

Infrastructures for Future Educational Software

Michael Bieber

bieber@njit.edu

<http://www-ec.njit.edu/~bieber>

Abstract

Educational content developers need structured tools, methodologies and guidelines specific to the educational domain for developing effective course-related software. We provide the CHEF framework for asking what we believe will prove the correct questions and for thinking effectively about the on-line software environment. We strongly encourage educational researchers and developers to develop effective guidelines, methodologies and tools for educational software environments. CHEF should prove a useful framework for identifying and thinking about these.

Motivation

Many educators, having successfully put up a home page on the World Wide Web, are now being asked to put up their entire course content and administrative features on the Web or a CD. At most, they have some technical assistance available, concerning HTML format or conversion, or their particular multimedia authoring tool. But educational content developers need structured tools, methodologies and guidelines specific to the educational domain for developing effective course-related software. Just as many of us at the university level have received almost no training on how to teach, few of us have any clue as to how to structure and author on-line course content effectively; how to customize and deploy existing on-line course materials effectively; how to use on-line courseware effectively as a teaching or learning tool; or how to use on-line courseware with regards to student assignments and other team projects. Similarly, we have received no training on how to use multimedia effectively within a teaching, learning or business environment, yet we often are expected to take full advantage of graphics and animation. Students too, have no guidelines on how to best use on-line tools, and instructors have no training in teaching students effective strategies for learning in this environment. How will a student learn to take notes within on-line software environments (both for him- or herself or as a study group member)? How should one effectively annotate or overlay a personal semantic link organization in order to help classify and synthesize important concepts across a semester or a curriculum of multiple courses? What do the developers of on-line course environments need to know about authoring, teaching, learning and collaboration in order to produce effective, high-level authoring and reading tools? As educational software becomes increasingly pervasive in distance learning programs, as well as in supplementing traditional classrooms and extra-classroom activities, the workload and frustration for content developers and users will grow.

This paper does not provide any answers to these questions. Instead we provide a framework for asking what we believe will prove the correct questions and for thinking effectively about the on-line software environment. The framework will help the builders of educational software tools to incorporate the correct set of features. It will guide research into teaching and learning methodologies for on-line software environments. It will assist researchers in developing guidelines for content authors, instructors and students within these environments. We have grounded the framework in educational theory to ensure that it successfully underlies all these endeavors.

Processes in Using Educational Software

CHEF will prove useful only if it can represent the many educational processes that a rich and comprehensive learning environment could facilitate. In this section we put forth an important subset of these processes. No comprehensive survey or theory exists putting forth a complete set, but we believe those we include cover a broad enough range of these processes to demonstrate CHEF's breadth. Ideally CHEF will help researchers develop a methodology (separate or comprehensive) and a set of guidelines for each process. We cannot predict whether most educational software of the future will support most or all of these processes, but we believe that it should.

- Content Authoring Processes

- authoring content
- using multimedia when authoring content

- Instructor Processes

- designing a course around an educational software environment
- customizing authored content for a particular class
- designing homework assignments and group projects within an educational software environment
- evaluating homework assignments and group projects within an educational software environment
- teaching a course designed around an educational software environment
- team teaching (including teams of one instructor, multiple recitation leaders and graders)
- course administration
- evaluating educational software environments and authored course content
- integrating different course software within a single course

- Student Processes

- how students use content and structuring tools to understand and study (including annotation and synthesis structures)
- doing homework assignments and group projects within an educational software environment
- using multimedia in group projects and reports

CHEF and Educational Theory

Educational theory should underlie the methodologies and guidelines supporting learning and teaching within educational software. The authoring and reading tools should incorporate features that best support educational theory. In this section we review the major educational theories, show how they support collaboration and hypermedia concepts, and begin a discussion of how they can guide development through CHEF.

CHEF is ideally suited to support both individual and collaborative learning because its conceptual tenants (collaboration, hypertext and multimedia) provide a sound platform for implementing theoretical approaches to constructivism. These include: anchored instruction [Ku95, MSS95]; cognitive flexibility theory [SFJ91]; student centered teaching [MD95]; scaffolded knowledge integration [Li95]; and active learning [RPL95].

CHEF and Technologies

CHEF also can assist in determining a list of requirements for, evaluating and choosing educational technology; as well as guiding the development of future educational technology. Content authors, instructors and academic departments, and perhaps even students will be making these decisions. Educational technology includes hardware and network communications, authoring tools, authored content, and the teaching and learning tools for working within authored content.

- determining a list of requirements for one's specific course or domain (both for content authors and for instructors)
- evaluating existing educational technology
- choosing existing educational technology
- evaluating the design of new educational technology
- guiding the development of future educational technology

Conclusion

Clearly our work on CHEF is at an early stage. Even in its current state we have found CHEF a very useful framework for getting instructors to unite behind the realization that today's generation of educational software is quite simple compared to what it should support. We also have found it useful in getting both instructors and content authors to think about collaboration, relationship management, sophisticated navigation and various forms of multimedia explicitly and realistically. Much of our future work will go into ways to best use the framework, and ensuring that it is comprehensive enough and correct.

In particular we intend to:

- identify all major educational processes and determine the types of guidelines, methodologies and software support each stage of each process requires to be effective

- eventually come up with a set of requirements for educational software in supporting complementary sets of educational processes.

- validate CHEF to ensure that it is comprehensive (how?)

- validate CHEF to ensure that it is correct and usable (how?)

- guidelines for determining a list of requirements for, and for evaluating and choosing educational technology

- a view of how CHEF can guide the development of future educational technology

- determine whether CHEF applies as is to vocational and corporate training environments, or whether it needs to be modified for them (and how)

We strongly encourage educational researchers and developers to develop effective guidelines, methodologies and tools for educational software environments. CHEF should provide a useful framework for identifying and thinking about these. We also call on all educators to demand that effective educational software is developed. As educational researchers and instructors, we face a very exciting but traumatic future as we adjust to the rapid changes that technology brings to our classrooms. Together we can guide software development so it truly and easily supports effective teaching and learning.

Acknowledgments

This research generously has been supported by the NASA JOVE faculty fellowship program, by the New Jersey Center for Multimedia Research, by the New Jersey Commission on Science and Technology, by the New Jersey Institute of Technology (NJIT) under grant #991967 and #42140, by the New Jersey Department of Transportation, by the National Center for Transportation and Industrial Productivity at NJIT, the National Science Foundation (NSF—IRI—9015236 and NSF—IRI—9408805), and by a grant from the AT&T Foundation.