Discussion paper for UNFOLD on-line seminar of January 2005

Learning Design and Representations of Instructional Intent
Griff Richards and Colin Knight

Some thirty years ago when educational technology was focused on audio-visual media production Edgar Dale proposed his “Cone of Experience” (http://www.fsu.edu/~ids/fac2002/Edgar%20Dale.htm) – a typology of media messages that moved from simple to greater realism, from text messages through pictures with text, sound/slide shows, film and simulation. The apex of the cone represented higher levels of abstraction such as text which are symbolically more efficient at storing and referring to an experience than the lower, broader configurations that provide a more realistic experience. Presumably a designer of media could choose the level of presentation to optimally replicate the level of fidelity required for the communication of the instructional goal.

Dale is often credited with the maxim that “People learn best when they are actively involved in the learning process”. And that that

People generally remember...

10% of what they read.
20% of what they hear.
30% of what they see.
50% of what they see and hear.
70% of what they say or write.
90% of what they say as they do a thing.


In considering the level of documentation that might be required to enable the replication of a learning experience, I drew a parallel between Dale’s model and the ways and means in which instructional plans might be described. Learning Designs are highly abstract representations – after all, they are rich text XML representations more designed to let instructional designers interact with computational “players” than they are “lesson plans” that could be easily read and interpreted by an instructor looking for innovative lesson plans. Yet some Learning Design systems use “human-friendly” graphs to enrich the visual appeal of XML documents, the diagrams presumably help those creating Learning Designs by providing a more intuitive interface than writing XML directly, and they should also enable the reader to interpret and understand the designs. As Buzza et al (2004) clearly state, it is important not to confuse IMS Learning Design, the formal XML expression with “learning designs”, the more general expression of an instructional design.
This discussion papers simply reviews some of these representations and ask the seminar participants to discuss the pro and cons of these rich representation systems for augmenting and interpreting Learning Designs so that they can more richly encode learning designs. I do not have a goal to promote nor critique any particular system. We need to remember that they arose to suit particular needs and of developers as they integrated software engineering techniques into education.

**Representation: Pattern Languages**

Probably best typified by Alexander’s (1994) text on architectural patterns, patterns language are used to control complexity by describing observations (or desired observations) about a system. A pattern language creates a classification system for such observations within a particular domain so that for those familiar with the patterns it becomes easier to recognize, describe and discuss relationships of elements in a system. Perhaps the most important aspect of pattern languages is recognizing that some one unfamiliar with the jargon with probably not understand the discussion. For example, the Versailles activity might be described as a “cooperative learning jigsaw” a pattern readily recognizable to those familiar with cooperative games, but the uninitiated will require background study.

Perhaps a subset of pattern languages are activity patterns – for example, the Eight Learning Elements Model (Verpoorten, 2004) provides a schema of eight dominant activities according to the main activity of the learner:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receives</td>
<td>passive listening or reading</td>
</tr>
<tr>
<td>Imitates</td>
<td>modeling of instructor’s actions</td>
</tr>
<tr>
<td>Practices</td>
<td>repetitive drill and practice (with or without external feedback)</td>
</tr>
<tr>
<td>Explores</td>
<td>may be cognitive or manipulative</td>
</tr>
<tr>
<td>Creates</td>
<td>produces a new product from given materials</td>
</tr>
<tr>
<td>Experiments</td>
<td>methodological exploration and reflection</td>
</tr>
<tr>
<td>Debates</td>
<td>takes a position, justifies and defends with evidence and logic</td>
</tr>
<tr>
<td>Meta-cognitive</td>
<td>reflects on learning and approaches to learning</td>
</tr>
</tbody>
</table>

Table 4. 8LEM learning events model (after Verpoorten, 2004)

Thus for an computer exploration activity we could say the learner

1. receives a visual/textual clue
2. explores a scene by moving the flashlight to likely targets
3. experiments by clicking on the target and seeing the result
4. practices the targets until capable of matching quickly

The advantage of the activity patterns is that they shift the focus from the instructional program to the learner activity.

**Representation: Shared Mechanics**
Shared mechanics is also a sub-category of Pattern Languages, but here the focus is on the mechanics of the game. For example, an educational game activity might be mounted on a Doom engine. Those familiar with Doom will instantly know what to expect and what roles will be played. Those not familiar with Doom will have to read or better yet observe or experience the game before understanding this type of activity.

**Representation: Flowcharts**

Flowcharts are simple process control models that indicate processes and logic at critical decision points. These can be effective for documenting simple processes, but complex diagrams may require several levels of documentation. The main shortcoming of flowcharts is that they do not distinguish the actors involved, and it is difficult to represent multiple threads of action.

**Representation: UML Activity Diagrams**

Universal Modeling Language (UML) provides a standardized method for visually representing activities and processes, making it ideally suited for depicting learning designs. There are a variety of UML diagrams but the collaboration and sequence diagrams are probably the most relevant to Learning Design. Tattersal (2004) provides an example of the sequence or “swimlane” activity diagram, a UML diagram that provides each actor with a column, and into which the activities are listed sequentially so that the overall pattern of actions and dependencies are represented. These representations are powerful because they are visual and they make it easier to see the sequences of a process, the interactions among actors, and how the various parts fit together to make a whole.

The other common type is the collaboration diagram that shows the actors and relationships between them. The sequence of events is normally read from left to right and along the conjoining arrows. The Versailles activity in the IMS-LD best practices guide is of this type. The collaboration diagram depicts a holistic view of an activity, but the sequence can be difficult to follow when there is a large number of actors or interconnections.

When considering the suitability of UML as a representation, it is also important to remember that UML is a visual language, primarily intended for use by software developers, that requires a degree of familiarity with its vocabulary and grammar to properly interpret the diagram. Although UML is designed to be highly intuitive, the potential complexity of activity diagrams can obscure the meaning to a large potential audience: educational professionals that may not have training in software development and UML.

How suitable are activity diagrams for describing learning interactions? In general, they are very effective in describing many types of interactions. They are most effective describing learning activities that have:
• structured interactions between players
• sequential or concurrent sequential activities
• between five and twenty-five elements - enough to fill a page but not enough to overwhelm the diagram

Both sequence diagrams and collaboration diagrams can represent the same activity, the choice is a pragmatic one depending on whether one wishes to focus on the sequence of events, or the relationships among the actors.

**Representation: Meta-Knowledge Model Diagrams**

Paquette (1998) uses a specialized type of diagram that attempts to represent knowledge in a wide variety of domains, including learning design, as a collection of meta-knowledge objects linked together by concepts such as instantiation, composition, specialization, precedence, and regulation. Essentially a rich UML diagram initially developed for an instructional engineering system, Paquette’s(2005) LICEF group has added enhancements to enable MOT Plus to serve as an LD editor.

The advantage of this type of diagram is that it is potentially very powerful, and can represent a wide variety of concepts, processes, and domain knowledge. The disadvantage is that any reader of this type of diagram would need to also read a primer on meta-knowledge representation and the conventions of this diagram system to be able to properly understand the meaning. For example the shapes of the diagram components convey information about the nature of each element, while the links of the MOTPlus graph has a semantic marker identifying the type of relation. For example, I = instantiation, C= composition, P = precedence, R= regulation, I/P = an input or product.

The MOT Plus editor provides an excellent user interface for creating Meta-Knowledge Model Diagrams. The editor has even added modes for different domains. In the LD domain, most basic objects from LD, including play and act, have been added to the toolbar. This makes it easy to quickly create diagrams of LD, and the learning curve for using the tool is very short. However, if one is not familiar with Meta-knowledge representation, it is doubtful that one would choose the correct link type from the wide variety of available links. If the correct link type is not chosen, it would confuse anyone familiar with the various link types and meanings.

**Representation: Multiple Elaboration Model**

The last representation category is not a single representation, but rather collects multiple descriptions of a learning activity to give the most complete view possible. The work that best typifies this approach is that of Buzza, Bean, Harrigan and Carey (2004) in creating prototype learning design repositories. Rather than focusing narrowly on the Learning Design, their proposed repository documents instructional scenarios - essentially case studies that provide both a contextual i.e. domain-specific example of the learning design, results of evaluation, technical resources required and the more formal expression as a Learning Design. The resultant collection could be searched from a variety of
perspectives. Even with these rich representations Buzza et al (2004) still note the need for a controlled vocabulary to describe the learning intentions. As with Pattern Languages the development of a discourse that developers and educators and build upon will require agreement on the words that we use to describe abstractions.

**Summary**

This paper has raised some representational issues for the documentation of learning activities so that they may be replicated or repurposed with new content. The issues are many and can go beyond the few raised here, but the intent is that such discussion will help set community goals about the degree to which Learning Design should be expected to provide replicability of learning designs, and the need to augment these representations with other types of documentation.

**References**


http://www.imsglobal.org/learningdesign/ldv1p0/imsld_bestv1p0.html


http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html

