

The monist project

A system for learning and teaching with educational simulations

Sören Lorenz, Wolfram Horstmann, Markus Oesker, Hubert Gorczytza, Andreas Dieckmann, Martin Egelhaaf
Department of Neurobiology, Faculty of Biology, Bielefeld University, Germany
soeren.lorenz@uni-bielefeld.de

Abstract: The monist-project has committed itself to the task of developing recipes for the use of simulations in higher education in neural and cognitive science. The basis of the monist-project is a software package, the monist console, designed for providing compact, ready to use educational simulations with variable application context. The monist console is a client-server application supporting online and offline learning with simulations. Simulations become educational as they are complemented by topical texts, instructional design, concrete tasks and an editor for individual notes and solutions. This 'format' is integrated in the monist-console. Educational simulations can be used in nearly every learning context found in higher education. They can selectively supplement lectures and tutorials or can be compiled to complete courses. Online file-management and communication tools as well as an learning management (via monist-server) help to organize personalized courses.

1. Introduction

In recent years, there has been great scientific benefit in explaining brain function by model simulation [1, 2]. In higher education, simulations are introduced only sporadically – 'recipes' for applying simulations in education are widely missing. The monist-project has committed itself to the task of supporting the use of model simulations in higher education. Primary goals are to improve the design for educational simulations and to develop an appropriate environment for learning with and authoring of this type of educational simulations. This paper describes the system design, its modules and contents in detail and demonstrates their possible application in higher education.

Though the main topics are neural- and cognitive sciences, the system is designed for the integration of educational simulations in higher education in general. The system is open-source (GPL-license) and is available at the monist-homepage [3]. Contents that have been produced during project time are available too, though in parts with restricted access. Potential users outside the monist-group are invited to share existing contents with the monist-group or to add their own contents to the system (monist content license [3]).

To bundle up expertise in the area of neural- and cognitive science, the monist-group consists out of nine departments in six German universities, all using simulations in science as well as in education. Each member is an internationally known expert in his field. Table 1 gives an overview. The monist-system, as introduced in this paper, has been developed by members of Bielefeld University. Every participating department contributes to the content with regard to its major scientific topic, except one group, evaluating project. Since the evaluation is still in progress evaluation results are not part of this paper. The project, existing since January 2001, is funded until April 2004 and offers several future perspectives. The monist-console will be published in May 2004.

2. The monist console

The monist-project does not only aim at designing a set of standard simulations for the neural- and cognitive sciences – it aims to build a complete system for the lasting integration of simulations in higher education. Therefore, it is not sufficient to offer ready-made and unchangeable simulations that will be actual next 2 or 3 years. For lasting integration, a flexible system is needed that supports functionality not only for learning but also for teaching and authoring.

The monist-console is such a system: a client-server application supporting online and offline learning with simulations. It is based on a standard client-server application written in Java. It applies Remote Procedure Calls

(RPC) and is connected to a MySQL-database both, locally and on the server. The latter assures online *and* offline learning with simulations. That comply the demand of learners and teachers to use the functionality offline as well.

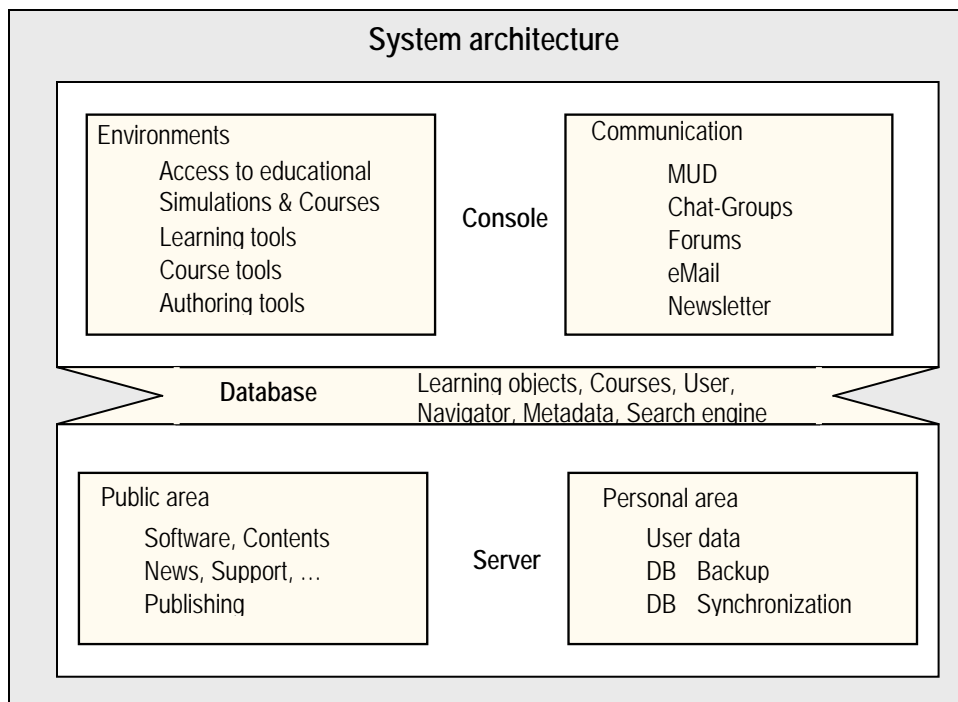


Fig.1: Architecture of the monist system. The figure illustrates the main modules of the monist-console.

There are mainly three parts, the console, the server and the database (locally and on the server) with its access tools. In combination with the database, the console provides access to all contents of the monist-system.

The server provides general tools, a public area and the personal area for every user. For details of functionality see section 2.

The monist-console consists of several modules, which are interconnected (see figure 1). The essential items of the system are compact, ready to use *educational simulations* with variable application context (see section 3 for detail). They can selectively supplement lectures and tutorials or can be compiled to complete courses. A *navigator* provides tree-structured access to all contents stored in the *monist-database*, e.g. courses, educational simulations, general media, external programs, etc. Any object in the database is a *virtual learning object* (vlo), no matter if it is a figure, a glossary link, a simulation, an educational unit, or even a user. Educational material is bibliographically and educationally declared by a standard metadata set for easy retrieval and to maximize the reuse of material by different users. The monist-metadata set is a subset of the Dublin core metadata standard (DC), completed by some specific educational and simulation metadata. It describes the educational use and simulation characteristics of a given unit. The actual educational simulation can be viewed with the *content browser* (see figure 1 and 2). Tasks are integrated in educational simulations as well as forms for solutions. Supported by a *learning management*, students are able to upload their solutions to the server (see figure 2, left part). Tutors have exclusive access to the students' solutions. Complete experimental records are supported by tutorials and forms. A *content editor* enables authoring of educational simulations.

Communication System

An integrated *communication system* supports synchronous and asynchronous communication between students and teachers within the system. That way, problems, task solutions and corrections can be discussed. Additionally, the communication system helps to organize different educational settings where a computer based communication is needed or at least useful. That applies to holds for online-learning, blended learning and forms of collaborative learning (see also section 4).

The communication system works with standard client-server architecture. The communication-server is originally based on the open-source-project "Key" (see <http://key.sourceforge.net>), a virtual world (MUD – multi user dungeon – or *talker*) written in Java that has been restructured and currently serves as a communication platform and meeting-point. The communication system is integrated in the monist-console. Like the monist-console, the communication system supports Remote Procedure Calls (RPC): Both client and server support the Monist-specific

RPC-mechanism which allows communication between different monist-applications. The talker can provide a virtual world with rooms that users can explore and use to communicate. The talker unites both basic features of typical synchronous (chats, messaging system) and asynchronous (email, news) communication-media. The communication functionality can be accessed either by command-line or by graphical user-interface (GUI) for a more intuitive approach to the basic features of common communication tools.

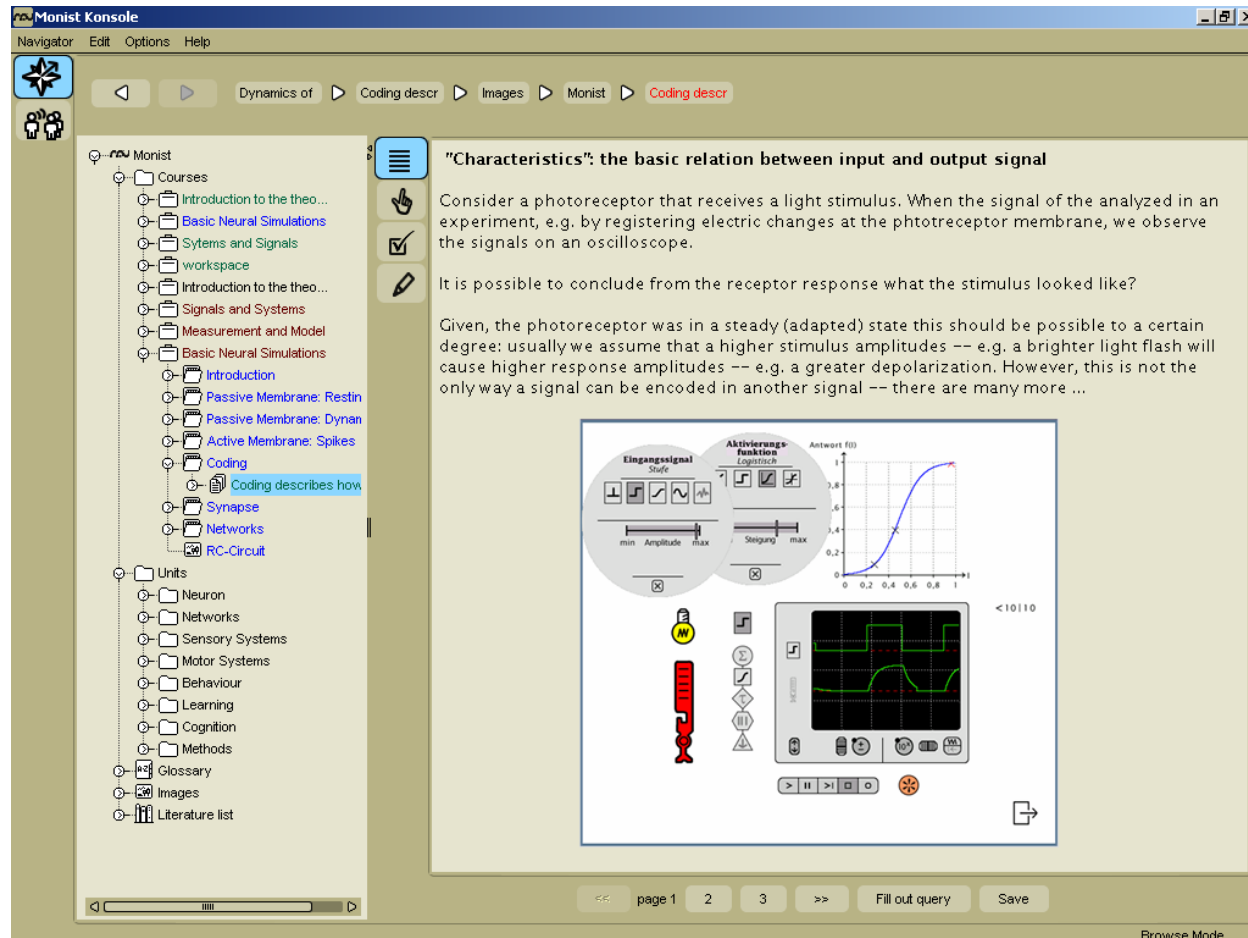


Fig.2: Screenshot of the monist console with *Navigator tree* (left part) and *content browser* (right part). Additionally, a RUBIN-simulation is linked to the unit, providing dynamic visualization and parameter variation for the actual model. The tree-structured navigator gives access to courses and units.

By integration, the navigator and talker can cooperate. One example is that any object stored in the monist-database (unit, learning simulation, image, etc.) can be sent directly via mail or news to other users. Tutors automatically receive an email, when students upload their solutions to the monist-server (e.g. to the navigator). The mail contains a link to the uploaded files.

User specific environments

By restricting or enabling the functionality of the described modules for different groups of users, specific environments emerge for learners, teachers, and authors. The restriction mainly comes with access rights of individual users. Used by a learner, for instance, the whole functionality for teachers and authors will be switched off. This is an easy way to create functional environments within the same system. Therefore, it is not necessary to split the environment into different program modules.

The *learning environment* is suited for browsing through courses, working with simulations, making individual notes and solving the given tasks. Solutions or records can be uploaded to the server. It gives access to glossary and literature. In the case of online-learning, the communication tool gives room for questions, discussions, and

commentaries. Additionally, students are not restricted to an actual course, but have the opportunity to view all offered contents for self-studying or exam preparation.

The *teaching environment* is equipped with additional functionality. Teachers can organize their courses and course members. They have access to a learning management. All uploaded task solutions can be viewed as well as comments can be sent via the communication system. Teachers have access to all educational content provided by the monist-database. They can use precompiled courses, organize personalized courses, or mix up several existing materials to a new course via the navigator. And if the current collection is not sufficient, they can produce their own material by using the content editor (see below). Using the communication system during an online-course, teachers can support learning or administrate via chat, mail and news. Additionally, they can create course rooms and newsgroups in the talker.

To support an easy production process for educational simulations, an *authoring environment* is provided for designing and modifying educational material and complete courses. It consists of the content editor, supporting the format for educational simulations as described above. Authors can write topical texts, design instructions and tasks. Existing glossary and literature entries can be linked or new entries added. Any type of media can be integrated into the unit, including different simulation formats. Already existing simulations, like applets, programs and the like, can easily be embedded in monist learning units via external links.

An integrated *simulation construction kit* is under development. It will also allow for designing interactive elements beyond the simulation itself, like visualizations or data analysis tools. Any new object can (and should) be equipped with metadata descriptions provided by the monist *metadata tool*. This assures finding and reuse of objects stored in the monist-database. Authored material can be submitted to the server for publication under the monist-content license.

3. Educational simulations

Scientific simulations are usually not functional for students. There is a lack of textual information and a clear description of the instructional use or the educational context, a simulation is intended for [4, 5]. A lot of examples can be found in the internet. The reuse of hundreds of simulations as applets suffers from this lack. They are rarely equipped with contextual information, are hard to handle and the data-visualizations are not self-explaining. In most of the cases, simulations can only be used with the extensive help of their authors.

Three main extensions are able to transform a given simulation into an educational simulation:

- Complementation by *contextual information*, like topical texts, instructional design, concrete tasks and an editor for individual notes and solutions, glossary and literature entries,
- *Interface design* allowing for synchronous and asynchronous manipulation of model functions and parameters.
- Dynamic spatiotemporal visualization of data, e.g. oscilloscope or animated model representations and interactive plots.

To assure a reusable design, educational simulations in monist are equipped with these features to make simulations transparent and self-explaining. All educational simulations follow the same structure, even though the simulation parts can origin from other programs and formats and differ substantially.

Contextual information is the basic feature that transforms a given simulation to an educational simulation. This context information is important for students to fully comprehend the models' behaviour and for teachers to understand the use of a given simulation (unless they are the author).

In monist, educational simulations are organized in pages. Context information is divided in topical texts, instructions, concrete tasks and a text editor for student's personal notes, available on every page of a given unit. They are accessible via tabs. Each text type can be exported to html. In this way, information on simulations can be used in contexts other than monist. All text types may be equipped with several link types: glossary links, literature links, links to other educational simulations and last but not least links to external resources like simulations, applets, or applications (see table 3 for examples).

Another crucial feature is the simulations interface design. Simulations often provide unfiltered access to all model parameters, easily more than 10, to allow a wide range of experiments and exploration of the models behaviour. Learners, who are unfamiliar with model simulations, are often unable to cope with this wide range of possible parameter sets. Therefore, the effective reduction of changeable parameters *per page* is an important concern of these educational simulations introduced here. This leads to the concept of restricted model access. For example, this can be achieved by offering a single parameter per page. In this way, the focus lies on this and only this parameter and its influence on the model's behaviour. Combinations of parameter variation can be introduced in an intelligible manner. The model access can get more complicated with each page.

The main advantages of scientific simulations are, in general, visualization possibilities of the models data. Evidently, this is also true for education. To describe the model's behaviour in words is inadequate. Even formulas as such will not give an insight to the way dynamics evolve in this kind of systems. Therefore, an appropriate visualization of the spatiotemporal dynamics is essential for understanding the processes going on in the simulation. The typical educational simulations provided in monist are equipped with several visualizations that can be applied in parallel, e.g. colour-coded spatial representation of the models overall activity (see figure 2), a virtual oscilloscope (allowing for the parallel observation of time courses) or static and dynamic plots. These visualizations can be applied in parallel to support the learners imagination of what happens to signals on different levels of processing.

4. Educational settings

The general disciplines involved in the field monist seeks to serve are neurobiology, psychology and informatics. In biology, series of experiments during practical trainings are one of the most common educational settings taking place in university laboratories with specific experimental setups. They are the most time expensive part of a degree. Other settings, like classical lectures or seminars, are comparatively less time consuming and their main function is restricted to the provision theoretical background. Therefore, first and foremost, using simulations in biology means their integration in practicals, which is, fortunately, the most active and intense educational setting.

The education in psychology and informatics mainly deals with classical lectures, seminars and tutorials. Contrary to the education of biology, the practical aspects of learning are attached to the theoretical background. Compared with the total time involved chalk and talk is still more common than practical learning, like dealing with educational simulations. This is why educational simulations mainly come along as demonstrations in lectures or tasks in tutorials in these disciplines. However, exercise courses offer appropriate educational settings for a more intense application of simulation also in psychology and informatics.

Educational simulations, as introduced in monist, can be used in almost every learning context found in higher education, i.e. *tutorials and lectures*. It is well suited for a flexible use in any discipline involved. The combination of content-browser and communication system allows the integration of modern educational settings in the traditional ones, as found in neural- and cognitive science. Because of its offline *and* online abilities, the monist console promotes *self-learning*, *blended learning*, pure *online-learning* and forms of *collaborative learning* even over long distances. Precompiled courses, already available in monist, have been tested in the form of tutorials equipped selectively or completely with educational simulations. Pure online-learning has also been tested as it has blended learning with one day presence a week and chat support. But nearly all simulations are also well suited for short demonstrations in lectures. In the meantime, far more than 1000 students came in contact with the monist-system, more than 10 complete courses have been held and several courses have been selectively enriched with educational simulations.

Despite the variety of educational settings applied, one recurred frequently and shall be described as a typical educational setting often used within the monist group: Any simulation is accompanied by tutorial support. After a short theoretical introduction, learners start the simulation and get familiar with the interactive elements provided. This is supported by the instructions included in the educational simulation. After that, the tasks can be solved by writing free text into the units, which lasts 15 min to several hours, according to the given simulation. At any time, questions can be addressed to the teacher. As a control of learning success, a least questions and answers are discussed in the learning group. Additionally, solutions of the tasks written in the educational simulation may be uploaded to the teacher or integrated in a larger record for the whole course.

5. Collection of Contents

Concerning the content, the monist group represents a wide range of disciplines involved in neural- and cognitive sciences. Each participating department represents one topical and one methodological area in its field. This composition ranges from biophysics to neurobiology and cognitive science as well as more formally oriented disciplines like neuroinformatics and robotics. Table 1 gives a general overview.

Topical focus	Methodological focus	Department
Sensory Systems	Experimental Neurobiology	Neurobiology, Bielefeld University
Motor Systems	System Theory	Biological Cybernetics/ Theoretical biology, Bielefeld University
Learning/Behaviour	Experimental Neuroethology	Neurobiology, FU Berlin
Neuronal Coding	Data analysis/statistics	Neurobiology & Biophysics, University of Freiburg
Spatial Cognition/Psychophysics	biol. Robotics, VR	Cognitive Neurobiology, University of Tuebingen
Cognition/Complex Problem Solving	Experimental Psychology	Psychology, University of Bamberg
Neuronal Network Theory	Neuroinformatics & Simulation Techniques	Neuroinformatics, Bielefeld University
Applied Neuronal Networks	»neural engineering«, Robotics	Neuroinformatics, University of Ilmenau
Didactics & Evaluation	Mental Models & Instructional Design	Educational science, University of Freiburg

Table 1: General overview of the topical and methodological fields covered by the members of the monist-group.

Within the scope of the monist-project, 180 ready-made educational simulations have been produced, covering the main disciplines involved. This collection is divided into 8 different more general topics. Table 2 gives a coarse overview of the main contents offered by the monist-system. Apart the category *Methods* every topic describes one specific level of information processing in the brain. The last category deals with those methods that are essential for neuronal modelling.

Neuron	Networks	Sensory Systems	Motor Systems	Behaviour	Learning	Cognition	Methods
Models	Feed-Forward	Visual System	Motor mechanics	Orientation	Non-Associative	Attention	Data analysis
Computation	Recurrent	Auditory System	Reflexes	Communication	Associative	Memory	System theory
Coding	Learning networks	Other Modalities	Pattern Generation	Navigation	Models of Learning	Thinking	Modelling

Table 2: A coarse overview of the main topics, educational simulations have been produced for.

The learning effort per educational simulation ranges from 5 minutes to 5 days. Typically, the effort resides in the range of one to two hours. This collection of educational simulations can freely be recombined to courses or parts of several educational simulations that can be combined to a new one, excluding existing texts. Several courses have been tested and evaluated (see section 6). As precompiled versions, they are provided by the system and can be reorganized according to individual needs. The system provides simulations for every level in higher education, ranging from undergraduate to postgraduate level.

Additional simulation tools

Besides ready-to-use simulations, five different open source simulation tools are accessible via the console. These simulation tools give students the opportunity to learn not only to explore simulations but also to learn *how to build and construct* simulations. They allow for the visual implementation of individual simulations. Table 3 gives an overview. The scientific tools have been adapted for educational purpose. Most of these tools are highly specialized

and will mainly be used in educational context by those departments that designed them for scientific use. Nevertheless, the monist-console offers tutorials and documentations for all users to give them the opportunity of operating these tools in a meaningful manner. All these tools are downloadable at the monist-homepage [3]. Especially for NEO/NST [6] monist is equipped with a plug-in, to view ready-made NEO/NST-simulations like applets. A collection of RUBIN-simulations [7] is integrated as well. All programs mentioned are stand-alone applications and therefore also executable without monist.

Tool	Main purpose	Main topic	GUI	Language	Main users
NEO/NST	Fast prototyping of artificial neural networks	Neuroinformatics	Visual programming	C/C++	Scientist
PSI	Implementation of a theory of human action and cognition	Psychology	Parameter manipulation	Delphi	Scientist, Students
DAS	Artificial Neural network modeller	Psychology	Visual programming	Delphi	Scientist, Students
GraSim	Prototyping of robot control systems	Neural engineering	Visual programming	C++	Scientist
SNNAP	Modelling single neurons & small networks	Neurobiology	System manipulation	Java	Scientists/ Students
RUBIN	Collection of high end educational simulations	Neural information processing	System manipulation	Java	Students

Table 3: Additional tools offered by the monist system.

6. Evaluation

An essential part of the monist-project is the formative and summative evaluation of the use of simulations in higher education. This evaluation is four-fold. It belongs to courses and educational contexts as well as to the educational simulations itself. For continuing application, a questionnaire is attached to each educational simulation and can be filled out quickly. It serves as an information source for a ranking of the educational simulations, offered automatically by the monist-system. If computer-based communication is included into courses by using the monist-talker, all user-action in the talker, if permitted by users, can be logged by the system into XML-files. This enables the analysis of specific user behaviour, to create statistics about the usage of the communication system and its features (favourite rooms, channels, etc.). Moreover, experiments for learner modelling are attached to the project, trying to evolve an experimental framework to facilitate the structure to educational simulations on the level of cognitive psychology. All evaluations still take place. Hence, no results will be presented here. For the future, further information on evaluation results will be available at the monist-homepage.

7. Discussion

The system described so far is a framework for teaching and learning with simulations (and other resources). Its architecture provides a variety of possible combinations of both the structure and use of educational simulations and the possible educational settings, these educational simulations are embedded. The variety of use cases is too broad to be outlined here in detail (one best practice scenario has been described in section 4). The variety reflects the many use cases that have been tested by the monist group members. Therefore, it should be mentioned that monist do not refer to a particular learning theory, because it is too difficult by now to sum up all these use cases into one learning theory. Instead, the use of simulations is motivated by the learning objective to *understand complex and dynamic systems*. Here, the contribution of the monist group member's experiences in educating with simulations is most important. Since most experts involved in this project do basic research on learning it is a significant goal for future work to bundle this expertise for developing paradigms for learning with simulations.

For now, the monist group refers to the use of educational simulations, as they are introduced in this paper. It provides a clear structure and can be combined with many different simulation formats, e.g. applets and applications. Units typically combine instructions and explorative elements. Hence, the explorative character of simulations is the ultimate learning objective but is also guided by instructions. Because of the intuitive interfaces used in most of the monist simulations, instructions will become less important as the user's experience during ongoing learning grows. The less intuitive the embedded simulations are the more detailed the instructions should be. This implies that an effective use of the educational simulations lies in the responsibility of authors and teachers, respectively. To strengthen the efficiency, the system offers recommendations for authors and teachers based on the experiences of the monist group. This includes an educational tutorial, sample units and a help system for general system features.

Teachers and authors from other disciplines, who are interested in designing educational simulations for their lectures with monist, are invited to study the available educational simulations for the neurosciences and to use them as an example work. They surely will benefit from the inspiration coming with the educational simulations in monist.

8. Conclusions and Outlook

The monist console is a system for learning and teaching with educational simulations. It also supports the authoring of educational simulations. Several tools are integrated in the system, that allow for comprehensive introduction of educational simulations in the neural- and cognitive sciences. The offered contents can be regarded as a good practice example, demonstrating the integration of simulations in higher education. Even though simulations in the neural and cognitive sciences are highly specific contents, the monist-console readily supports learning with any kind of educational simulations. The system can be adopted by any discipline, using simulations in science and education. Understanding simulation as educational media [5], monist provides a framework for a broader use and integration in education in general, not only in university context.

Acknowledgements

We are indebted to all the many people involved in the project. They are too many to mention them here. Instead, we like to refer to the monist-homepage, giving access to all participants of the monist-project.

This project has been supported by the Federal Ministry of Science and Education (BMBF) from 2001 to 2004.

Reference List

- [1] Churchland, P.S., Sejnowski, T.J., *The Computational Brain*, MIT-Press, 1992.
- [2] Gazzaniga, M.S., *The new Cognitive Neurosciences: 2nd Edition*, Bradford Book, 1999
- [3] Monist homepage: www.monist.de.
- [4] Crookall, D. (2001). *State of the art and science of simulation/gaming*. *Simulation & Gaming* 32, 449-450.
- [5] Horstmann, W., *Standardizing simulations – uphill all the way*, Campus 2002, eds. Bachmann, G., Haefeli, O., & Kindt, M., pp. 218-230. Waxmann, Zürich.
- [6] Graphical simulation tool kit Neo/NST: http://www.techfak.uni-bielefeld.de/ags/ni/index_projects_d.html
- [7] Lorenz, S., Horstmann W., *RUBIN- Project description, internal report*, Bielefeld University, 1999, available in german at rubin@uni-bielefeld.de.