Open Protocols for Web-based Educational Materials

Terence Ahern†  Nancy Van Cleave‡
Brooks York⋆

†Texas Tech University
Department of Educational and Instructional Technology

‡Eastern Illinois University
Department of Mathematics

⋆Austin, TX
Introduction

Current technology is under–utilized in our public schools

- **High cost** of proprietary systems
- Lack of **software availability** for legacy computers, different platforms
- Lack of **software flexibility** to adapt to needs
- Steep **learning curves**; little invested in training or support
Underutilization of Technology—continued

- Lack of **time** and **energy**—how is technology to be integrated?
- limited classroom **usefulness** in general
Relevant Educational & Technological Issues

Considerations

- Proprietary vs Open Systems
- Separating Content from Delivery
- Web-based Education and Browsers
- Creating Educational Protocols
- Instructional Tools Requirements List
Proprietary vs Open Systems

Most current educational products and software are proprietary.

Closed-system designs usually:

- are more expensive—both to license and to maintain
- hamper third-party add-ons or extensions to system
Open Systems

Open is the opposite of closed.

- much less expensive, if not free
- protocols and data representations are published publicly
- often require less server and system admin support
- sometimes less functionality, fewer administrative-type tools
Separating Content from Delivery

Electronic gaming industry: delivery engine

• saves development time and expense

• splits task into expertise areas:
  – programmers create authoring and presentation software
  – instructional designers create the presentation models (matching content with appropriate delivery model—the protocols)
  – educators produce the lesson information—the content
• ability to easily share materials
• modification is much easier
• current educational software is integrated, making it difficult and expensive to modify
Web-based Education and Browsers

The Next Great Educational Cure-All—the Internet!!

- use as an educational tool is taken for granted
- provides access to a wealth of information
- any web-based instructional tool must direct learner through material
- could use browser to launch presentation engine
Creating Educational Protocols

There are many distinct *instructional paradigms* for content delivery... ranging from lecture to small–group discussion... each of which needs its own protocol.

Different delivery methods are more appropriate for different types of content.

Software must perform flawlessly and apply the correct instructional delivery strategies (sequencing, or how a user interacts with and navigates the content).
Educational Protocols

Two basic choices:

1. develop proprietary delivery environments
   • proprietary software
   • difficult to adapt to specific needs
   • difficult to maintain or modify on demand

2. instructional development environment (e.g., WebCT or Toolbook)
   • ignores multiple instructional paradigms
Educational Protocols

Once developed, protocols must be made public and disseminated widely.

Need a mechanism to:

• create tutorials and other instructional aids
• these materials must be recognized, parsed, and delivered by the presentation engine
• associate the appropriate educational protocol with the correct material
Educational Protocols

Open delivery protocols provide greater flexibility for software developers.

XML can be used to develop standard instructional delivery protocols.

The end user is allowed more creativity in mixing and matching instructional materials—providing more control over content.
Instructional Tools Requirements List

Any computer-based instructional tool should satisfy the following constraints:

- Utilize open standards and be accessible through the Internet.
- Be cross-platform (available on a wide variety of machines and configurations).
- Separate content from delivery mechanism, allowing educators to control and share components.
- Provide creation, editing, and delivery mechanisms.
Satisfying Constraints

We can break educational software into three components

- authoring software
- presentation software
- an instructional delivery protocol
Suggested Solutions

• an open protocol for tutorials based on a Document Type Definition (DTD) in XML

• A Java application which incorporates the DTD in order to produce tutorial files consisting of information (course content) such as filenames and URLs, which conforms to the DTD

• a Java applet which can be used to view the tutorial files created by the Java application
Open Standards

Educational material must utilize public interfaces and openness to take advantage of the Internet

- Document Definition
- A Tutorial Protocol
- Programming Language
- Cross-Platform
- Separation of Content from Delivery
- Creation, Editing & Delivery Mechanisms
Document Definition

Extensible Markup Language (XML) is the metagrammer used to define the grammar of the logical structure of a document.

The resulting set of rules, the DTD, dictates:
- how the elements within the document relate to one another
- which elements can contain other ones
- in what order the elements can occur.

The DTD provides a protocol that may be shared across platforms.
A Tutorial Protocol

Gagne’s Nine Events of Instruction

1. gaining attention

2. informing the learner of the objective

3. stimulating recall of prerequisite material

4. presenting the stimulus material

5. providing learning guidance
Gagne’s Nine Events of Instruction—Continued

6. eliciting the performance

7. providing feedback about performance correctness

8. assessing the performance

9. enhancing retention and transfer
A Syllabus DTD in Tree Form
Programming Language

Java and XML are generally accepted in the Internet community and are extremely flexible to use.
Cross-Platform

Java, XML, and any web browser which supports them, satisfies the cross-platform issue.
Separation of Content from Delivery

The DTD allows the separation of content from delivery

• can use DTD as stand-alone or integrate into document

• allows flexibility and control, easier development and authoring

• cheaper and takes less development time
Creation, Editing & Delivery Mechanisms

An authoring and editing application was developed in Java to match instructor’s input (filenames and URLs) to the DTD format.
Viewing the Tutorial
Summary

Currently there are two large extremes available for educational software

1. Off-the-shelf package
   - bundling content with delivery engine
   - expensive
   - difficult to modify
The Other Alternative

2. Authoring systems which allow the user to control content

- no control over delivery paradigm
- requires extensive training and authoring time
- may be limited in the types of material the author is allowed to include.
Summary of Work

We provide a feasible, attractive alternative in which

- course authors maintain control over content, with easy development and editing
- instructional designers provide the instructional protocols
- software engineers develop and maintain the tools
- students get correct content in the correct manner at the correct time
Conclusion

We have shown the feasibility of

- an **open standard** for instructional materials through XML
- **cross-platform compatibility** through use of Java, XML, and the Internet
- separating **content** from **delivery** through XML
- creating an **authoring** and **editing** application using a Java application and an applet