Open Learning Objects: The Case for Inner Metadata

O. Rodriguez, S. Chen, H. Shi, & Y. Shang
Department of Computer Engineering & Computer Science
University of Missouri-Columbia

Motivation

- Learning objects are the basic units of learning material in e-learning.
- There are standards for packaging learning objects (IEEE, IMS).
- However, learning objects remain opaque entities.
  - little visibility and controllability into their internal composition and operation
  - limited adaptability to learners

Open Learning Objects

- driven by inner metadata
- multi-layered
- multi-media based
- adaptable to individual learners
- interactive
- delivered through agents
- implemented using open standards

Inner Metadata (1)

- Current LOM standards only provide LO description from external viewpoint.
  - ✓ Metametadata, General, Lifecycle, Rights, Technical, Educational, Classification, Annotation, Relations
- To effectively use LOs, e-learning systems need access to LO internals, in a vendor-independent standardized way.
- We define InnerMetadata as a set of markup languages for the internal structure and composition of a LO.

Inner Metadata (2)

- Text.dtd
- SVG.dtd
- Animation.dtd
- Assessment.dtd
- Interaction.dtd

Multimedia Based Content

- Multimedia helps accommodate a large variety of learners with different learning styles and skill levels.
- Open learning objects adopt text, SVGs, and animation to organize learning material.

Adaptation Dimensions

- OLO design accommodates not only multiple media preferences but also more pedagogically meaningful adaptations.
  - Language
  - Skill level
  - Learning style
  - Accessibility

Layered Architecture (1)

- The inner metadata (defined by DTDs) is organized in layers.
- The DTDs plus semantics provide an open way to adapt and enhance learning objects.
- We define five layers:
  - Concept
  - Presentation
  - Animation
  - Interaction
  - Integration

Layered Architecture (2)

- Each layer features the same structure:
  - Content sub-layer
  - Metadata sub-layer
    - Adaptation sub-sub-layer
- Advantages of layered Architecture
  - Decoupling of content sub-layers
  - Decoupling of adaptation sub-sub-layers
  - Independent authoring of layers
    - Scalable OLO authoring by fostering collaboration
    - Easy integration and coordination
    - Per-layer specialization

Concept Layer

- Content sub-layer
  - LO knowledge domain concepts, Definitions, Explanations, Examples, Exercises, Assessments
- Metadata sub-layer
  - Pedagogic concepts, Ontology (expressed in OLOconcept.dtd)
- Adaptation sub-sub-Layer
  - Prior knowledge
  - Language
  - Learning style and skill level

Presentation Layer

- Content sub-layer
  - Alternate representations for concept layer, using graphic metaphors, shapes, icons, speech markup
- Metadata sub-layer
  - Shapes, icons, clips or synthesized speech (expressed in SVG graphics and speech markup language)
- Adaptation sub-sub-Layer
  - Accessibility adaptations: color, contrast, size, fonts
  - Text-to-speech

Animation Layer

- Content sub-layer
  - States representing a learner-meaningful configuration of the concept & presentation layers
- Metadata sub-layer
  - Synchronization of the concept and presentation layers: Sequencing of states with branch/actions that animate representations, default and optional branching, backtracking
- Adaptation sub-sub-Layer
  - Line-of-reasoning (LOR): cognitive load & transition grain
  - Ordering of LOR groups per skill & learning style

Interaction Layer

- Content sub-layer
  - Primitive events (represented with graphic shapes and other UI widgets)
- Metadata sub-layer
  - Sequencing of states with branch/actions activated by events
- Adaptation sub-sub-Layer
  - User controls for OLO navigation

Integration Layer

- Content sub-layer
  - References to other layer's documents
  - Agents to be integrated with the OLO
- Metadata sub-layer
  - LOM metadata <Relations> elements to link OLO components
- Adaptation sub-sub-Layer
  - Adaptation to learner and browser being used by integrating different agents

Generic Agents for OLO

- Generic software agents are driven by OLO layers to integrate components, animate presentation, and trace interaction with the learner.
- These agents embed the markup semantics and provide a key function: adaptation.
- These are: Learner Agent, Course Agent, Facilitator Agent.

Implementation of Agents

- Agents coded with open-source software.
- Open-source components include
  - Apache: Webserver
  - Tomcat: Servlet container
  - Cocoon: XML content processing
  - Xerces: XML parsers
  - Xalan: XSLT transform engine
  - ECMAScript, SVG viewer, IE/NS

Facilitator Agent (1)

Adapts OLO on-the-fly for skill/style

- Interprets the animation layer file
- Handles back-tracking
- Collects learner interaction trace
- Implemented as a “situated” or “reactive” micro agent

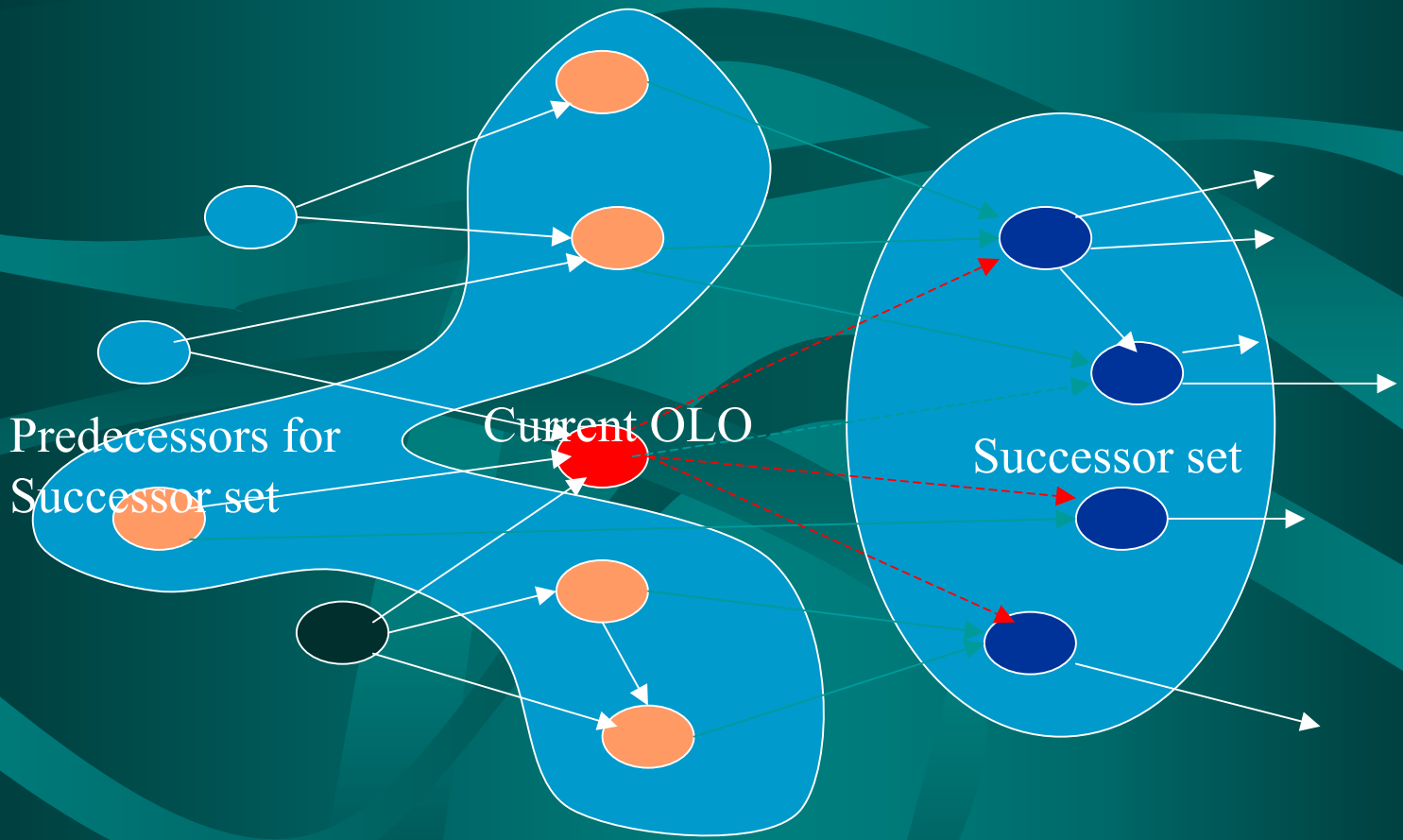
Facilitator Agent (2)

- Uses a blackboard metaphor (virtual classroom)
  - User-controlled or automatic step-by-step presentation
  - Backtrack allowed
- Maintains stacks for
  - events
  - backtrack
  - interaction trace

Course Agent

- Enables adaptive navigation between OLOs
- Performs OLO integration and server-side adaptations such as language and skill level
- Uses XSL stylesheets for adaptation of OLO content model
- Implemented as servlets and JSPs
- Co-located with learner agent

OLO Adaptive Navigation



Learner Agent

- Encapsulates an abstract LIS-independent representation of the learner model
- Updates the learner model with the interaction trace
- Handles learner authorization and authentication
- Implemented as servlets

Summary (1)

- OLOs as an open way for curriculum decimation
  - Decimate curriculum into a set of standardized topics to be learned/taught.
  - Devise/administer assessment items for OLO-scope topics.
  - Let teachers and authors gradually come up with blended-versions and ever more improved versions of OLOs meeting the assessment requirements on the topics.

Summary (2)

- Open learning objects simplify tasks:
  - content adaptation
  - learner interaction trace and learner modeling
  - adaptive navigation
- Open learning objects provide better support for personalized e-learning.

Questions?
Answers!

Thank YOU!