VideoEasel: Architecture of Virtual Laboratories for Mathematics and Natural Sciences

Sabina Jeschke[1], **Thomas Richter**[1], **Ruedi Seiler**[1]

[1]Berlin University of Technology, Faculty II – Mathematics & Natural Sciences, 10623 Berlin, Germany

"Virtual Laboratories" present a model of interactive, explorative learning tools for supporting lectures, hands-on training, homework assignments and research. First, we prepare a list of requirements that define Virtual Laboratories as a multi-purpose platform for cooperative, explorative learning and, starting from this list, we give specification guidelines on how to design these laboratories.

We propose a three-layer architecture for Virtual Laboratories as networked applications along the formulated principles, and separate laboratories into the following modules: Physical modelling implementing a simulation core, an interconnection layer that allows linkage of various laboratories and computer algebra systems to built-up experiments, and a user interface layer that provides several GUIs targeting different user groups.

Starting from the pedagogical requirements of Virtual Laboratories, we demonstrate the consequences for the implementation on a prototypical lab in the area of statistical mechanics using the proposed software design. Cellular Automata as simulation core provide a rich class of algorithms for implementing microscopic laws of many-body systems. Graphical development toolkits like Oorange are used as interconnection layer. Maple and browser front-ends act as graphical user interfaces. We conclude with an outlook on how to enrich Virtual Laboratories by means of intelligent agents.

Keywords: Virtual Laboratory; Explorative Learning; Mathematics; Cellular Automaton; Cooperative Virtual Knowledge Spaces

1. Background

The relevance of computer-aided visualisation and computation has been rising massively in the recent years. Computer algebra software removes the burden of manual routine computation and computer visualisation that of manual plotting and drawing. However, by the very same process the ability to comprehend new matters quickly and efficiently becomes more important than ever. Due to the massive growth of knowledge, life-long learning is imperative, making it necessary to teach methods for efficient self-controlled learning [5].

Even though many efforts have been made to support and enhance education in the engineering sector by the New Media, only a handful of projects exist that target students of mathematics or physics. Furthermore, several projects are limited to document management applications that present their contents — sometimes enriched by active components like applets — to passive recipients and for that reason are ill-suited to support and encourage self-controlled active learning experiences.

We present here the concept of a Virtual Laboratory that uses the metaphor of a scientific lab to emulate scientific hands-on training within a computer network, thus forming the second main pillar of the education besides classical frontal lectures. Due to the high load of universities, especially in Germany, classical training is either limited by financial or areal capacities, or — as for the field of mathematics or theoretical physics — by not even being in the focus of the education.

Virtual Laboratories build a bridge between the theoretical fields and practical sciences by supplying experiments that run on computer-implemented algorithms that either emulate real devices under idealised situations or represent theoretical concepts. Using multimedia-based eLearning environments these experiments can be made available to all students, making them independent of working hours, financial

Corresponding author: Thomas Richter, e-mail: thor@math.tu-berlin.de, Phone: +4930 3142 7684, Fax +4930 3142 4413

limitations or workstations in classical labs. Last but not least, Virtual Laboratories finally allow an experimental access to abstract phenomena in the theoretical fields.

2. Towards a Pedagogical Profile

One of the defining principles of a Virtual Laboratory is that it does not define “learning units” – instead, it defines “learning spaces” for virtual experiments. The definition of learning goals remains to be given by the lecturer or the leader of the seminar or training course. Virtual Laboratories are tools to achieve a high-quality education *within* a course by providing virtual devices, algorithms, etc. within the specific field. The goal is rather to allow students to develop skills in problem solving and self-controlled learning that are required for their future professional work. Especially laboratories should provide enough freedom for individual and/or unusual experiments outside the limitations set by the curriculum [3]. Thus, a laboratory should allow an explorative learning style and should *teach how to learn* rather than just present concepts.

Virtual Laboratories ought to be able to adapt themselves to the learning style of the individual user. This benefit of modern multimedia technology is required to address the broad audience from undergraduate students to researchers providing highly varying background and goals: Similar to real laboratories, the lab itself provides at first only an infrastructure to setup and control experiments, let it be to enrich lectures by demonstrations, to accomplish electronic exams or to provide a framework for homework or research assignments. The freedom to setup and combine experiments by themselves trains students to solve problems by themselves rather than adapting given solutions, also addressing creativity and by that, increasing the learners’ motivation. The necessity to handle multiple application targets with partially diverging goals has several impacts on the software design we will discuss in the next section.

Due to the high degree of specialisation we find today, scientific results are more and more accomplishments of cooperations between individuals and the outcome of teamwork; this is due to the high complexity of today’s problems that requires the cooperation of experts from various fields. Therefore, *teamwork* has to be actively promoted by Virtual Laboratories as well. By using networked applications, even cooperation across borders should be made available.

Last but not least, a laboratory should integrate into an existing software infrastructure by using standard components from the everyday environment of the working scientist or engineer; this includes products like Maple, Mathematica or Matlab. In the first place these tools provide numerical algorithms to analyse measured data, but using them also allows the students to familiarise themselves with software required for their professional life.

3. Consequences for the Software Design

The pedagogical demands formulated in the previous section show their consequences in the design of a Virtual Laboratory. Since laboratories have to address various targets and interest groups, a highly granular software design is required; this means the separation of the laboratory into components that each fit their respective task and audience best. Furthermore, once the components are equipped with *open* interfaces using *accepted open standards*, the possibilities to combine these components freely to experiments beyond their initial application target and to reuse them outside their initial operational area are gained. Ideally, these experiences should be contributed to an international standardisation group.

For these reasons we promote the following three classes of components:

- **Simulation- and Computation Components:** These components implement the number crunchers in Virtual Laboratories; they do not provide any visible user interface, but rather implement the physical modelling of the entities to be measured. The only kind of interface they provide is that which defines the parameters for the experiment they emulate and that allows to extract the experimental results of that experiment.

- **Connectors:** Connectors are software components that aid the user to combine and link the components of a laboratory to an experiment. In the simplest case, they could be realised by means of a script that extracts measurements from a laboratory kernel as described above, and feeds this data back into a numerical algebra program. Ideally, this kind of linkage should be carried out with a minimum of graphical effort. The second task of the connectors is to translate and adapt the languages and interfaces between distinct laboratories. Even though we should enforce a unique interface definition for all laboratory components, it seems to be unrealistic to achieve this goal in practice, especially when having to deal with components from various sources.
- **User Interfaces:** User interfaces address the needs and goals of the user group experimenting with a Virtual Laboratory. Depending on the application target, a user interface may present a readily set-up experiment for demonstration purposes in a lecture, an applet, an Internet browser, an experiment in a GUI showing an experiment for students in practical training or even a computer algebra system talking to the laboratory kernel through a connector. Thus, in general more than one user interface will be required.

Once the separation into the above classes is understood, it is natural how to address the requirement of supporting cooperative learning scenarios: If we extend interfaces between components to cross machine-boundaries and allow exchange of data in a network, one simulation component running on a server can be observed and investigated by more than one student at once — each measuring from a client at a possibly remote location. Side channels would then allow students to communicate and exchange their experiences. A peer-to-peer network would be an alternative architecture for a distributed Virtual Laboratory. Nevertheless, in our understanding *cooperative learning requires networked applications* [1].

4. VideoEasel — a Virtual Lab for Statistical Mechanics

We now turn to an implementation of a Virtual Laboratory prototype in the above sense, demonstrating the impacts of the above demands on pedagogics and software design. The Virtual Laboratory VideoEasel [8], developed at the DFG research centre MATHEON of the Berlin universities focuses on statistical mechanics, usually lectured in the second semester of the "Mathematical Physics" course. Typically the audience is a mixture of mathematics and physics students.

VideoEasel implements the microscopic rules of physical systems that are of interest in this area, for example the classical Ising Model or Lattice Gas Models, by using so-called *Cellular Automata* [9]. We choose this field because it ideally combines mathematical research and its applications to important problems of natural science and engineering: Probability, analysis and dynamical systems are the prominent disciplines which can be brought to action in an environment of interesting applications, ranging from image denoising to questions of reversibility in statistical systems. Furthermore, cellular automata are on the one hand simple enough to be described by elementary math while on the other hand they show a rich set of complex phenomena. Thus, they invite to experiments without establishing high barriers when entering the field.

According to our demands, VideoEasel is a three-tiered software design separated into a computation kernel implementing the microscopic dynamics of a physical system, an interface/connector layer and several GUI front-ends that allow users to observe and manipulate the experiment. The interface between the kernel and the connector is realised by means of the well-established CORBA middleware, thus making the laboratory a networked application. The kernel is designed to handle several connections at once, even from several users observing the same experiment and therefore supports cooperative and remote learning scenarios.

The microscopic rules are written in a C-like programming language that is compiled and linked to the kernel at runtime; a set of predefined programs implementing various experiments are available on the server, though the user is always invited to modify and change these rules locally if desired. Similar to the experiments, measurement tools exist as microscopic rules defining the physical entities to be observed and measured. Following our granular design philosophy, they can be plugged into each experiment as long as the objects referred to by the measurement tool exist in the object to be measured on.

Several user front-ends are provided: The simplest one is a Java applet which can display an experiment along with some of its parameters in a web browser, thus making it applicable for on-line experiments or quick demonstrations in a lecture. A more complete stand-alone Java GUI hides most of the complexity while making a large subset of the possibilities available to its users, including the attachment of measurement tools and the editing of the microscopic rules. This interface was mainly designed to be used in practical training since it makes readily set-up experiments quickly available for the students.

A sophisticated interface to the Oorange toolkit, a Java programming tool also developed at the TU Berlin, is provided to make VideoEasel applicable to research problems. It represents the objects of VideoEasel, i.e. algorithms and measurement tools as well as parameters controlling its operation as boxes which can be linked together by means of “drag and drop”. As long as a Java interface is available, Oorange can be used to have VideoEasel talk to external applications. It is therefore a classical connector in the sense of the last section.

Last but not least, we also supply an interface to the computer algebra system Maple to analyse the measured data and to control the laboratory from there in order to run even more complex measurement tasks.

5. From Microscopic Dynamics to Algorithms

To shed some light on the way VideoEasel works, let us briefly discuss a classical model of statistical mechanics, the Ising Model [4]. The configuration space is built from a set of spins located on a rectangular lattice. Each spin can be in one of two possible states, classically called "spin up" or "spin down". Within the VideoEasel world, the spin configuration is visualised by a two-dimensional drawing canvas, with spins pointing up shown in yellow and spins pointing down depicted in black. The user front-ends we offer enable the user to manipulate the spin-configuration in a GUI that looks and feels very much like a painting program.

An additional button starts the dynamics of the system; this dynamics is defined by a "microscopic rule" that computes the next state of a spin — thus of a pixel on the screen — by its own state, the states of its neighbouring pixels and the set of external parameters that can also be defined in the GUI. For the model at hand, the dynamics is given by the Metropolis Algorithm [6]. Leaving some technicalities aside, it computes the local energy for each spin and compares this energy with that of the same configuration but with the configuration of the currently considered spin state replaced by its opposite, i.e. with the spin "‘flipped around"’. If the energy of the flipped configuration is smaller than that of the current one, the spin is flipped. The spin flip is performed as well, if the energy difference between new and old configuration is smaller than the energy a "heat bath" contributes. In all other cases, the spin remains in its current state. External parameters control the heat bath as well as an additional energy source representing an external magnetic field.

This algorithm is formulated in a C-like programming language by a program that is conceptionally executed in parallel on each spin. In order to compute the local energy contributions, access to the neighbourhood of a spin is granted. The heat bath is modelled from a random generator of a suitable random distribution, made available as a built-in language intrinsic.

Measurement tools operate on the very same spin configuration, carrying out the measurement process by applying their defining algorithm once for each available spin. In the simplest case of measuring the overall magnetisation of the Ising Model, this algorithm would simply consist of adding 1 to the magnetisation for each spin pointing up and subtracting 1 for each spin pointing down. Similar, though more complex rules can be formulated for other macroscopic observables, e.g. entropy, internal energy or Helmholtz free energy. Both, parameters as well as measurements can be manipulated by the CORBA interface.

Measurement tools and local dynamics need to match each other to make the measuring process meaningful, though. In the case of VideoEasel, this identification is done in the simplest possible way: An automaton assigns names to its configuration, and the measurement process identifies the configuration to measure on by this name.

6. Conclusion and Outlook

VideoEasel is currently still in a prototypical state, only few practical experiences have been gained so far. A school project at the Heinrich Hertz School in Berlin revealed that the technology developed for cooperative learning and teaching is also very useful to help the administrator in providing individual support to students and to demonstrate individual achievements to the class. On the other hand, unrestricted access of each student to the workplace of every other student as it was available in a very first version causes a lot of turbulence in the class room. Therefore a minimal user administration was added. In the long run, VideoEasel should be integrated into a more complete framework in the sense of virtual knowledge spaces [2].

Some early experiments on intelligent assistants have been made; a mini-course about image convolution includes a user-specific feedback mechanism that analyses the failures the user experienced so far and tries to provide custom-tailored solutions for those problems. Given our pedagogical goals, we have to deal with a relatively broad audience. Adapting to the audience by providing several user interfaces of varying complexity is one way of addressing this issue, though user-adapted tutoring by providing a “storyboard” to perform a specific experiment another.

To conclude, let us remark that we are not aiming at replacing frontal lectures or training courses by electronic media; we rather impose a *blended learning* approach: Electronic media will *enrich* traditional courses by providing learning experiences that have not been possible before, and that are more necessary than ever due to the changing demands of education.

Acknowledgements: The support by the DFG (Deutsche Forschungsgemeinschaft) in the framework of the DFG Research Centre Matheon (Mathematics for key technologies) is greatly acknowledged.

References

[1] T. Hampel, R. Keil-Slawik. sTeam - Designing an Integrative Infrastructure forWeb-Based Computer Supported Cooperative Learning. Proceedings of the 10th International World Wide Web Conference, May 1-5, 2001, pages 76-85, Hong Kong, 2001.
[2] T. Hampel, R. Keil-Slawik. sTeam: Structuring Information in a Team - Distributed Knowledge Management in Cooperative Learning Environments. ACM Journal of Educational Resources in Computing, 1(2), 2002.
[3] T. Hampel, R. Keil-Slawik, F. Ferber. Explorations - A New Form of Highly Interactive Learning Materials. Proceedings of WebNet 99 - World Conference on the WWW and Internet, pages 463-468, Charlottesville: Association for the Advancement of Computing in Education, 1999. P. de Bra, J. Leggett (eds.).
[4] E. Ising. Beitrag zur Theorie des Ferromagnetismus. Zeitschrift für Physik, 31:253-258, 1925.
[5] S. Jeschke, M. Kohlhase, and R. Seiler. eLearning-, eTeaching- & eResearch-Technologien - Chancen und Potentiale für die Mathematik. DMV-Nachrichten, Juli 2004.
[6] N. Metropolis, A.W. Rosenbluth, M.N. Teller, and E. Teller. Equations of state calculations by fast computing machines. J. Chem. Phys, (21):1087-1091, 1953.
[7] Mumie community. Mumie. http://www.mumie.net .
[8] Thomas Richter. VideoEasel. http://www.math.tu-berlin.de/~thor/videoeasel/ .
[9] T. Toffoli and N. Margolus. Cellular Automata Machines. MIT Press Cambridge, 1987.