# DIDACTICS AND IMPLEMENTATION OF CONTENT FOR NON-SUPERVISED DISTANCE LEARNING

Irina Bodrow, Stefan Wolf, Wladimir Bodrow
University of Applied Sciences Berlin

## ABSTRACT

With this paper we aim at introducing a special e-Learning solution which provides for non-supervised distance education in programming. We consider the specifics of knowledge to be communicated to students when learning programming languages and technology. Based on this several decisions in respect to knowledge representation have been made namely: the metaphor of knowledge representation, the design of man-machine-interface as well as the interaction scenario.

## INTRODUCTION

Content production makes up the fundament of e-Learning. This is where the usage of computer supported teaching begins and ends. During the last twenty years various authors and companies have dealt with this topic. As a result of these dealings a number of concepts and solutions for the development of teaching units have come up, which can be used with the help of already established information and communication technologies. Authoring systems, CBT- and WBT-Shells integrated into the one or the other e-Learning environment belong to these technologies. WebCT, Blackboard, LUVIT, TopClass, DLS, ILIAS [LI - L6] etc. represent some examples of such environments. The multitude of already existing and practiced solutions forms a solid foundation for the formulation of an e-Learning ontology.



## TOWARDS AN E-LEARNING ONTOLOGY

Ontology means: generalized formal definition of a single object or term in a particular knowledge area. An ontology therefore embodies all basic functions and features of a subject in a certain context. The contemporary discussion about ontology in informatics as well as other sciences [3, L7 - L8] can give helpful impulses for formulating an e-Learning ontology, too. Such an ontology could mark the groundwork for the design of future e-Learning solutions across geographical and cultural borders. Therefore it requires a high level of generalization. But this general definition is hard to apply in a concrete project, where details play a crucial role.

In this paper we do not analyze abstract or theoretically possible learning contents, knowledge transfer metaphors, interface concepts and user models. Rather we discuss the concrete applicability of these components in a specific environment. Thus the decisions made have only limited relevance for other contexts, where knowledge transfer with the help of e-Learning is practiced.

A corresponding analysis of the results presented is nevertheless worthwhile, since experts can find certain generally valid principles in every special solution toward an e-Learning ontology.
In detail we will deal with the development of an e-Learning solution for primary and secondary education in programming [L9]. Thereby we will discuss the learning of programming techniques and programming languages with the special emphasis on didactic aspects. In the development of our metaphor we mainly concentrate on the organization of learning without a permanent counseling by a teacher from the very first phase of knowledge presentation on [1, 6, 8].

The following aspects influence the process of education (face-to-face or frontal-teaching and e-learning too) severely [1, 4, 5, 7]:

- the specifics of knowledge and content
- the learning goals and the derived form of knowledge
- the different ways of knowledge representation according to the media resources available

- the adequate metaphor of the learning process
- different possibilities of interaction and controlling between students and teacher (or learning program)

For e-learning the design of the interface and interaction as much as knowledge representation is subject to the chosen medium.

In our particular example the knowledge domain has principally two sides, which a teacher has to communicate to the student. On one hand there is the source code, which is central for learning a programming language and programming in general. On the other hand the teacher or his electronic artefact employs explanations in natural or symbolic language. The natural language has a higher priority, since the student does not have to acquire it anymore. The symbolic explanation or other medial instruments have lower priority, but can be used, too. That is why the screen is generally used for representing the source code and explanations. The other media will be placed additionally. This strategy is stringent and provides for a linkage between source code and the student's new learned understanding from the explanations given. In the conventional learning process commentaries are mainly given in spoken language.

Based on this duality we decided to split the interface into two sides. The left half represents the source code and the right half the explanations in natural language. The source code in text format along with the explanations likewise in text format poses the basis for the book metaphor.
Additionally graphics and animations are considered as a complement to the explanations in the text and appear in a separate window.

Because the explanations are often substantial (a lot longer than the source code) they are also shown in two versions: long and short. This way different user models can be served: beginners and advanced. The explanations for the advanced appear first on the screen and can be replenished for beginners if needed. This working method allows a compact and economic utilization of the screen (one line explanation for one line source code). With the integrated link >More< the learner can access additional explanation to the source code.



As mentioned before there are two user models integrated into this teaching process resulting in increasing acceptance and time saving for the learner. After working through some examples the acquired knowledge will form the basis for the following steps. Consequently the short version of the explanation will be frequented higher. But the order of the learning objects is not strictly prescribed, thus it is principally not predictable, how much knowledge has been gained after completing each learning object. This form of explanation again supports the generally given flexibility of e-Learning especially regarding the studying tempo.

## SCREEN DESIGN

Since many examples contain a lot more than one screen page of source code, only one part of it can be represented at once. The subdivision of the source code into single paragraphs is not preferable because with it the interrelation of the source code is lost. The installation of a step-by-step likewise a page-by-page navigation interferes with the learning process and leads to a loss of continuity in the source code on the other hand. We implemented a slider as a dividing line between the two screen halves (and texts). It allows for a user based simultaneous movement of both parts of the screen.

The student can navigate quickly to the desired point in the source code and start the knowledge acquisition process from there. Above that the already mentioned button for additional information >More< can be activated. This will open an extra window which is not split anymore.

Our chosen structure in table form promotes an efficient knowledge adoption because it's consistent organization in columns. The advantages of this knowledge representation become noticeable especially when using electronic media. In Fig. 1 we present our development template as it will appear on the screen.

**Fig. 1** Template for implementation of Learning Objects

An example of the >***More***< window is presented in Fig. 2 and an animation example is shown in Fig. 3 respectively.



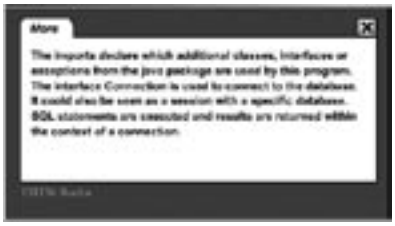


**Fig. 2** Example of additional explanation in the >***More***< window.

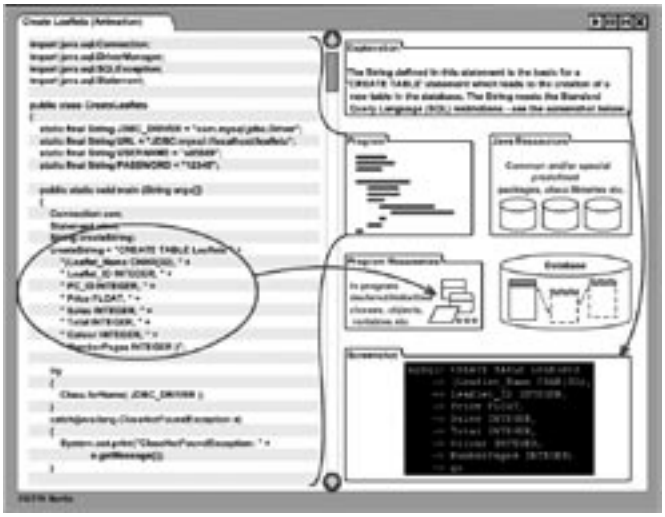

**Fig. 3** Example of the implementation of animations.

In order to form the solution as standardized as possible there will be only few navigation buttons. This provides for a mere concentration on the process of passing on knowledge undisturbed by navigation- and interaction elements.

## OUTLOOK

For students of informatics the metaphor described can be used to create e-Learning modules for teaching the functionality and operation of different software tools or operating systems. Instead of the source code in our example there would be screen shots and animations. Also for the field of programming: e.g. programming graphics, those graphical elements are essential parts for imparting knowledge. They have to appear together with the source code and the explanations on the screen. By this means the learner will see the results of the application of each function in the programming language right away. In conclusion after a suitable analysis our metaphor will be applicable in other domains to improve the comprehension of knowledge within e-learning.

## REFERENCES


[1] Arnold, P. (2001), Didaktik und Methodik Telematischen Lehrens und Lernens, Münster: Waxmann.

[2] Bruns, B.; Gajewski, P. (1999), Multimediales Lernen im Netz, Berlin: Springer.

[3] Fensel, D. (2004), Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, Berlin: Springer.





[4] Harrison, N. (1999), How to Design Self-Directed and Distance Learning, New York: McGraw Hill.

[5] Horton, W. (2000), Designing Web-Based Training, New York: John Wiley.

[6] Kerres, M. (2001), Multimediale und Telemediale Lernumgebungen, München: Oldenbourg.

[7] Lankau, R. (2000), Webdesign und Publishing, München: Hanse.

[8] Schulmeister, R. (1997), Grundlagen Hypermedialer Lernsysteme, München: Oldenbourg.


### WEB-Resources


[L1] http://www.webct.com

[L2] http://www.blackboard.com

[L3] http://www.luvit.com

[L4] http://www.wbtsystems.com

[L5] http://www.ets-online.de

[L6] http://ilias.de

[L7] http://encyclopedia.laborlawtalk.com/Ontology

[L8] http://en.wikipedia.org/wiki/Ontology

[L9] http://www.codewitz.com