FORMAL DESCRIPTION OF COMPUTER COURSEWARE

Jaak Henno[1], Hannu Jaakkola[2], Jari Multisilta[2]

ABSTRACT

There is great need for courseware description language, which would allow describing computer courseware in an implementation-independent manner. This would allow to study courseware design before implementation, systematically introduce design methods based on learning/instruction theory, separate knowledge entities from instruction/interaction framework and thus enable to use the same knowledge entities with different modes learning/instruction (presentation, assessment) and enable to re-use both the knowledge objects with different learning/instruction frameworks and instruction shells with different subject domain knowledge.

INTRODUCTION

The use of computer courseware, Computer-Based Training (CBT), Computer-Assisted Learning (CAL) etc is rapidly increasing. In 1997, CBT accounted for 15% of total training effort of American business, government, and education institutions (from all questioned organisations in USA)[i] and CBT market was witnessing a 38% annual growth; already in the next year[ii], average percentage of training effort delivered via CD-ROM or LAN-based CBT was estimated 23% and 80% of respondents said this type of training delivery was growing in their organisations. On university level[iii], percentage of classroom and online delivery is expected to be equal by year 2000.

Thus more and more trainers, teachers, university professors get involved in creating computer courseware. Quite often courseware created in one university or for training needs of enterprise remains in this university or enterprise, i.e. the distribution level of courseware is very low. Courseware designers learn best from existing examples, but if those examples are not distributed, they also do not help to raise the general design level. Creation of computer courseware is still very implementation-dependent and very labour intensive: "... computer-based multimedia interactive instruction, is too labour intensive usually requiring more than 300 hours of development for a single hour of instruction"[iv] One of reasons for is uniqueness of every implementation, currently re-use of successful designs is at very low level.

Evaluation of courseware only after implementation has also other weaknesses. Even when courseware is reviewed or shown on exhibitions, quite often "cool" multimedia features and visual effects are considered far more than systematic, based on learning/instruction theory design.

To discuss design virtues and to help distributing and re-use of successful designs there should be methods for formal description of courseware, its content and user interactions. Such descriptions allow to pre-program courseware shells and thus allows subject matter

[1] Tallinn Technical University, Estonia.

[2] Pori School of Technology and Economics, Tampere University of Technology, Pohjoisranta 11, PO Box 300, FIN-28101 Pori, FINLAND

experts to design effective computer-based, interactive multimedia without requiring them to have extensive training in instructional design or authoring systems.

Currently, most often the courseware design is described using text-based storyboards. There are several problems with such informal descriptions. They are

- often implementation-dependent, e.g. in the above description: "window", "viewer", "highlighted" (some authoring/implementation environment may not allow these constructs); therefore implementation in different computing/operating environments becomes very difficult;
- leave many things ambiguous or are too specific - what is "panel to the left" and why the "left" is better than "right", "below" or "upper"?
- are (in most cases) topic-specific, i.e. can not be re-used for teaching/explaining other similar tasks;
- the purely descriptive (what happens on screen at run-time) representation does not allow distinction between instruction/teaching style (i.e. user interactions) and exposed data (knowledge objects), what makes it nearly impossible to check soundness of design according some learning/instruction theory; this also often results in cognitively difficult to understand screen and interaction designs and makes very difficult collaborative authoring, which is very desirable because of labour-intensity of courseware design and implementation).

Thus storyboard-style informal and implementation-dependent descriptions do not allow evaluate courseware design before implementation, compare different design decisions and re-use successive designs. It is very difficult to base design on some established learning/instruction theory, since it is very difficult to understand in the above description, which decisions were made because of theory predictions and which because designer thought this to be "cool". It does not provide guidance for interaction design and does not specify an adequate syntax for knowledge representation and is therefore not adequate for computer-based interactive instructional multimedia development. With the storyboard method, every courseware application can basically be adequately evaluated only after it has been implemented. But after implementation, implementation features (visual appeal, "cool" multimedia tricks and other implementation features) often "screen off" theory virtues, hinder evaluation of instructional value of design. The situation is somewhat similar to the famous "no significant difference" discussion about value of different media in instruction[v]. There is abundance of learning theories, e.g. [vi] provides overview of 50 learning theories, but it is very difficult to evaluate value of any theory using some already implemented courseware. Hopefully virtues of theory, level of application of a theory etc can be better evaluated using a formal description of a courseware.

PRINCIPLES OF COURSEWARE DESCRIPTION LANGUAGE

The proposed formalism for courseware description should allow to describe courseware on implementation-independent level, so that the visual design, "cool" multimedia/technical tricks can not influence evaluation of the design, only some implementation guidelines are provided. It should be possible to consider basic features of design: subject domain knowledge representation and quality of instructional interactions (instructional design) before any implementation; it should also be possible to re-use successive elements of design or re-implement courseware (e.g. when there appear new multimedia technologies).

The main target of this proposed method is description of possible interactions. This allows implement instructional methods, proposed by learning theories[vii][viii]

The description should make it possible to describe and classify semantically different interactions. Interactivity is one of the worst understood features of any software[ix],[x] and often causes conceptual difficulties in understanding intended functionality of courseware. The formalism should allow to describe interaction structure of a courseware application and to classify different types of interactions - navigation, providing further (deeper) explanations, providing parallel (using a different media or language level) explanations, implementing a prescriptions of a learning/instruction theory for different instruction modes (presentation, exploration, practice) etc.

Interactive multimedia courseware has complicated structure. Courseware is a special case of interactive hypermedia, and all interactive hypermedia applications have at least these three "inner" structures[xi], [xii]:

- conceptual structure of the subject matter
- material organisation (presentation) structure; in courseware the presentation should correspond to Instruction (learning) Theory.
- appearance - user interface (visual design), "cool" multimedia tricks.

The proposed formalism is based this observation and targeted to describe the first two levels; only some guidelines are given for implementation (implementation level also changes most rapidly). Conceptual structure of subject matter corresponds to database of so-called knowledge objects and the presentation structure should be based on some learning/instructional theory. Unfortunately, there are quite many theories (aimed for different types of learning/instruction); of all learning/instruction theories, the presented method seems to be most close to principles of the Instructional Design Theory of M.D. Merrill[iv]:

- "subject matter is data and as data it can be uncoupled from the instructional strategy used to teach this subject matter"[xiii]
- "there is a number of different kinds of instructional transactions and the same data (knowledge objects) can be used with different instructional transactions, and the same instructional transactions can be used with different data (knowledge objects)"

Thus the main difference of courseware (compared to arbitrary hypertext) is that there are (at least) two type of links (user interactions).

References:

[i] Floyd Kemske. "1997 Computer-Based Training Report. A Comprehensive Study of CBT and Multimedia as Instructional Delivery Systems". SB Communications, pp 1-151

[ii] The 1999 CBT Report. CBT Solutions Magazine, vol. 2, N 11, 1998, http://www.cbtsolutions.com

[iii] Highways to 21th Century Training. CBT Solutions, May/June 1998

[iv] M. David Merrill and ID2 Research Group. ID Expert™ : A Second Generation Instructional Development System. Instructional Science, 1998, 26(3-4), 234-262, http://www.coe.usu.edu/it/id2/idexpert.htm

[v] Thomas L. Russell. The No Significant Difference phenomen. Instructional Telecommunications, North Carolina State University 1999; overview: http://tenb.mta.ca/phenom/

[vi] The Theories. http://www.gwu.edu/~tip/theories.html

[vii] Martin Ryder, Instructional Design Models. http://www.cudenver.edu/~mryder/itc/idmodels.html

[viii] The Theories. http://www.gwu.edu/~tip/theories.html

[ix] Rod Sims. Interactivity: A Forgotten Art? Faculty of Education University of Technology, Sydney http://www.education.uts.edu.au/education/staff/rod_sims/itforum.html

[x] Stephen W. Draper. Content and Interactivity. http://www.psy.gla.ac.uk/~steve/interactivity.html

[xi] J. Henno. Design of Intelligent learning Support Environments. Proc Nordic Conference on Computer Aided Higher Education, Helsinki 1991, 119-137

[xii] J. Henno. Using SA methods and multi-level grammars to design hypertext materials. Ohsuga S., Kangassalo H., Jaakkola H., Hori K., Yonesaki N. (eds), Information modelling and Knowledge Bases III: Foundations, Theory and Applications, IOS Press, Amsterdam 1992, 98-113

[xiii] M. David Merrill et al, Instructional Transaction Theory: Instructional Design based on Knowledge Objects. Educational Technology, 1996, 36(3), 30-37