level of interaction between related intelligent tutoring systems. These capabilities can lead to important benefits such as increased efficiency of the development of tutors, improving the capabilities of tutors, and broadening the availability of tutors constructed form interoperable, reusable components. However, much as the World-Wide Web could not exist prior to the proliferation of both graphical user interfaces and the Internet, the benefits of interoperability and reusability for ITS will not accrue until the necessary technical specifications are applied, and supported by, researchers and developers of intelligent tutoring systems.

In an ideal interoperable ITS research and development environment supported by researchers and developers, it would be possible to readily create customized intelligent tutoring systems from widely available, reusable, components. An interoperability standard would define a set of interface protocols for ITS authoring system modules, allowing the researcher or developer to mix and match components, including pre-existing knowledge bases, in order to meet a particular need. For example, a researcher or developer might build an intelligent tutoring system for teaching *first-aid* by combining a *cognitive apprenticeship* instructional model module from one source with an *emergency medicine* knowledge base from another source, then adding a *human anatomy* simulation module from another source. The researcher or developer might continue by selecting a generic student model software module and an expert model software module from other sources, and then configuring those modules with appropriate data to create the desired ITS. Finally, the researcher or developer would design an overall tutoring strategy by configuring a generic inference engine, and a generic central controller for the ITS.

Each of the software modules in this interoperable ITS could be manufactured by a different software vendor or research lab, and each would follow standard interface guidelines for all communication between modules, and for structuring the content in the knowledge databases for use among all of the interoperable tutoring system components. The ITS component modules would not be tied to any specific content area, so the job of the ITS researcher or developer would be to configure each module so that all the modules together form a robust ITS in the desired content area. In a sense, the researcher or developer would be developing the ITS at a high-level, taking advantage of available ITS components, and developing new components only as needed, in order to quickly test ideas and strategies.

The development of the interoperable ITS could also be facilitated by the use of automated design tools. For example, given adequate interoperable ITS resources, automated "ITS designer" software modules could help the researcher or developer assemble tutors based on high-level instructional objectives. Such ITS designer software would allow the researcher or developer to quickly define objectives, constraints, resources available, and do forth, and then select components and rapidly test various ideas.

Another interesting possibility for a truly interoperable environment for ITS research and development would be to design a distributed ITS that uses resources from across the Internet, automatically selecting and retrieving appropriate interoperable learning resources for use by a central ITS system. Much as the role of the human tutor includes the function of managing the curriculum in order to engage the learner at many levels by finding and using appropriate instructional materials, the interoperable ITS could also search resources from many compatible sources in order to provide the instructional interaction needed by the student. Interoperable ITS systems could work as automated instructional managers, given adequate electronic interfaces between interoperable, distributed electronic resources. This type of distributed system will become possible in an environment in which researchers and developers support and follow standards of interoperability. The success of this syle of distributed, interoperable ITS would benefit from the cataloging of compliant applications, using electronic clearing-houses to maintain current versions of interoperable ITS standards definitions and resources.

This vision of interoperability for the ITS will require a great deal of flexibility in software module designs. It will also require cooperation among researchers and developers, and the sharing of information at a high level. This sharing of information will allow many ITS models to interact and work together to accomplish larger goals. The benefits could include the ability of researchers and developers to rapidly create new tutoring systems, and the ability of the interoperable ITS to meet needs wherever they arise through globally-distributed intelligent tutoring systems.

Possible ITS Interoperability Standards

Interoperability is a general capability that can exist at many levels of a given system. For intelligent tutors, there are various possible levels of interoperability, from the definition of data in their underlying knowledge databases, to the development of common electronic interfaces between ITS and other information systems used in many educational settings. The possible levels of interoperability for ITS research and development will vary depending on the depth and complexity of the tutoring desired, and the type of instructional domain. The following list is an example of possible levels of ITS interoperability. For each of these levels, the development of an interoperability standard would require the definition of interactions and transactions between system components at that level, as well as the definition of the interactions of system components at that level with components at other levels.

Levels of ITS Interoperability

1. Flexible data definitions for various components of the interoperable ITS. These data definitions will include standard definitions for knowledge databases, instructional authoring systems, expert performance models, student skill development and cognition models, simulation environments, inference engines for instructional decision-making, etc.

2. Transaction definitions for interaction between the components of an interoperable ITS. This would include a definition of information needed and

standard processes to be followed by the controlling module of an interoperable ITS.

3.Strategies and communication standards for interaction between the interoperable ITS and outside software, such as productivity tools, or other ITS software.

In addition to these levels of interoperability, the development of intelligent tutors from interoperable, reusable components could benefit from standards for user interfaces. For example, a user interface dictionary could help maintain standardize icons for navigation within an ITS, or standardized utility and maintenance functions (ie, `Exit' always has the same meaning).

Components of an Interoperable ITS

A prerequisite to developing useful standards of interoperability that will lead to successful reusability, will be to identify universal components of an interoperable ITS. A simple definition of possible components can be derived from the traditional model of the ITS. This could include, for example, a student model, an instructional model, an expert model, a database of knowledge resources, an inference engine for decision-making, a local software environment set-up template, and a central controller for managing the resources of the interoperable ITS.

Example Universal ITS Components

1. A *central controller module* for managing the resources of the interoperable ITS. This is the heart of the run-time operation of the standardized tutoring system. The interoperable definitions required for this engine will likely include operational parameters for managing each of the ITS models and knowledge resources. Central controllers will negotiate between various interoperable components acquired via local networks, or the Internet.
2. An *interoperable student model* . This model will allow the central controller module to extract those components of a student model that are useful to the goals for the tutor. This means that a large student model could be composed from components selected at run-time, in order to operate on local data according to the needs identified by the tutoring system manager model.
3. An *interoperable instructional model* definition. Like the student model, this can be a flexible definition of technical parameters for the instructional approaches to be used, indicating the range of techincal capabilities available through the instructional model. The running tutor software will need to be able to acquire and use only those elements of instructional model called for by the tutoring system model.
4. An *interoperable expert model* definition. This will be a flexible definition of parameters for expert performance models, indicating the range of the expert solutions to the tutoring problems. This definition should accommodate popular approaches to evaluating student responses to the tutoring. The expert model definitions will also include appropriate reference information for electronic sources to assessment modules, or other related components of the desired tutor.

5. *Interoperable knowledge resources* . This definition will include standardized access methods and identifiers for the locations and access methods of various instructional knowledge resources, structured according to standardized representations. This may include parameters to direct the operation of those resources, or outline the required operation of those resources.
6. An *interoperable inference engine* for decision-making. The inference engine definition will include standards for the inputs and outputs of the decision-making process, such as the inputs of condition statements and rule sets, as well as the outputs of selected actions to take, or selected advice statements. The interoperable inference engines will be required to adhere to the input and output requirements, but they will be free to utilize their own expert system processes, or other approaches to artificial intelligence, for their own internal functions.
7. A local software *environment set-up* model. This model will be necessary to manage some of the pragmatic issues of a standardized, interoperable, ITS environment. This will include a model of the hardware and software (OS) configurations for a given platform or installation. This will be used by the central controller to apply appropriate system resources for the tutor. This model will also keep track of the location of student files.

One key to the success of an interoperability definition will be the acceptance of a universal set of ITS components, such as the seven interoperable components in the above example. These components, along with their related standards for an interoperable ITS, will only be useful if they are flexible enough to meet the varying needs of ITS researchers and developers, and if they are ultimately designed to benefit the end users of the ITS. To benefit the end users, the standards used for these components must be shared, reside in the public domain, and ITS producers must produce functional components for interoperable tutors that adhere to the standards.

Tasks for the Implementation of ITS Interoperability

In an article in Cause/Effect (Rowley, 1995), I identified tasks for designing, developing, and implementing interoperable educational software systems. These tasks can be applied to the application of interoperability to ITS research and development. Table 1 from that article summarizes these tasks in general terms.

TABLE 1. Tasks for the design, development, and implementation of interoperable systems

DESIGN/RESEARCH

- Developing *design requirements* for educational software interoperability, including the identification of user needs, technology potential, desired outcomes for interoperable educational software designs, acceptable APIs, common approaches to data access, and universal formats for indicators of student performance results.
- Fostering the involvement of appropriate knowledge and information *stakeholders* in areas of education that could benefit from the

increased information available in an interoperable, technology-supported system of education.

- Identifying or developing research tools capable of *studying complex linkages* between the elements of diverse educational software, and composite effects in the environment of a computer-integrated educational system.
- Developing *core models* for interoperable technologies, taxonomies for technology-supported systems of education that would integrate all educational software: instructional software, productivity software, administrative educational computing, automated quality systems, accountability and records systems, decision support systems, and others.

DEVELOPMENT

- Identifying *development issues* through reviews of new standards of interoperability, and case studies of the design and implementation of new standards.
- Asking the right questions to assure that *newly developed products* will have the capability to become interoperable, and the interoperability will be designed to lead to a more customer-centered, student performance-oriented system of education at all levels.
- Determining how diverse applications, including legacy systems, should *coexist* in an interoperable educational software environment.
- To fully exploit interoperability, identifying *new educational software applications* and opportunities made possible by common user interfaces and compatible data sources.

IMPLEMENTATION/MANAGEMENT

- Analyzing *organizational and social issues* related to the integration of interoperability and increased data availability into practice. This will include learning how to address the complexity of data from integrated systems in a manner meaningful to students, teachers and administrators.
- Measuring the *effects* of various implementations of interoperability on: system performance, software features, product acceptance, and most importantly, user learning and performance.
- Defining *evaluation standards* and benchmarks for components of interoperable educational software systems.
- Determining *soundness* of implementations. Interoperability standards should be developed with input from those involved in the real-world implementation problems of the related hardware and software technologies.
- Analyzing the short- and long-term *costs and benefits* of educational software interoperability.

In summary, there are many critical factors in the design and implementation of standards of interoperability for researchers and developers of intelligent tutoring systems. The benefits of such interoperability standards could be significant, but a number of important technical and implementation issues must be addressed before cost-effectiveness and reusability will occur. This will include developing the needed researcher and developer support base for the standards, determining which interoperability standards are needed, identifying acceptable universal components of an interoperable ITS, and resolving numerous technical and social issues related to implementing such standards. As these issues are addressed, the interoperable ITS should become a flexible development environment that will provide researchers and developers with a more cost-effective, robust means of testing ideas and designing increasingly effective intelligent tutoring systems.

References

Rowley, Kurt. (1995). Understanding software interoperability in a technology-supported system of education. Cause/Effect , 18 (3), p.20-26. (this article is also available over the world-wide web, see http://cause-www.colorado.edu/pub/pubs.html)

Biography

Kurt Rowley has worked and written in the area of interoperability for educational software. He helped initiate an interoperability standards effort for the State of Florida's Educational System, under the sponsorship of the Florida Department of Education's Schoolyear 2000 program. He currently works as a research associate in the Intelligent Training Branch of the U.S. Air Force's Armstrong Laboratory in San Antonio, where he has recently completed the design, and is managing the development, of an ITS to teach English composition skills.
