

The Essen Learning Model – a Multi-Level Development Model

Jan M. Pawlowski
Information Systems for Production and Operations Management
University of Essen, Germany
jan.pawlowski@wi-inf.uni-essen.de

Abstract: The Essen Learning Model is a generic development model supporting development processes on three levels: development of curricula, learning sequences, and learning units. We focus on finding an adequate combination of didactical methods and educational technologies. Secondly, we focus on interoperability, reusability, and ease of use, using standards on the modeling level (ARIS), architecture level (LTSA, IMS), and implementation level (XML). The approach will enable educators, project managers, and authors to efficiently develop and implement Computer Supported Learning Environments.

Introduction

The Essen Learning Model is a development model to ensure the overall quality of the development process of learning environments on different levels. New technologies in the educational sector require a well planned and controlled development process. Our experience has shown weaknesses of current development models for the utilization of new technologies in the educational sector.

The use of new technologies is often seen as a universal solution to all problems and challenges in the educational sector. Educational concepts like Virtual University, Multimedia, or Edutainment have gained an increased attention. Despite of extensive research efforts, many immature and unsuitable products and concepts can be found. Quality management is not yet a natural part of educational projects.

The use of development models, originating from the field of software engineering, is a first step towards quality management and assurance. Nevertheless, several restrictions remain: software development models do not include didactical concepts adequately. Specific models for the development of hypermedia systems or CBT-systems require the selection of didactical methods before the start of the project. To overcome those weaknesses of most development models, a multilevel development model was designed, supporting educational processes on three levels: the design and implementation of curricula, learning sequences, and learning units.

I present basic concepts of learning environments and the Learning Technology Systems Architecture which is the base for the development of learning environments. By following this standard architecture, systems implemented using the Essen Learning Model can easily be integrated and coupled with other systems, reducing the development effort significantly. Thereafter, I describe design principles and the levels of the Essen Learning Model together with a sample implementation for the development of a learning sequence on “Simulation”.

Computer Supported Learning Environments

Our development model is based on the Learning Technology Systems Architecture. I will briefly describe this standard and our implementation.

Classification of CSLE

According to (Euler, 1992), Computer Supported Learning Environments might be classified into five classes: *Drill & Practice Systems*, *Intelligent Tutoring Systems*, *Hypermedia Systems*, *Cognitive tools*, and *Simulations/Micro Worlds*. Those criteria only describe trends of developments. They cannot be used for evaluating or selecting a certain learning environment. In contrast, the Essen Learning Model uses a detailed multi-dimensional classification including attributes like learning objectives, strategies, learning contents, temporal aspects, location, interaction, and users.

Architectures of CSLE

Concepts like the Virtual University or distance learning are based on the exchange and combination of courses and learning contents distributed world-wide. Unfortunately, the majority of learning environments and authoring systems do not use standardized methods in order to support exchange of materials or courses. This leads to an enormous number of isolated systems, not utilizing and being utilized by other systems. To overcome

those weaknesses, several standardization projects have been started. The Essen Learning Model uses an implementation of the proposed standard architecture by the Learning Technology Standards Committee of IEEE.

The Learning Technology Systems Architecture (LTSA) is an abstract model describing the architecture of learning environments (Fig. 1). It contains the following components: *Processes* describe the main subsystems of learning processes: Learner, Delivery, Evaluation, System Coach. *Flows* describe in- and outputs of the processes: Behavior, Assessment, Performance, Query Index, Content Index, Locator Index, Learning Content, Multimedia, Learning Style. Finally *Stores* store relevant information: Records Database, Knowledge Library.

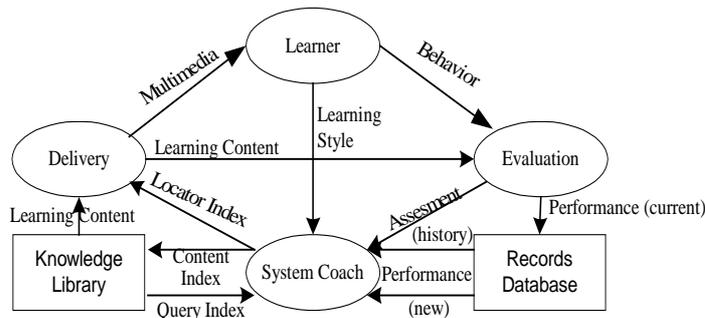


Fig. 1: Learning Technology Systems Architecture

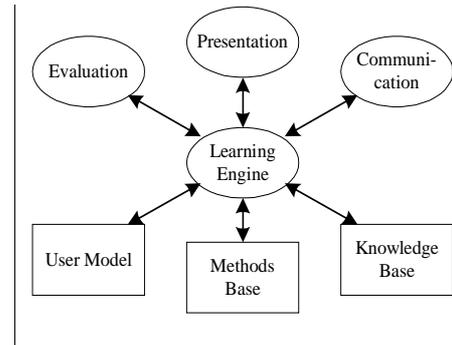


Fig. 2: Architecture of ELM

Furthermore, the LTSA defines formats for the communication and interaction of systems. The implementation of ELM (Fig. 2) uses the standards of the LTSA specification in order to combine and integrate different learning environments.

The main element of a CSLE is a *Learning Engine* for the steering, control, and co-ordination of the components. The *Knowledge Base* contains learning objectives and contents used by the CSLE. The *Methods Base* of a Learning Environment contains different didactical methods and concepts, supporting authors and educators. A *User Model* is designed in the development phase. This model contains attributes, characteristics, and knowledge of a user. In collaborative learning environments, additional information about the group and the group process is modeled (Kinshuk, Patel, 1997). The *Presentation Component* allows the generation and presentation of learning contents in different ways. The *Communication Component* determines the level of interactivity of a Learning Environment. It contains methods for communication, co-operation, and its technical realization. Communication between learner, teacher, and system is to be co-ordinated in different forms (Heeren, 1996). The *Evaluation Component* determines the performance of the learner, providing tests or ratings.

Development Models in Software Engineering

The main function of a software engineering model is to establish the phases in which major tasks are performed within a project, and to establish the transition criteria for proceeding from one task to the next. A software development process model supports the implementation of software projects together with the planning, scheduling, and controlling of time and budget constraints. According to (Balzert, 1998), a development model specifies the following issues: *Processes* (scope, activities, work breakdown structure, schedule, prioritization), *Organization* (competencies, responsibilities, roles, budget), *General Issues* (standards, guidelines, methods, techniques, tools, reporting). There are various software development models, following different strategies. For many years the *Waterfall model* (Royce, 1970) was the most popular process model. It is a highly structured approach, breaking down the development process into multiple phases. Each phase creates a result which triggers the next phase after approval. If the result is rejected, the performed phase is repeated. Backtracking is limited to successive stages due to complexity reasons and to minimize expensive re-work. In general, the model is transparent and easy to understand, however, there are essential disadvantages. Requirement changes during the development are hard to handle because of the limited backtracking mechanism. Involvement of the users is only provided during the requirement specification in a very early phase. Especially for learning environments, the lack of motivational factors leads to acceptance problems. Due to the sequential approach, concurrent activities are not supported, hence extending the duration of a project.

Prototyping (Balzert, 1998) is a popular method to provide „user friendly“ software. It is based on the development of executable models and patterns, defining software characteristics and requirements successively during the development process. These basic steps are repeated until all requirements are well defined. The

prototype itself is discarded and the real system is build, based on the prototype. In comparison to the Waterfall model, the development effort is higher due to the additional development of prototypes. The main advantage of Prototyping is early user involvement. Because of this important advantage, the Essen Learning Model incorporates prototypes into the development process.

The *Spiral model* (Boehm, 1986) combines positive aspects of the Waterfall model and the Prototyping approach. It can be used for rapid development of incremental versions. The definition of the software product and software process is elaborated in a series of repetitive steps. These steps are represented as a spiral progression through a project. Each cycle of the spiral elaborates product and process objectives and constraints, alternative product and process solutions, identifying and resolving major risks. The disadvantages are concerns about the high level of control and negotiation difficulties caused by the amount of control. High expertise in risk management is required. Advantages are the match of changing requirements in the whole development process, and the consideration of risks at all stages of the development. Especially for learning environments these characteristics are significant: therefore this approach was chosen for the Essen Learning Model.

Development Models for CSLE

There are various models specifically designed for the development of CSLE. Major approaches are: models based on *Didactical Frameworks*, e.g., on the „Berliner Modell“ (Heimann, Schulz, Otto, 1965). They do not consider to use technical innovations. Therefore, they can be used for didactical planing but not be applied to new methods and technologies. *Specific CSLE* models for certain instances, e.g., CBT development (Steppi, 1997) or *Hypermedia Development Model* (Garzotto, Paolini, Schwabe, 1993). The weaknesses of these models are obvious because didactical decisions must be made before the selection of the CSLE. *Instructional Design* models (Issing, 1997) describe the structured development process of the educational systems, courses, and teaching products. They fulfill the requirements much better than the above described models, however, they support authors and teachers only insufficiently because the conceptual phase lacks flexibility.

As a conclusion, all the above mentioned models do not consider technological and didactical aspects sufficiently. Either they lack flexibility, or developers and users are highly restricted in their creativity. The Essen Learning Model is an approach which integrates and improves positive aspects of different development models.

The Essen Learning Model

The Essen Learning Model was developed at the University of Essen. It is currently being evaluated to be used in several departments of our university. I will describe the design principles and the general flow of the approach.

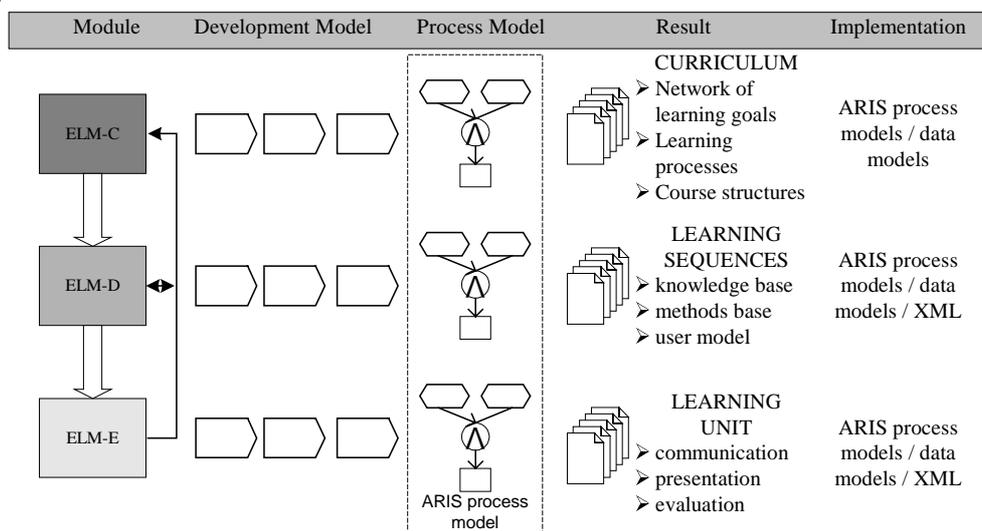


Fig. 3: The Essen Learning Model

The Essen Learning Model (Fig. 3) is a multi-level development model. The complex development process is divided into modules supporting both, the development process and the use of the system on different levels. The support of curriculum development and design (C-level), planning and implementation of learning sequences (D-level), and the design and implementation of learning units (E-level).

We distinguish three abstraction levels: The *development model* provides generic knowledge for a variety of contexts. This generic model is customized depending on the user's needs and preferences, and transformed into a *specific process model* for each project. The process model provides a framework for educational technology projects, specifying the timeline of the project, involved participants, and information technologies. It supports conceptual and implementation decisions. Furthermore, it can be used to integrate learning processes and business processes to close the traditional gap between education and every-day-work (Adelsberger, Körner, Pawlowski, 1998). The third level is the *result* of the development process, resulting in certain implementations on each level. The result of ELM-C is a detailed network of learning objectives and goals. The structure and relations of learning sequences (courses) are determined. Based on these results, learning sequences are developed in ELM-D. The focus of this phase is to find an adequate didactical method including the right technology depending on learning objectives and the user group. Single learning units are designed and implemented in ELM-E.

ELM follows the approach of the *spiral model*. By using this approach, changes concerning the needs, requirements, and wishes of the project participants can be integrated dynamically. Furthermore the user is involved in the development process, evaluating prototypes and contributing actively to the development. By using this approach we ensure the quality of the product from the beginning to the final implementation. One of the most important goals is the interoperability with other systems. The architecture of the learning environment is consistent to the Learning Technology Systems Architecture of the Learning Technology Standards Committee of IEEE and to the specifications of the Instructional Management Systems Project (EDUCAUSE, 1999). Furthermore, we used standard tools and formats for both, the development and the implementation. The learning processes are modeled in ARIS, a process modeling tool (Scheer, 1998). Our approach is not limited to this tool, process definitions in IDEF (Adelsberger, Körner, 1995), UML (Booch, Jacobson, Rumbaugh, 1998), or object oriented modeling languages can be integrated easily. We have implemented a web based learning environment in XML. XML (Extensible Markup Language) is a text-based language, describing, exchanging, and presenting structured data objects (e.g., data, applications) (W3C, 1999). Using this approach, web based learning environments can be combined and integrated with other resources. In the following I present the modules of ELM in detail and a successful implementation: a learning environment for students of Business Computing at the University of Essen. A learning sequence on "Simulation" was implemented and is being evaluated.

ELM-C – Curriculum Development

The ELM-C module starts with an analysis of the context. Educational and structural aspects are investigated in this phase. First of all, overall policies and strategies are investigated. This is necessary to determine the role of education for a specific company or institution. Then existing educational concepts and the corresponding business processes are analyzed. To acquire knowledge about the future users, the organizational structure is investigated, determining roles, stereotypes, and responsibilities of the users. As another important aspect, the structure of existing technologies has to be analyzed. This leads to conclusions about which resources can be utilized for learning processes. Furthermore, potential changes of the existing structure are identified.

After this first phase, the generic development model is customized, resulting in a specific process model for the development process. The actual project planning, including team building, scheduling, and guidelines are determined. Furthermore, the main processes for the design, implementation, and evaluation of the curriculum (Fig. 4) are included into the process model.

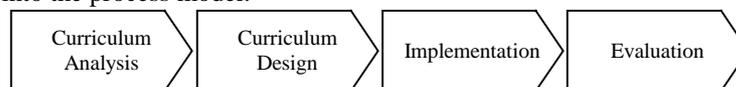


Fig. 4: Main processes of ELM-C

In the design phase, learning goals are determined. Documents gathered during the analysis phase, business processes, and external sources (such as reports on domain trends) are examined. Educators and domain experts are questioned, both formally and in creativity sessions. The results are mapped into a scheme, which is a matrix of the learning goal dimension (Bloom, 1956, Krathwohl et al., 1964, Simpson, 1966) and the abstraction level, represented in an ARIS data model. Furthermore, relations between the learning goals, such as prerequisites and conceptual similarities, are modeled. As mentioned before, a learning sequence on "Simulation" has been developed based on our approach. Fig. 5 shows a simplified excerpt of a network of learning goals. Based on the network of learning goals, a possible structure of learning sequences (courses) is determined, serving as a prototype for user evaluation and as input for ELM-D.

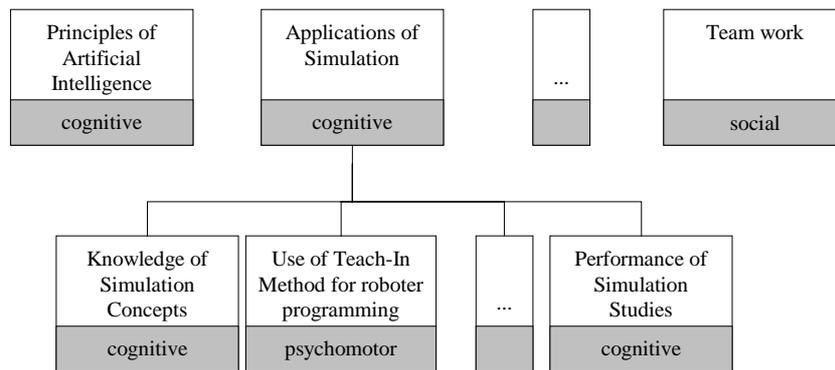


Fig. 5: Network of learning goals for a course on "Simulation"

ELM-D – Development of Learning Sequences

The focus of ELM-D is the development and implementation of learning sequences. Based on the specified learning goals from ELM-C, the most important concepts of a certain topic are specified in the phase “knowledge acquisition”. Those concepts are modeled as a data model in ARIS (or the chosen modeling tool). Furthermore a user model describes the proposed group of users. It contains personal and job related data (role, competencies), learning preferences, characteristics, and a history of learning activities. This data is also modeled in ARIS and used for an individual adaptation of learning sequences.



Fig. 6: Main processes of ELM-D

The crucial part of this phase is the selection of a didactical method, using educational technologies. This selection process is rule based, depending on learning goals, contents, and user characteristics. The selection generates alternatives, serving as a suggestion and support for the author. The ARIS process model furthermore supports the author in designing a specific learning sequence. Fig. 7 shows a sample model from the implemented learning sequence “Simulation”. In this context, the method “Simulation Game” has been chosen based on the complexity of the cognitive learning goals, the experiences of the user group, and the goal “teamwork”. Our methods base contains the “Goettingen catalog of didactical models” (Flehsig, 1991) and the didactical models from (Paulsen, 1995). Thereafter, the conceptual and implementation design of the specific CSLE is generated.

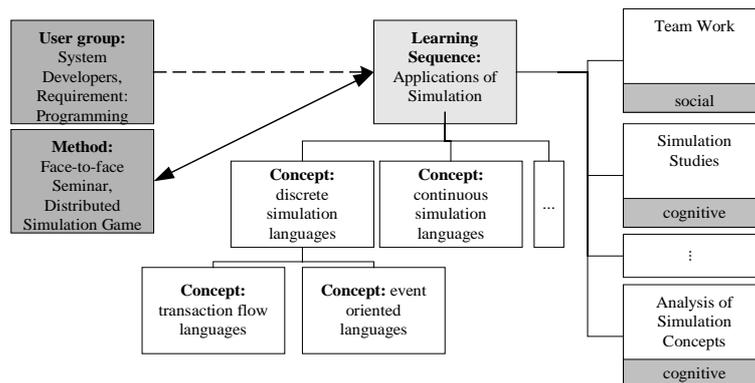


Fig. 7: Learning sequence on „Simulation“

ELM-E – Development of Learning Units

Finally, single learning units are designed and implemented in ELM-E. A pattern for the unit is generated, not limiting the user in the sense of Drill & Practice approaches, but serving as a guideline. Depending on the learning content, user group, and the selected method, structures for presentation and interaction/communication are suggested and implemented.

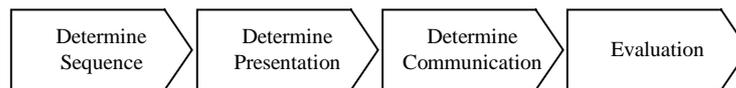


Fig. 8: Main processes of ELM-E

A process model of each unit is generated, serving as an orientation for the author and the users. Furthermore, this model can be integrated in other systems or, as suggested in (Adelsberger, Körner, Pawlowski, 1998) combined with business processes. The author is responsible for the consistency and final evaluation. Finally, the CSLE is implemented using standard authoring tools, or, in our university, in XML.

Conclusion

The multi-level approach of the Essen Learning Model leads to a variety of supporting features for both, developer and user. All phases of the development process are supported, using standard methods and formats (ARIS, LTSA, XML). This approach leads to several synergy effects: reusability, integration, and coupling with other resources. These effects are essential for future developments in the educational sector, helping project managers, designers, and authors of cooperative educational programs. The use of a development model will ensure the quality of educational technologies, helping companies and institutions becoming learning organizations and meeting the requirements of the future.

References

- Adelsberger, H.H., Körner, F. (1995). Data Modeling with IDEF1X, In: Adelsberger, Lazansky, Marik (Eds.): Information Management in Computer Integrated Manufacturing, *Lecture Notes in Computer Science*, 973, Springer: Berlin.
- Adelsberger, H.H., Körner, F., Pawlowski, J.M. (1998). A Conceptual Model for an Integrated Design of Computer Supported Learning Environments and Workflow Management Systems, *Proc. of Teleteaching '98: Distance Education, Training and Education*, XV. IFIP World Computer Congress, IFIP, Vienna.
- Balzert, H. (1998). *Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*. Spektrum, Akademischer Verlag: Heidelberg, Berlin.
- Bloom, B.S. (1956). *Taxonomy of educational objectives, Handbook I: Cognitive Domain*. David Mc Kay: New York.
- Booch, G., Jacobson, I., Rumbaugh, J. (1998). *The Unified Modeling Language Reference Manual*. Addison-Wesley: New York.
- Boehm, B. (1986). A Spiral Model of Software Development and Enhancement, *ACM SIGSOFT*, 11.
- EDUCAUSE (1999). *IMS Learning Resource Meta-data Best Practices and Implementation Guide. Version 1.0*, <http://www.imsproject.org/metadata/mbd01.html>, Download 1999-10-12.
- Euler, D. (1992). Didaktik des Computerunterstützten Lernens. In: Holz, Zimmer (Ed.): *Multimediales Lernen in der Berufsbildung*, 3, BW Verlag: Nürnberg.
- Farance, F., Tonkel, J. (1999). *LTSA Specification - Learning Technology Systems Architecture, Version 4.0*. <http://www.edutool.com/ltsa>, Download 1999-08-01.
- Flehsig, K.-H. (1991). *Kleines Handbuch didaktischer Modelle*. 3, Göttingen: Zentrum für didaktische Studien e. V.: Nörten-Hardenberg.
- Garzotto, F., Paolini, P., Schwabe, D. (1993). HDM: A model-based approach to hypertext application design, *ACM Transactions on Information Systems*, 11(1).
- Heimann, P., Schulz, W., Otto, G. (1965). *Unterricht - Analyse und Planung*, Schroedel: Hannover.
- Heeren, E. (1996). Technology Support for Collaborative Distance Learning. *CTIT Ph.D.-thesis series*: Enschede.
- Issing, L. (1997). Instruktionsdesign für Multimedia. In: Issing, L., Klimsa, P. (Ed.): *Information und Lernen mit Multimedia*. 2, Beltz Psychologie-Verlags-Union: Weinheim, Basel.
- Kinshuk, Patel, A. (1997). A Conceptual Framework for Internet based Intelligent Tutoring Systems, *Knowledge Transfer*, Vol. II, S. 117-124, London.
- Krathwohl, D.R., Bloom, B.S., Masia, B.B. (1964). *Taxonomy of educational objectives, Handbook II: Affective Domain*. David Mc Kay: New York.
- Paulsen, M. F. (1995). *The Online Report on Pedagogical Techniques for Computer-Mediated Communication*. <http://www.hs.nki.no/~morten/cmcped.htm>, Download 1999-10-10.
- Royce, W.W. (1970): Managing the development of large software systems, *IEEE WESCON*, August 1970.
- Scheer, A.-W. (1998). ARIS. In: Bernus, Mertins, Schmidt (Eds.): *Handbook on Architectures of Information Systems*. Springer: Heidelberg.
- Simpson, J.S. (1966): *The classification of educational objectives, psychomotor domain*. Office of Education Project No. 5-85-104, University of Illinois, Urbana.
- Steppi, H. (1997). *CBT- Computer Based Training: Planung, Design und Entwicklung interaktiver Lernprogramme*. Klett: Stuttgart.
- W3C, World Wide Web Consortium (1998). *Extensible Markup Language (XML) 1.0*. W3C Recommendation, 02-10-1998.