

Chapter 2.12

Engineering Reusable Learning Objects

Ed Morris

RMIT University, Australia

ABSTRACT

We adapt the object-oriented software engineering design methodology for software objects to engineering reusable learning objects. Our approach extends design principles for reusable learning objects. The resulting learning object class is a template from which individualised learning objects can be dynamically created for, or by, students. The properties of these classes refine learning object definitions and design guidelines. We adapt software object levels of cohesion to learning object classes. We demonstrate reusability increases when learning object lessons are built from learning objects, like maintainable software systems are built from software objects. We identify facilities for learning management systems to support object-oriented learning object lessons that are less predetermined in sequencing activities for each student. Our overall approach to the design of learning object lessons is inde-

pendent of, and complementary to, instructional design theory underlying the learning object design process, and metadata standards adopted by the IEEE for learning object packaging.

PERSPECTIVE

We approach the shared aim to design reusable e-learning objects by adapting software engineering methodology, where it enhances the reusability of software objects. Our work (Morris, 2005) contributes software engineering techniques to the design and evaluation of learning objects with enhanced reusability.

We also provide a different perspective on the continuing pedagogical debate over granularity and context for optimal e-learning object reusability (Littlejohn, 2003) as we focus on the user interface and internal structure of a learning object to enhance its reusability.

OBJECTIVES

The objectives of this chapter are:

1. To explain how object-oriented software engineering design methodology can be applied to the design of a learning object to enhance its reusability. See the “Designing Learning Objects as Software Objects” subsection and the “Example Learning Object Classes” section.
2. To show how object-oriented software engineering extends and refines Boyle’s (2002) design principles for authoring dynamic reusable learning objects by enabling individual learning objects to be dynamically created for or by students from a template learning object, which we call a learning object class. See the “Designing Learning Objects as Software Objects” subsection and the “Example Learning Object Classes” section.
3. To contribute toward a learning object lesson design methodology that will facilitate the design and implementation of larger scale lessons, courses, and educational programs. See the “Developing Learning Object Lessons as Software Systems” subsection and the “Object-Oriented Design Principles for Learning Objects” section.
4. To explain how the properties of a learning object class refine existing learning object definitions and design guidelines. See the “Criteria That Define Object-Oriented Learning Objects” subsection.
5. To show how reusability is further enhanced by standardising the interface of a learning object class to provide its learning activities as services that can be invoked by other learning objects. See the “Developing Learning Object Lessons as Software Systems” subsection and the “Example Learning Object Classes” section.
6. To explain how software object levels of cohesion can be applied to the design of a learning object class, such that the higher the level of cohesion, the more it is reusable. See the “Object-Oriented Design Principles for Learning Objects” section.
7. To identify facilities required in a learning management system to support learning object lessons that are less predetermined in their sequencing of activities for each student. See the “Support for Object-Oriented Learning Objects” section.
8. To explain how our object-oriented software engineering approach to the design of learning object lessons is independent of, and complementary to, (a) instructional design theory underlying the learning object design process, and (b) metadata standards adopted by the IEEE for learning object packaging. See the “Support for Object-Oriented Learning Objects” section.

INTRODUCTION

Early research and development of online learning materials did not focus on their reusability. For example, our previous research focussed on the cost effectiveness (Zuluaga, Morris, & Fernandez, 2002) and educational effectiveness (Morris & Zuluaga, 2003) of our online learning approach. This involved both online course development and online course delivery phases. We also addressed the deployment, management, and scalability of our online courses over a network of learning management system servers (Zuluaga & Morris, 2003).

Most of our early online courses (1999-2002) were developed for 100% online delivery, utilising mostly textual learning materials, plus (on average) four short multimedia supplements such as Java applets, Flash animations, voice overs, and video clips. During online delivery of such

16 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/engineering-reusable-learning-objects/29418

Related Content

The Application of FOOM Methodology to IFIP Conference Case Study

Judith Kabelian and Peretz Shoval (2003). *Practicing Software Engineering in the 21st Century* (pp. 82-95).
www.irma-international.org/chapter/application-foom-methodology-ifip-conference/28112

Classification-Based Optimization of Dynamic Dataflow Programs

Hervé Yviquel, Emmanuel Casseau, Matthieu Wipliez, Jérôme Gorin and Mickaël Raulet (2014). *Advancing Embedded Systems and Real-Time Communications with Emerging Technologies* (pp. 282-301).
www.irma-international.org/chapter/classification-based-optimization-of-dynamic-dataflow-programs/108448

Benefits of CMM and CMMI-Based Software Process Improvement

Maged Abdullah, Rodina Ahmad, Lee Sai Peck, Zarinah Mohd Kasirun and Fahad Alshammari (2014). *Software Design and Development: Concepts, Methodologies, Tools, and Applications* (pp. 1385-1400).
www.irma-international.org/chapter/benefits-cmm-cmmi-based-software/77762

Multilevel Clustering of Induction Rules: Application on Scalable Cognitive Agent

Amine Chemchem, Habiba Drias and Youcef Djenouri (2014). *International Journal of Systems and Service-Oriented Engineering* (pp. 1-25).
www.irma-international.org/article/multilevel-clustering-of-induction-rules/117766

Approximate Algorithm for Solving the General Problem of Scheduling Theory With High Accuracy

Vardan Mkrtchian and Safwan Al Salameh (2019). *International Journal of Software Innovation* (pp. 71-85).
www.irma-international.org/article/approximate-algorithm-for-solving-the-general-problem-of-scheduling-theory-with-high-accuracy/236207