

SOFTWARE DEVELOPMENT PROCESS MODEL AND METHODOLOGY FOR VIRTUAL LABORATORIES

ANSGAR SCHERP

Multimedia and Internet Information Services

Kuratorium OFFIS e.V.

Escherweg 2, D-26121 Oldenburg/Germany

email: Ansgar.Scherp@OFFIS.DE

ABSTRACT

The existing software development process models and methodologies of the software engineering are too abstract for the development of virtual laboratories and not profitable in practical use. The intent of this paper is to develop a software development process model with corresponding methodology, that is specifically orientated on the development of a small application-class of multimedia CBT systems, the so-called virtual laboratories. It was not the aim to produce another development process model for an as wide as possible application-area. As basis the Rational Unified Process (RUP) was used because it is especially simple to adapt and to enlarge to a concrete application-case and it already includes a roll-concept to support the heterogeneous developer team of virtual laboratories.

KEY WORDS

Software Engineering, Modelling and Simulation, Intelligent Tutor Systems, Rational Unified Process

1. INTRODUCTION

The virtual genetic laboratory (*GenLab*) developed at the OFFIS¹ in the work group multimedia systems of Prof. Dr. H.-J. Appelrath makes the interactive and simulation-based training of genetic experiments in a virtual environment on a customary computer system possible [1]. An extensive knowledge-component illustrates the corresponding molecular-biological reactions and explains the right contact with the laboratory devices and substances (see also [2]). However, the development of a virtual laboratory like *GenLab* is very time-consuming and expensive. For this reason, *GenLab* is followed by the project *VirtLab*. In *VirtLab*, we abstract from the concrete implementation of the virtual laboratory *GenLab* and create methods and tools for the development of more general scientific, technical laboratories and practical courses (*virtual laboratories*). Declared goal of *VirtLab* is to enable a fast and cost-efficient production of high-quality virtual laboratories [3].

¹ See: <http://www.offis.de/indexe.htm>

1.1 MOTIVATION

Today professional software development is no longer imaginable without the use of a process model. Astonishingly, the software development process models and methodologies of the software engineering accrued on the basis of the software crisis in the sixties hardly find application in the development of multimedia systems and multimedia CBT systems [4, 5]. Particularly the area of the multimedia CBT systems puts very high requirements at the project organization, planning and conception. A reason for this is the heterogeneity of the developer team (see section 4). Therefore it doesn't astonish that improved process models are named in the study of [4] regarding the deficits and wishes in the development of multimedia CBT systems. The further while developing the *GenLab* system it was determined that besides a development process model, a corresponding methodology is important for virtual laboratories. The existing software development process models and methodologies of the software engineering are too abstract to be profitable for the development of virtual laboratories. This article therefore devotes time on the construction of a development process model and methodology that is specifically orientated on the application-case of virtual laboratories, the so-called *VirtLab-Process*. In contrast to many other process models of the software engineering, the attempt here was not to produce a development process model with corresponding methodology, that can be applied for an as wide as possible application-area. The project *GenLab* has played a fundamental role thereby. The results and experiences won in *GenLab* have flowed in inductively into the development of the *VirtLab-Process*.

1.2 STRUCTURE OF THIS ARTICLE

The term virtual laboratory is clarified in the next section. Thereafter a short overview and description of the *VirtLab-Process* is given in section 3. In section 4 the roles of the interdisciplinary developer team of virtual laboratories are described. After that a workflow of the *VirtLab-Process* is introduced in section 5 as an example. In final section 6 a conclusion of the *VirtLab-Process* is given.

2. VIRTUAL LABORATORIES

The term virtual laboratory is used in many ways: In [6] a virtual laboratory is for example a real laboratory at an university with corresponding laboratory devices, receptacles and substances, that can be observed by tele-users with help of cameras and can be remote controlled with robots. On the other hand textual web-pages with integrated animations, *Java-Applets* or *Shockwave-Movies*, that are used for simulation of laboratory equipments or processes, are frequently named virtual laboratories. An example is *Virtual Lab: Exploring Genetics* (see: <http://library.thinkquest.org/11375/>). In *VirtLab* the multimedia reproduction of a real laboratory into a computer system is understood by a virtual laboratory. The components of a virtual laboratory correspond to those of a real laboratory and are for example virtual laboratory devices, accessories and substances. A virtual laboratory can consist of different workspaces. The learners can move with help of mouse and keyboard in the virtual laboratory, use laboratory devices and perform experiments. Particular mark is the high interaction of the user interface. Laboratory devices, accessories and substances can be used in almost any order. Virtual receptacles and substances can be moved for example with help of a *transportation-ledge* between different workspaces. A virtual laboratory is especially used for learning the partial very complex work-steps and the needed technologies of scientific, technical experiments. The individual work-steps of the lab procedure are shown in an *instruction-window* for example [7]. Figure 1 shows an example of such a workspace (from *GenLab*).

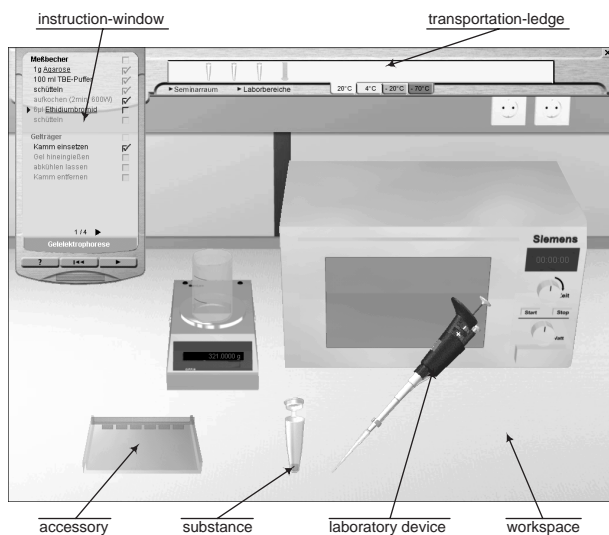


Figure 1. Example of a workspace of a virtual laboratory

Summarized a virtual laboratory is a multimedia, simulation-based, explorative CBT system. On this basis, a software-system, that consists in the main of a *simulator* [8] and an *intelligent tutor system* [9], is understood in this article by a virtual laboratory. The simulator becomes use

as pure tool but yields a multimedia CBT system in connection with the intelligent tutor system.

3. OVERVIEW TO THE VIRTLAB-PROCESS

The *VirtLab-Process* is based essentially on the RUP [10]. It consists of the four sequential phases *Inception*, *Elaboration*, *Construction* and *Transition* (see figure 2). At the end of each phase there is at least one milestone that defines the artifacts of the phase. Each phase consists of at least one iteration. An iteration consists of the ten workflows of the *VirtLab-Process*. As shown in figure 2 the *VirtLab-Process* contains almost all workflows of the RUP. In contrast to the RUP, however, it contains no workflow for Business Modeling since this is only partially required. The workflow of the Tutor Concept is new. This includes the didactic aspects of virtual laboratories (see section 5). The relevant activities of the workflow Business Modeling have been integrated in this workflow. The *VirtLab-Process* was supplemented the further by the workflow Media Production. This comprises all activities for planning and manufacture the media. Furthermore the workflow Test was supplemented about evaluation of the virtual laboratory (Test and Evaluation). At the end of an iteration there is again at least one milestone. This defines important intermediate artifacts in a phase.

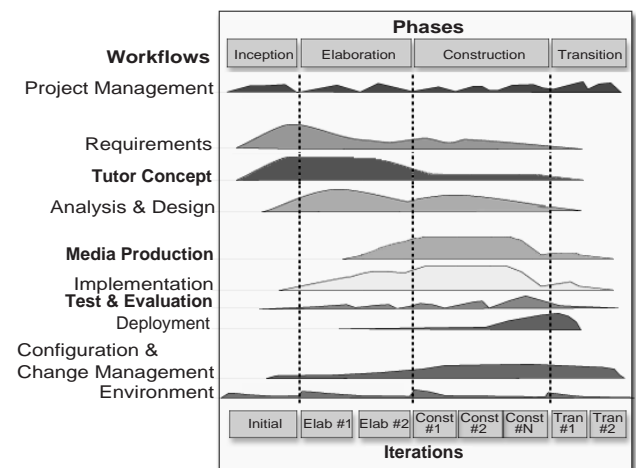


Figure 2. Phases and workflows of the development process model for virtual laboratories

The basic requirements of the ordering customer are investigated in the Definition Phase and hold down in the customer requirement specification. In addition, the required time and costs for the development is clarified and the feasibility of the development is rated. If doubts exist at the feasibility of requirements, preliminary studies or simple prototypes are created. The further all use-cases are identified and hold down in the use-case model. The most im-

portant use-cases are those to selection and accomplishment of the individual experiments. The initial specification of these use-cases is already given by the corresponding experiment protocols. The Definition Phase is always enforced, if a new virtual laboratory is developed [11] or extensive maintenance work for a legacy virtual laboratory is planned [12].

A detailed analysis of the virtual laboratory is enforced in the Elaboration Phase. Furthermore the architecture, i. e. the framework of the virtual laboratory, and the didactic concept is defined. The most important result of the Elaboration Phase is an almost² completely specified use-case model, i. e. at least for each experiment is an annotated protocol and a storyboard available (see section 5.4).

In the Construction Phase, the requirements and architecture of the virtual laboratory are widened and refined. Within an iteration, the activities respecting analysis, design, implementation, media production, test and evaluation are sorted and enforced by priority. Each iteration is connected with a new internal release, the test-courses and a demonstration at the ordering customer. If all experiments are built, the Construction Phase finishes with the first external release.

In the Transition Phase, the virtual laboratory is tested under realistic conditions. If the introduction is connected with a big organizational effort, for example if the virtual laboratory is put into combination with a real practical course, then the Transition Phase can take place in several iterations. If the virtual laboratory was tested adequately and regarded as ready for use, the Transition Phase finishes with the acceptance of the last external release by the ordering customer and the project is declared as completed.

4. THE ROLES OF THE VIRTLAB-PROCESS

With the development of virtual laboratories the involved persons of the interdisciplinary developer team³ have different roles. The most general role is the *project collaborator*. This can be apportioned into the four organization units *domain expert* of the domain to be modelled, *domain tutor*⁴, *media specialist* and *computer scientist*. The domain experts give their knowledge about the domain to be modelled. This consists of the theoretical basics, the comprehensive knowledge about the laboratory equipment and lab procedures. The domain tutor is responsible for the preparation and conception of the domain contents. This includes the representation of the learning content cut on the target group as well as defining the navigation-structure and -possibilities of the lessons. The media specialists are responsible for an appealing multimedia presentation of the

²In [13] is demanded, that approximately 80% of the use-cases are completely specified.

³Compare the spreading of the developer team in [14, 15].

⁴The explicit introduction of a role for the didactics of CBT systems is not necessary, because the didactic concept of virtual laboratories is already fixed by the Tutor Concept.

virtual laboratory and enforce all activities of the media production. So, for example lifelike images of the laboratory devices and receptacles have to be created. The computer scientists are responsible for analysis, technical design and implementation of the virtual laboratory. Moreover, the project organization is adopted by them. Further on the two external roles *ordering customer* and *final user* exist. The ordering customer is the investor of the virtual laboratory and carries the financial risk of the development. The final users respective learners are the recipient of the virtual laboratory.

5. EXAMPLE FOR A WORKFLOW: THE TUTOR CONCEPT

In this section – exemplary for the remaining workflows of the *VirtLab*-Process – the workflow of the Tutor Concept is introduced (see figure 3). This includes the development of the contents and didactic parts of the intelligent tutor system and the experiments of the virtual laboratory (see [16]). In the next paragraph the workflow is described on the level of the development process model, i. e. it is decided when which activities are enforced. In the following sections 5.1 to 5.5 the activities are viewed in detail and it is described how to create and use the artifacts of the development methodology.

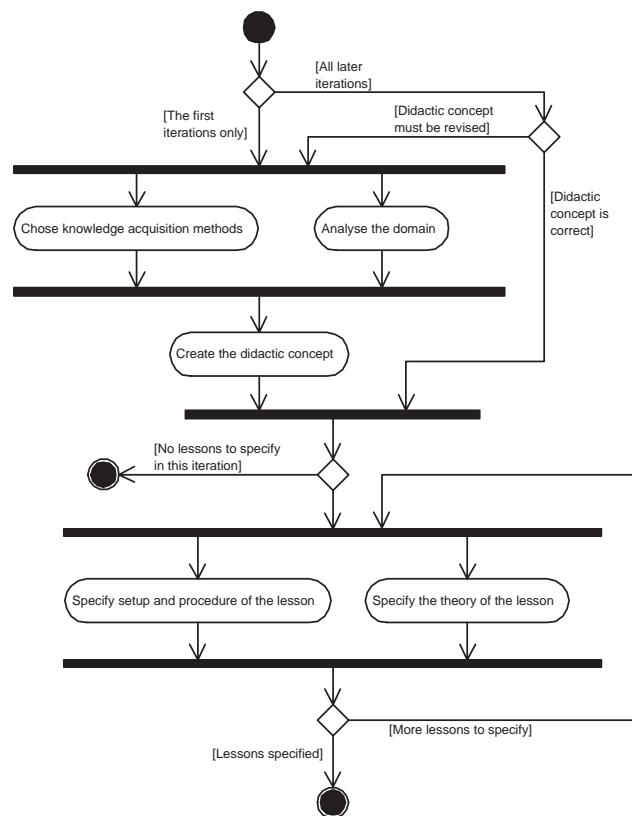


Figure 3. The Workflow to the Tutor Concept

First of all the construction of the didactic concept is viewed while the primary iterations of the *VirtLab*-Process. As to see in figure 3, the available sources and methods for the knowledge acquisition are determined (see section 5.1) as well as an detailed analysis of the domain is enforced (see section 5.2). Afterwards, the didactic concept of the virtual laboratory is clarified (see section 5.3). Already in the first iterations of the *VirtLab*-Process it is possible to specify the experiments and skills (see section 5.4) and the needed theory (see section 5.5) precisely. Moreover, the didactic concept can be corrected in each iteration. The methods for the knowledge acquisition are chosen again before and an analysis of the domain is to enforce.

5.1 CHOSE KNOWLEDGE ACQUISITION METHODS

In this activity the methods for the knowledge acquisition are determined. Possible sources are books or documents of the domain in printed or electronic form, like the experiment protocols of a real practical course for example. These can be processed by means of text analysis. The domain expert is available as further source of information. Relating to this, the methods of the interview, formal techniques and observation techniques can be applied.

5.2 ANALYSE THE DOMAIN

The domain of the virtual laboratory is analysed in this activity. Therefore the available sources of the knowledge acquisition are analysed and edited with the methods selected before. The further the needed laboratory devices, receptacles, substances and accessories are determined for each attempt and each skill. Furthermore, the theory of the experiments is identified and the safety regulations are determined. With the analysis of the domain it is especially important to take the different linguistic usage and different ways of thinking of the project collaborators into account. Same terminologies and concepts are often differently or even incorrectly used and understood [17]. Possible misunderstandings are to remove and an unification about the usage of the terminologies and concepts is to reach. Result of this activity is the definition and preparation of the learning content, i. e. a *list of lessons* sorted by priorities. A lesson respective learning unit is either an experiment that is to conduct or a skill that is to acquire.

5.3 CREATE THE DIDACTIC CONCEPT

In this activity the *navigation-structure* over the lessons is defined first. This describes the logical dependencies between the individual experiments and skills and represents the base of the didactic concept at the same time. With help of the navigation-structure, a *guided tour* through the virtual laboratory can be suggested for example. In contrast,

a *free navigation* is imaginable with which all lessons are selectable at each time.

The *navigation-possibilities* in the lessons go from *free experiment-possibility*, with which all workspaces, laboratory devices, receptacles and substances can be used in any order, over *locking of individual workspaces, laboratory devices and receptacles* until to *guided experiment-conduction*, with which only one action can be executed at the same time, and finally the animation of the lab procedure (*no interaction-possibility*). Respecting the navigation-possibilities in the lessons, also the interaction-design is to be taken into account. The interaction-forms of graphic controls and direct-manipulative metaphors are available for it primarily. Furthermore the input devices are to determine like for example mouse and keyboard and how these can be used for the interaction-forms (see [18]).

The variants of support by the virtual laboratory for the learner go from *no support*, that is meaningful with free experimentation, beyond a *permanent support*, for example with an instruction-window (see section 2), the *support on inquiry* through the learner until to *adaptive support*. With the last variant, the intelligent tutor system decides, when and how often the learner gets help. This type of support is meaningful with the acquisition of a plot, i. e. the sequence of work-steps.

For the conception of the *learner-analysis*, the criterions for the learning success must be determined. The further it is to determine which support material is still allowed beside the virtual laboratory as source of information. *Learner-data* has to be recorded for learner-analysis. This are all information, that can be determined from the input devices. It is important that only the relevant information is recorded. Therefore a suitable *protocol structure* should be created. It is for example important to record in which sequence the work-steps are conducted and how much time is required for it. Also, number and type of the mistakes have to be recorded, because this provides important information about the learning success.

In order to valuate the actions of the learner, a *score-pattern* based on the protocol structure is introduced and a score is given for each possible activity of the learner. The valuation of a lesson is determined by simple accumulation using either summation or averaging over the scores of the single actions the learner does during experiment conduction (cp. [19]). If learner-data are recorded, the learner has to be informed about it and the purpose has to be declared [20].

5.4 SPECIFY SETUP AND PROCEDURE OF THE LESSON

In this activity the setup and procedure of the lessons, i. e. the experiments and skills of the virtual laboratory, is specified with help of the corresponding *experiment protocols* of a real laboratory practical course. Experiment protocols contain a short and sequential description of the work-steps that are necessary to conduct an experiment and are written

by the domain experts. With the specification of the lessons is to regard, that important additional information is not contained in the experiment protocols and is presupposed implicitly as known. For example the piece of information is missing, which laboratory devices and receptacles are taken and how they are used. Also is missing in which receptacles the substances are stored, how they look like and which storage and safety regulations are to be kept. The work-steps are described sequential in the experiment protocols, but in principle it is possible that certain work-steps can be processed in another sequence. In the experiment protocols is missing, how the single work-steps depend on each other and which can be interchanged. The further the experiments often consist of several sub-experiments. These are identified in order to reuse already implemented sub-experiments, laboratory devices and receptacles. Experiment protocols supplemented with these and further information (see [16]) are named as *annotated (experiment) protocols*. The annotated protocols are checked by the domain tutor on gaps and mistakes. Furthermore work-steps are simplified if possible.

With the specification of the lessons, the limits of the *virtualisation* should be known by the domain expert and domain tutor, i. e. the problems, that appear with the implementation of the experiment protocol and corresponding laboratory devices, receptacles and substances on a computer system. These appear in the differences between the real and the virtual experiment conduction. An example (further in [16]): If many tests (e. g. 30 pieces) are produced in a real experiment, these can no longer meaningfully be represented and handled in the virtual laboratory. The high number of tests is necessary in general to determine a significant test result because of the variation in the yield at all. For the virtual laboratory is to examine, whether it can be abstracted and idealized from the high number of tests, so that also a more inferior number is enough for a successful experiment conduction.

On the basis of the annotated protocol a storyboard is created to visualize the experiment. This consists in essential of a graphic sketch of the experiment setup and procedure. From the experiences of *GenLab* is to proceed thereby as follows: The lab procedure is sketched first (see figure 4 left), i. e. the work-steps of the annotated protocol are drawn down. The sketch of the experiment protocol has to be intuitively comprehensible and needs no further explanation. After that the setup of the experiment is sketched (see figure 4 right). Referring to this the needed workspaces are determined on basis of the lab procedure sketch, i. e. the single pages of the storyboard are looked at and the needed workspaces are counted. Thereafter the laboratory devices, receptacles and substances become arranged on the workspaces. If the workspaces don't own adequate place, so the virtual laboratory must be extended accordingly.

For the acceptance of the virtual laboratory, it is crucial that the specification of the lab procedures are examined by external domain experts to determine variations and suitable

abstractions of lab procedures at other universities or institutes.

5.5 SPECIFY THE THEORY OF THE LESSON

To each experiment, laboratory device, receptacle and substance belongs a theoretical background knowledge (*theory*). This is determined by the domain expert and prepared by the domain tutor. Different media types can be used: Text for example is used to describe the theory of the experiments. Graphics, like drawings and digitized photos for example, are used to visualize reaction-equations and to represent laboratory devices and receptacles. With 3D-models, the learner can look at the laboratory devices, receptacles and the molecular structure of the substances from all sides and can experience these interactively. Furthermore the theory can be supplemented by videos, sounds and animations.

6. CONCLUSION

The *VirtLab*-Process introduced in this article represents a systematic approach of a software development process model and methodology, that is specifically developed for the application-case of virtual laboratories. As shown in [16], the workflows of the RUP could be adopted to the application-case of virtual laboratories almost unchanged. A high recognition effect accrues therefore to those readers, who already know the RUP, what increases the acceptance of the *VirtLab*-Process. Summarized the *VirtLab*-Process fills the wish for improved software development process models for multimedia CBT systems [4] respecting the particular application-case virtual laboratories. Unfortunately only the workflow of the Tutor Concept could be described in this article. But in order to understand how the *VirtLab*-Process works, it is already sufficient to look at one workflow only. For a complete and detailed description of all phases, workflows, activities and artifacts of the *VirtLab*-Process see [16].

A set of CASE-tools are imaginable to support the development methodology of the *VirtLab*-Process. These can make the development of virtual laboratories more efficiently and affordable, so that the manufacture becomes economic also for very small target groups and more attractive for publishers [21]. Also the framework plays a crucial role to make the production of virtual laboratories affordable. The framework cannot be limited to one domain but must be applicable for all types of scientific, technical laboratories. Independently from this point, it should become deliberate – motivated by the successful adaption of the RUP for the application-case virtual laboratories – which further application-classes are worthwhile to create a particular development process model and methodology for it (cp. [22]).

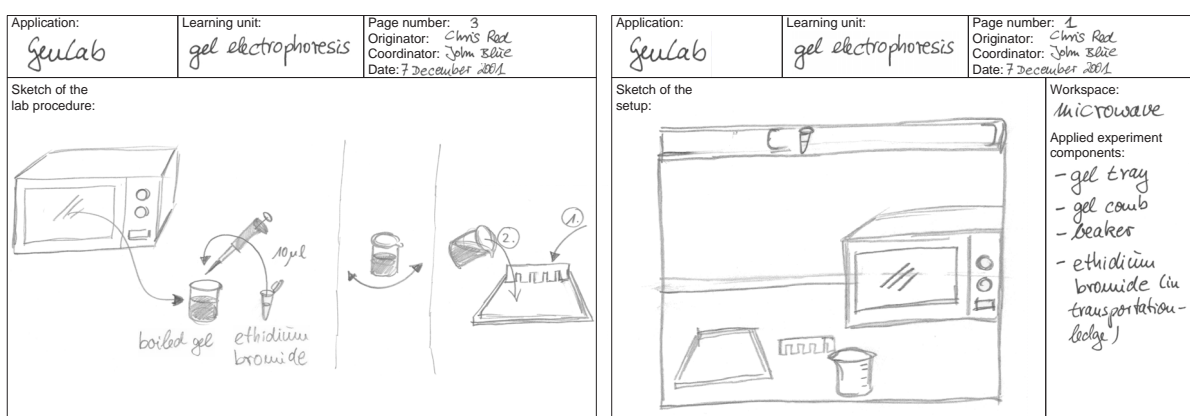


Figure 4. Example of a storyboard representing procedure and setup of an experiment

REFERENCES

- [1] GenLab - Das virtuelle gentechnische Praktikum, 2001. URL <http://www.offis.de/genlab/>.
- [2] OFFIS. *OFFIS annual report 2000*. Vorstand des Kuratorium OFFIS e.V., 2000. URL http://www.offis.de/jahresbericht/jb00_english/jb00.htm.
- [3] VirtLab - Methoden und Werkzeuge für virtuelle Labore, 2001. URL <http://www.offis.de/virtlab/>.
- [4] H. Balzert, A. Behle, U. Kelter, M. Nagl, P. Pauen, W. Schäfer, H.-W. Six, J. Voss, J. Wadsack, C. Weidauer and B. Westfechtel. *Softwaretechnische Anforderungen an multimediale Lehr- und Lernsysteme*, 1999.
- [5] D. Boles, E. Boles, P. Dawabi, M. Schlattmann, C. Trunk and F. Wigger. *Objektorientierte Multimedia-Softwareentwicklung: Vom UML-Modell zur Director-Anwendung*. In *Tagungsband: Workshop Multimedia-Systeme der GI-Jahrestagung*, 1998.
- [6] VVL. *Verbund Virtuelles Labor*, 2001. URL <http://www.vvl.de/>.
- [7] A. Hasler and M. Schlattmann. *Multimedienprojekt Gentechnisches Praktikum - Endbericht*, 2001.
- [8] V. Claus and A. Schwill, editors. *Duden Informatik: ein Sachlexikon für Studium und Praxis*. Dudenverlag, 1993.
- [9] O. Schröder. *ITS, Intelligente tutorielle Systeme*. In *Wörterbuch der Kognitionswissenschaft*. Klett-Cotta, 1996.
- [10] P. Kruchten. *The Rational Unified Process: an introduction*. Addison Wesley, 2000.
- [11] H. Balzert. *Lehrbuch der Software-Technik*, volume 1. Spektrum, 2000.
- [12] P. Kruchten. *Software Maintenance Cycle with the RUP*. In *The Rational Edge*. Rational Software, August 2001.
- [13] K. Bittner, P. Kruchten and C. Larman. *Wie Sie ein Projekt mit dem Rational Unified Process garantiert in den Sand setzen können*. *OBJEKTSpektrum* 4/2001, 2001.
- [14] M. Kerres. *Multimediale und telemediale Lernumgebungen: Konzeption und Entwicklung*. Oldenbourg, 1998.
- [15] L. J. Issing. *Instruktionsdesign für Multimedia*. In L. J. Issing and P. Klimsa, editors, *Information und Lernen mit Multimedia*. Psychologie Verlags Union, 1997.
- [16] A. Scherp. *Vorgehensmodell und Entwicklungsmethodik für virtuelle Labore*. Universität Oldenburg, 2001.
- [17] B. Gaines and M. Shaw. *Knowledge Acquisition Tools based on Personal Construct Psychology*, 1995.
- [18] D. Boles. *Das IMRA-Modell - Integrationen von Interaktionen in das Autorenwerkzeug FMAD*. Universität Oldenburg, 1994.
- [19] P. Norvig and S. J. Russell. *Artificial Intelligence: a modern approach*. Prentice-Hall, 1995.
- [20] BDSG. *Bundesdatenschutzgesetz*, 2001. URL <http://www.bfd.bund.de/information/>.
- [21] Schlattmann et al. *Werkzeuge und Methoden zur Entwicklung virtueller Labore*. In E. Wagner and M. Kindt, editors, *Virtueller Campus*, volume 14 of *Medien in der Wissenschaft*, 2001.
- [22] W. Bartussek, P. Strooper and D. Weiss. *Defining Software Families*. In *Software Engineering Research and Education: Seeking a new Agenda*, 1999. Dagstuhl Seminar 99071.