Tech., Inst., Cognition and Learning, Vol. 3, pp. 25-31
Reprints available directly from the publisher
Photocopying permitted by license only

Learning Objects: Promise versus Reality

JOSEPH M. SCANDURA*

MERGE Research Institute
Visiting Research Professor, Drexel University and
Adjunct and Emeritus Professor, University of Pennsylvania

Creating standards for LOs has the potential of increasing opportunities for reuse. What a LO is, however, often depends on undisclosed assumptions regarding media, content and pedagogy. Abstract Syntax Trees (ASTs) (Scandura, 2003, 2005) offer one promising candidate for defining LOs. All knowledge can be represented as ASTs. Furthermore, LOs correspond to nodes in ASTs, making it possible to precisely define both media and content. It also has been shown recently that essentially any pedagogy can be defined independently of content (see TutorIT in Scandura, 2005). Nonetheless, we must be cautious. An inadequate 'learning object" paradigm (SCORM, IMS) could easily become dominant – due to political or commercial clout – not technical superiority. At present, the best we can do is assume de facto standards for media. The danger lies in premature definition of standards for content and pedagogy.

Keywords: Learning Object, reuse, abstract syntax tree, SCORM, IMS, Knowledge Representation.

INTRODUCTION

Neither learning objects (LO), nor related work on Standards, has been a significant focus in my research. I feel obliged, however, to raise some

* Corresponding author: scandura@scandura.com

cautionary notes, along with positive suggestions, based on a long history of research in both education and technology.

The main argument in favor of LOs, including Standards for LOs, is reuse: Build instructional modules once and reuse them in other learning systems. The potential benefits of reuse are enormous. Nonetheless, I urge caution. Beginning with a short look at the origins of LOs, I then ask what a LO actually is. Next, I outline the role of Abstract Syntax Trees (ASTs) in the Structural Learning Theory (SLT) and how they might be used to define LOs. Finally, I show how standards for LOs impact Media, Content and Pedagogy.

SCIENTIFIC & TECHNICAL FOUNDATIONS

Like many new ideas in education, LOs are adaptations of ideas developed in more established disciplines. Indeed, education has had more than its share of misapplications of science. In many cases important developments in established sciences have been adopted in education too uncritically.

How many recall the application of Skinnerian behaviorism to teaching machines in the 1960s? It's not that human behavior cannot be shaped like pigeons, but rather that shaping is far too inefficient given the sheer magnitude of what must be learned by humans in any advanced society.

LOGO illustrates another extreme. It is not that the idea of recursion cannot be useful in problem solving. Rather, recursion is only one of many kinds of higher order thinking. Early LOGO enthusiasts are not fundamentally different from proponents of mental discipline at the turn of the last century - i.e., 1900. The latter believed that simply studying "hard" subjects, like Latin and geometry, would automatically teach people how to think. It might! But how, why and under what conditions?

One can easily find other misapplications of science and value laden philosophical thinking in education. As constructivists are fond of pointing out, for example, people certainly can —and sometimes even do think. But to simply assert this fact only unlocks the door. The real questions are: How are we to know what people are or might think in given situations? And, how can we develop good thinking habits?

LOs were motivated by object-oriented design/programming (OO) in software engineering. Software engineers understand the important advantages of OO: Organized, hierarchical packaging and the ability to

reuse code are generally considered most important. More specifically, OO puts an emphasis on: abstraction, encapsulation, inheritance, polymorphism and message passing.

Even in software engineering, however, OO comes with undesirable baggage:

- OO puts the emphasis on packaging — not behavior – telling problems to solve themselves or screens to reformat themselves vs. solving problems or reformatting screens. Both packaging and behavior are important. Packaging organizes objects for reuse. Behavior is what educators actually observe.
- Message passing applies when talking about communication between different entities — teachers and learners, but not when talking about how an individual solves problems. Even in the former case it is not so much telling an object what to do as the object receiving a message and DECIDING what to do.
- It can be difficult to adapt objects to meet specific needs. It is one thing to have objects for producing displays for example. It is quite another to find, or adapt one to achieve a desired result.

DEFINING LOS

LOs adopt only a small part of what OO is about. The term itself, however, has been applied to everything from media to content to pedagogy. How to determine what learning objects are needed? The sheer variety of content, and varied needs of learners – exceeds by orders of magnitude anything conceived of in most software applications, where objects tend to fall in a far smaller number of categories – e.g., screen displays.

LOs depend on what the learner already knows, and how well. No one set will suffice — rather a plethora of LOs representing different kinds and levels of expertise.

LOs are different than objects in OO hierarchies OO objects are static in nature. Actions are packaged - hidden in those objects. In OO programming, software tells cars to drive themselves, or numbers to add themselves, or sentences to analyze themselves.

In education, the only thing we can observe is behavior. Can the learner drive a car? Or, add numbers? Or, diagram sentences? LOs are dynamic, not static. Behavior also becomes increasingly automatic at higher levels of expertise. Different levels of expertise call for different LOs.

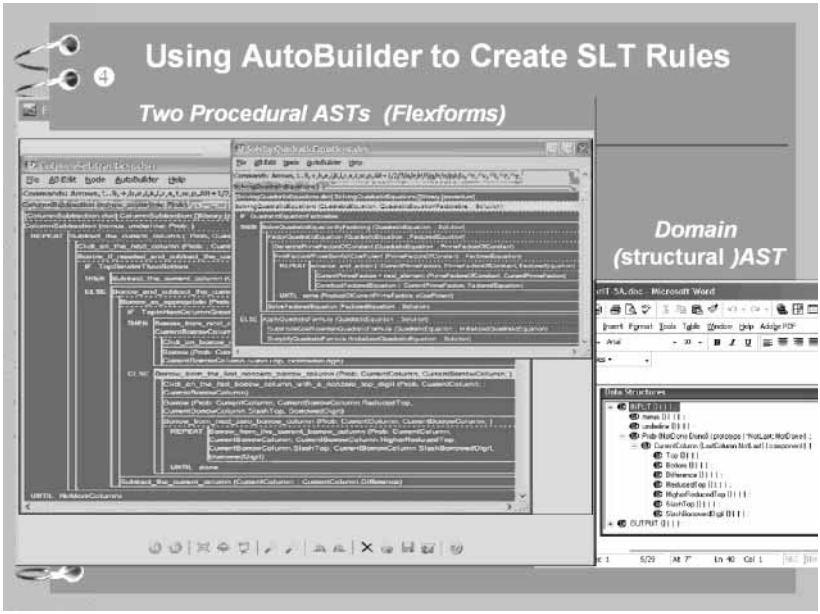

FIGURE 1
AST representations of procedural and data ASTs as they appear in AuthorIT's AutoBuilder component.

DEFINING LOS VIA ABSTRACT SYNTAX TREES (ASTS)

Hierarchies are universal. They can be observed in content as simple as column subtraction, and as complex as analyzing poetry or proving theorems in topology. Rules of knowledge in the Structural Learning Theory (SLT) are represented hierarchically, as ASTs (Scandura, 2001, 2003).

The AuthorIT ITS authoring system was built on this foundation (Scandura, 2005). Nodes in ASTs correspond to specific chunks of knowledge at multiple levels of abstraction. Instruction is associated with individual nodes in ASTs. These nodes correspond directly to LOs. They might be Flash, audio or any similar kind of file. LOs in AuthorIT are plugged into nodes by simply changing file references. Nodes in ASTs effectively define LOs.

Figure 1, for example, shows what an abstract syntax tree looks like in AuthorIT – specifically data structure and procedural hierarchies associated

with column subtraction and solving quadratic equations. Whereas most people would consider the latter topic more advanced, notice that procedural subtraction AST is actually a bit more complex. The complexity of a knowledge representation depends not just on the content but also the size of the building blocks – prerequisites assumed of the intended learner population. LOs are inherently dependent on content, level of expertise and target population.

All knowledge can be represented as ASTs – declarative/structural as well as procedural, domain dependent and domain independent, lower order knowledge and higher order knowledge. Furthermore, AST hierarchies can be constructed in a systematic manner via structural analysis. Structural analysis is a specific form of cognitive task analysis that can be continued indefinitely, via a small number of kinds of refinement (Scandura, 2003), until prerequisites assumed for the learner population have been identified.

This is not the place to explicate ASTs, however, or to compare them with other forms of representation. This has been done elsewhere (esp. Scandura, 2005). IT researchers have had different opinions over the years as to what constitute the best representations. The essential point here is that the way knowledge is represented plays a profound role in determining what LOs are identified and whether they will be useful.

STANDARDS FOR LOS AND REUSE.

What about standards? Some believe that creating standards for LOs will dramatically increase opportunities for reuse. Several groups (SCORM, IMS) are working to develop such frameworks independently.

Standards can certainly be useful — like making sure different video players use the same media. They also can be a major impediment to creativity and progress — especially when imposed early in the evolution of a technology. In software engineering, after 20 years of research, there is considerable difference of opinion on the value of the so-called Unified Modeling language (UML). Reducing programming to “pictures” is not only incomplete, but in many ways more complex than the programming it was meant to replace. Better no framework than a poor framework that institutionalizes the status quo and impedes progress.

What might be standardized in IT then? Media, content and pedagogy are three essentials in IT development. Media is already governed by de facto standards: Most IT learning systems make heavy use of media players–

Flash, Media Player, Sound Recorder, etc. by giants like Macromedia and Microsoft. Such tools have near universal application and IT's influence in their evolution will be limited. Nonetheless, media files serve as LOs – they can easily be reused by any IT system supporting relevant tools (e.g., Flash Reader). Accordingly, the documentation, classification and broad availability of such media could provide a potential source of reuse in IT.

Realizing reuse in practice, however, is often illusory. The potential universe of LOs is indefinitely large, making it difficult to create more than a small fraction of what may be useful. Even when a LO exists, one has to find or adapt it. Nonetheless, developing repositories of LO media has little downside and potential good. We should do it!

The situation with content is more problematic. There is little agreement on how content should be represented. Production systems, relational networks and hierarchies have all been used for this purpose (cf. Scandura, 2005). The increasing use of XML in web browsers has led some standards groups to propose XML as the preferred means of knowledge representation. XML is hierarchical and broadly available, and media can easily be associated with nodes in XML. Nodes effectively define the semantics that corresponding media are to convey.

XML, however, is largely static in nature. Foshay and Preese (2005), for example, caution that interactivity in eLearning today is more limited than CBI in the early 1990s. XML makes it difficult to duplicate highly dynamic interactive functionality.

The use of ASTs in AuthorIT directly addresses this limitation. Like XML, ASTs are hierarchical — but they are also fully executable. LO media are attached to semantically meaningful nodes in ASTs, making them easily the equivalent of XML. More important, the software infrastructure is largely in place in the form of AuthorIT. Moreover, AuthorIT is freely available for research purposes.

Nonetheless, it is premature to say that everyone should adopt ASTs as **the** standard for representing content. Further R&D, yes! And, real world application too! But, we should be very cautious when it comes to imposing standards at this point.

What about pedagogy? One approach to be discouraged would be standard LOs for teaching various kind s of knowledge - facts, concepts, procedures, etc. More generally, Friesen (this issue) cautions that standards invariably impose pedagogical constraints of one sort or another.

TutorIT (the general purpose delivery component in AuthorIT) (Scandura, 2005) raises another interesting possibility. AST structures alone

provide a sufficient basis for specifying essentially any pedagogy, independently of content semantics – whether the pedagogy is highly prescriptive, strictly constructivist or otherwise. TutorIT can easily be customized to deliver content as desired by simply filling in a dialog box. TutorIT is installed locally, working like Flash Reader. Effectively, TutorIT is just one "great big LO" — covering *all* pedagogies.

SUMMARY

The good news first. We already have de facto standards for media. The danger lies in standards for content and pedagogy. An inadequate 'learning object" paradigm could easily become dominant – due to political or commercial clout — not technical superiority.

Hopefully, current efforts to prematurely standardize content and pedagogy will prove no more likely to dominate IT than did CASE or Ada dominate software development 15 years ago — despite strong attempts by the military to impose them.

REFERENCES

Foshay, W. R. & Preese, F. Do we need Authoring systems? A commercial perspective. *Technology, Instruction, Cognition & Learning (TICL)*, 2005, 2, 3, 249-260.

Friesen, N. *Technology, Instruction, Cognition & Learning (TICL)*, 2005, 3, 1, this issue..

* Scandura, J.M. Structural learning theory in the year 2000. *Special Monograph of Journal of Structural Learning and Intelligent Systems* (a special monograph), 2001, 14, 4, 271-305. (Similar version sans Appendices in Scandura, J.M. Structural Learning Theory: Current Status and Recent Developments. *Instructional Science,* 2001, *29*, 4, 311-336.)

** Scandura, J.M. Domain Specific Structural Analysis for Intelligent Tutoring Systems: Automatable Representation of Declarative, Procedural and Model-Based Knowledge With Relationships to Software Engineering. *Technology, Instruction, Cognition & Learning (TICL)*, 2003, 1, 1, 7-57.

** Scandura, J.M. AuthorIT: Breakthrough in authoring adaptive and configurable tutoring systems.. *Technology, Instruction, Cognition & Learning (TICL)*, 2005, 2, 3, 185-230.

* *On-Line at: www.scandura.com under TICL On Line*

** *On-Line at: http://www.oldcitypublishing.com/TICL/TICL%201.1%20contents.html*