

Chapter 4

Creating Open Source Lecture Materials:

A Guide to Trends, Technologies, and Approaches in the Information Sciences

William H. Hsu
Kansas State University, USA

ABSTRACT

This chapter surveys recent and continuing trends in software tools for preparation of open courseware, in particular audiovisual lecture materials, documentaries and tutorials, and derivative materials. It begins by presenting a catalog of tools ranging from open source wikis and custom content management systems to desktop video production. Next, it reviews techniques for preparation of lecture materials consisting of five specific learning technologies: animation of concepts and problem solutions; explanation of code; video walkthroughs of system documentation; software demonstrations; and creation of materials for instructor preparation and technology transfer. Accompanying the description of each technology and the review of its state of practice is a discussion of the goals and assessment criteria for deployed courseware that uses those tools and techniques. Holistic uses of these technologies are then analyzed via case studies in three domains: artificial intelligence, computer graphics, and enterprise information systems. An exploration of technology transfer to college and university-level instructors in the information sciences then follows. Finally, effective practices for encouraging adoption and dissemination of lecture materials are then surveyed, starting with comprehensive, well-established open courseware projects that adapt pre-existing content and continuing through recent large-scale online courses aimed at audiences of tens to hundreds of thousands.

1. TRENDS IN OPEN COURSEWARE FOR INFORMATION SCIENCES

1.1 Tools

This section provides a brief history of open educational resources (OER) for the information sciences, followed by a taxonomic survey of OER development tools.

1.1.1 Brief History

Open educational resources (OER) for the information sciences date back to the early decades of the field, beginning with the development of *PLATO* (*Programmed Logic for Automated Teaching Operations*), the first computer-assisted instruction (CAI) system, at the University of Illinois. (Van Meer, 2003; PLATO History Foundation, 2011)

DOI: 10.4018/978-1-4666-7363-2.ch004

The first version of *PLATO*, implemented on the ILLIAC I circa 1960, included what is now termed lessonware and was funded jointly by the U.S. Army, Navy, and Air Force. Meanwhile, by the late 1960s, video lecture consortia such as the Stanford Honors Co-op were delivering proprietary closed-circuit television content to corporate sponsors (House & Price, 2009). The 1970s brought a wave of intelligent tutoring systems (Carbonell, 1970; Sleeman & Brown, 1982; Iiyoshi & Kumar, 2008). By the 1980s, cable-access distance learning and extension courseware had begun to be distributed using precursors of open source licenses, culminating in the founding of the Free Software Foundation in 1985 and the first releases of the Berkeley Standard Distribution (BSD) License (1988), GNU General Public License (1989), Open Content License (1998), and Creative Commons License (2001). (Free Software Foundation, 2012) Abelson, a founder of the Massachusetts Institute of Technology OpenCourseWare (MIT OCW) initiative (Abelson, The Creation of OpenCourseWare at MIT, 2007; Attwood, 2009) and founding member of Creative Commons (Creative Commons Corporation, 2011), had been distributing *Structure and Interpretation of Computer Programs*, a leading introductory textbook in computer science, online. With the advent of MIT OCW, video lectures prepared for the MIT/Hewlett-Packard consortium (House & Price, 2009) as early as 1986 were made available (Abelson, 2005).

1.1.2 Technologies for Producing Open Source Software

When discussing “open source tools,” professionals and students in science, technology, engineering, and mathematics (STEM) fields often refer only to *open source software* (DiBona, Ockman, & Stone, 1999; Raymond, 1999; Open Source Initiative, 2006)¹ rather than the more general concept of *open content* (Wiley, 2011) as coined by David Wiley in 1998 (Wikipedia, 2012). The

means of production are diverse for both forms of creative work, with free redistribution and access being the unifying characteristic. For open source software, however, the chief production technologies are software engineering tools: integrated development environments; content management systems; and version control systems, also known as “source code control systems.”

Integrated development environments (IDEs) are suites of development applications consisting of source code editors, compilers (and/or interpreters), and build/execution controls, plus optional components such as build utilities, interfaces to version control systems, visual code layout and refactoring tools, and interactive code inspection and debugging tools. (D’Anjou, Fairbrother, Kehn, Kellerman, & McCarthy, 2005; Nourie, 2005) They range from the proprietary (*e.g.*, Microsoft *Visual Studio* and Apple *Xcode*) to open source (*e.g.*, *Eclipse* and Oracle *NetBeans*). The range of available IDEs depends foremost on the programming languages to be supported and secondarily on the development platform, comprising the computer architecture, operating system, and compilers or interpreters. For ease of use, efficiency, and portability, many open source developers use simple editors, version control, and compilation tools to augment or replace full-featured IDEs when their full power is not required.

A content management system (CMS) is a collection of procedures (implemented manually or computationally) for organizing and carrying out work flow in a collaborative environment. (Depow, 2003; Mauthe & Thomas, 2004) Specific CMSes may be implemented as web services or using other software as a service (SaaS) architectures, or as standalone applications such as most wikis. Both types of CMSes occur in both proprietary and open source varieties. Schaffert *et al.* (2006) describe *semantic wikis*, which capture information on the deep relational structure between pages and provide this information to agents and services beyond mere linking. These are referred to as *semantic wikis*, after the Semantic Web, or Web 3.0.

25 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/creating-open-source-lecture-materials/121833

Related Content

The Role of Teacher Leadership for Promoting Professional Development Practices

Patricia Dickenson and Judith L. Montgomery (2016). *Innovative Professional Development Methods and Strategies for STEM Education* (pp. 91-114).

www.irma-international.org/chapter/the-role-of-teacher-leadership-for-promoting-professional-development-practices/139653

Engineering and Art: Putting the EA in STEAM

Sara B. Smith (2020). *Cases on Models and Methods for STEAM Education* (pp. 258-273).

www.irma-international.org/chapter/engineering-and-art/237799

Scaffolding Hypothesis Formation and Testing During Simulation Coding

Lucas Vasconcelos (2023). *Technology Integration and Transformation in STEM Classrooms* (pp. 19-39).

www.irma-international.org/chapter/scaffolding-hypothesis-formation-and-testing-during-simulation-coding/317526

The Power of Computational Modeling and Simulation for Learning STEM Content in Middle and High Schools

Mahnaz Moallem, Shelby P. Morge, Sridhar Narayan and Gene A. Tagliarini (2018). *K-12 STEM Education: Breakthroughs in Research and Practice* (pp. 916-950).

www.irma-international.org/chapter/the-power-of-computational-modeling-and-simulation-for-learning-stem-content-in-middle-and-high-schools/190136

Fostering Inclusivity: Nurturing Diversity Within Elementary STEM Teacher Preparation Programs

Surjit Singha (2024). *Using STEM-Focused Teacher Preparation Programs to Reimagine Elementary Education* (pp. 286-311).

www.irma-international.org/chapter/fostering-inclusivity/338418