Development of a Virtual Laboratory System for Science Education and the Study of Collaborative Action

Nils Jensen
University of Hanover
Learning Lab Lower Saxony
German Pavilion, Expo Plaza 1
D-30539 Hanover
jensen@learninglab.de

Stefan Seipel
Uppsala University
Dep. of IT
Box 337
SE-75105 Uppsala
stefan.seipel@it.uu.se

Gabriele von Voigt
University of Hanover
Dep. of Distributed Virtual Reality
Schloßwender Str. 5
D-30159 Hanover
vonvoigt@rrzn.uni-hannover.de

Siegfried Raasch
University of Hanover
Dep. of Physics
Herrenhäuser Str. 2,
D-30419 Hanover
raasch@muk.uni-hannover.de

Stephan Olbrich
University of Hanover
Learning Lab Lower Saxony
German Pavilion, Expo Plaza 1
D-30539 Hanover
olbrich@learninglab.de

Wolfgang Nejdl
University of Hanover
Learning Lab Lower Saxony
German Pavilion, Expo Plaza 1
D-30539 Hanover
nejdl@learninglab.de

Abstract: The paper specifies the development of a toolkit to run synthetic science laboratories. The aim was to facilitate collaborative experimenting for problem-based learning in a virtual lab. The goal was to demonstrate virtual experimenting by use of interactive 3D visualization and simulation. Technology was developed over six years and in part designed in explicit accordance to didactic models. For tests, we built a virtual lab that comprised media tools and complex computer simulations, and we evaluated it with promising results. Students used data from meteorology and experimented together. Generally, they enjoyed using the system and collaborated in a motivated way. We identified which tools they preferred. The paper indicates ways to improve the design of virtual labs by use of our toolkit.

Introduction

Network-distributed virtual learning labs are online synthetic spaces and artifacts where (i) experiential distance learning by use of interactive visualization is effective (Trindade et al., 2002), and (ii) computer-mediated communication (CMC) is an adequate means to make students learn together (Sonnenwald et al., 2003). Compared with users of real labs they experiment nearly whatever, whenever and wherever they wish (Emigh, 1998; Youngblut, 1998). The paper specifies the user-centered development of software to develop and run virtual labs. The goal was to show how learning theory drives requirements, and to test installation and use.

Researchers and students used software to study the dynamics of fluids, molecules, and atmospheric convections by use of interactive, user-specified computer simulations and 3D visualization to track multi-dimensional patterns in physical phenomena. The phenomena are complex because they are partly specified by equations that are numerically solvable by the use of many processors and much memory. The results of computations are difficult to perceive and understand. For sense-making we use scientific visualization and virtual reality (VR) to activate tacit knowledge for the inspection of virtual entities (Dede et al., 1999).

Virtual lab systems are toolkits to build virtual labs. However, for higher science education (which we emphasize in the paper), software was either limited to the display of few data (e. g. Churchill et al., 2001; Trindade et al., 2002), rudimentary support for integrated multi-modal online collaboration (e. g. AVS Express at www.avs.com), and availability for proprietary hardware (e. g. Leigh et al., 1997). Our toolkit does not have these restrictions. We specify the influence of didactic principles in development, and report on a pilot study that we carried out with six students and a lecturer (see Nielsen, 1994, chap. 5, on Heuristic evaluation).

Didactics and Design Rationale

The mandate for virtual labs emerges from the theory of constructivist learning to solve problems in education, for example misconceptions by students and the difficulty to adopt rapid change in collective knowledge. In accordance to constructivism, human beings develop knowledge by way of engaging in a situated task (endogenous constructivism) and social negotiation (dialectical constructivism). Constructivists suggest the use of authentic tasks and tools to avoid misconceptions due to otherwise inappropriate simplifications (Duffy & Jonassen, 1992). Complexity challenges students' mental models in a demanding and motivating way. A disadvantage is that few students have access to research material. Even if they had, how would we manage not to overwhelm their perceptual, motor, or – most important – cognitive skills? Initial lack of proper learning attitude and little experience is the problem. A solution is to amplify human capabilities to plan and conduct tasks together, reason about problems in collaborative discourse, and peer-review results via different scaffolds that are provided by teachers (Fretz et al., 2002). Students develop self-confidence and perceive progress in building collective knowledge (Koschmann, 1995). We conclude that virtual labs are adequate tools for the support of constructivist learning if they carry

authenticity: make research tools easy to share

complexity: visualize complex data to promote insight

collaboration: support computer-mediated communication

The approach has didactic implications. The first is that experiments are open-ended. The measure of progress of students is therefore not pre-determined. Second, virtual labs must provide the means to test several computational models; each helps students to supplement and complement [sic] means to understand a physical phenomenon. Third, students must develop mechanisms to cope with complexity. Scaffolds support novices.

The approach has technical implications. Gomez et al. (1995), Kouzes et al. (1996), Bly (1998), Finholt (2001), and Sonnenwald et al. (2003) and other specify systems that implement collaboratories. Jensen et al. (2003) give indicators how to build similar, but virtual, collaboratories by way of improving available toolkits. They classify virtual lab systems and conclude that no system for educational settings implements complex data visualization that helps experts and novices to share data and tools, and no system has customizable facilities for the support of collaboration in accordance to user requirements. The shortcomings limit adaptability to areas of application that are anticipated by developers.

So far, we have reviewed desirable properties of static elements of tools for learning and have seen that they are just partly implemented. Kolb's Experiential Learning Cycle characterizes dynamic properties, in example workflow, in the system that we want to build (Kolb, 1984, p. 42). The Cycle is a theoretical model of learning acts that comprises four iterative and incremental "modes": (i) through apprehension learners perceive sensations; (ii) through comprehension they order sensations; (iii) through active experimenting they extend knowledge; and (iv) through reflective observation, learners realize intention.

Gomez et al. (1995) specify indirectly an example for the usefulness of Kolb's model to design the functionality of virtual labs. In the CoVis project, researchers have developed visualization-based tools for U.S.-American K-12 scholars. The system is a collection of unified Web services for basic group work and data analysis to teach environmental science. Evaluation results show students accept the system, but still there is a problem. Students do not use simulations to match their beliefs and observations against self-chosen hypotheses in an interactive way. This leads to the result that "[…] many students ignore[…] their research finding" (Gomez et al., 1995, p. 23). Note that the coverage of Kolb's "extension" quadrant was missing so we cannot exclude the possibility that a better coverage would have prevented the shortcoming.

The choice for Kolb's model is backed up by more empirical studies. We do not go into more detail but we just state that probably the most severe weakness is the lack of discussion of emotional factors. But we argue they depend on individual settings that we can take into account by use of empirical studies.

Our basic lab design that amalgamates constructivist learning and Kolb's Cycle is that learners

- apprehend authentic, complex phenomena by means of visualization (Christian, 2003);
- comprehend (categorize) what they see through annotations, filters, and references to supplements (Whitelock et al., 2000);
- extend their viewpoint by way of changing simulation results and parameters (experimenting) (Trindade et al., 2002, p. 486); and
- synthesize knowledge during passive observation of material, processes, and actors.

Apart from the concrete result, we have seen that didactic models help developers to design and improve systems for users. However, developers usually do not acknowledge these models to the degree they should.

Technology for the Development of Virtual Labs

CoVASE is a tool for students, researchers, and tutors in the natural and engineering sciences.

Static properties: constructivist learning theory helps us to define infrastructure. Users must achieve *authenticity* through sharing research data and simulations. This means the system must visualize *complex* data from simulations that might or might not run on a student's workstation, but that run (perhaps exclusively) on proprietary laboratory hardware. The solution is to distribute software on networked processors. Simulations generate visualizations on servers, users receive results on clients. Users control the system in accordance to what they see in real-time (other technical strategies exist but are not described). Clients run software that receives data over network, displays graphics, and manages input. A client sends input partly back to the data generator via the server, and partly to other clients for the support of *collaboration*. A technique for inter-client communication is shared memory on one of the clients to which other clients write, and from which other clients read. The approach is simple but not scalable beyond about 10 users, and a single point of failure would close client communication. However, these are minor issues because most self-driven learning groups are small, and they usually use the same infrastructure. Large groups that work simultaneously and autonomously on the same experiment are supported by way of starting several concurrent "central clients" that work on copies of data.

Dynamic properties: Kolb defines classes of actions of learning that help us to predict basic user behavior. Basic support for all quadrants has been implemented, but with emphasis on the visualization and simulation part. (i) Visualization supports *apprehension*. (ii) For *comprehension*, we suggest basic support for the establishment and use of dialogue as a means to find common properties in a series of observations. For this purpose, the system has text-based, vocal, and pictorial chat facilities, symbolic data panels to externalize information, tele-pointers to reference visual entities, and virtual desktops that connect to remote computers to run external applications. (iii) *Extension* is realized through simulation control and the manipulation of the 3D graphics model. (iv) *Intention* is supported by way of capturing client-based events and screen content to recapitulate action.

Implementation: the system comprises a run-time library that displays and manages interface elements. A second run-time library manages data traffic over TCP/UDP/IP (standard Internet protocols for lossless and lossy transfer). Graphics that fit in workstation memory are locally stored; the rest is received part-wise in an efficient format that is comparable to binary VRML (Virtual Reality Modeling Language). Data flow is as follows: the data generator that hosts the computer simulation forwards graphics to the server when intermediary results are complete. The server replicates graphics to clients that connect to a URL (Internet address). Users control a Web browser plugin that displays 3D graphics, and that processes commands from the user and remote plugins. Some commands are sent to the server, and some of them are forwarded to the data generator. RTSP/IP is the generic protocol to manage play out over network. The plugin is used in a Web browser and in the proprietary 3D graphical user interface that is similar to a Web browser to display 3D graphics. Users edit an XML (eXtended Markup Language) file to define virtual lab front-ends. The interface runtime system reads the file. It contains specifications of graphics, remote graphics sources, and causality chains between elements. An example shows code to start a simulation when a button is pressed:

```
...<button1> 3dsim : play : stream </button1>...
<component name="3dsim" type="vpiDocShowVR">
<source>"www.example.edu/3dstream.dvrs"</source> </component>...
```

Extensible elements are the animated visualization and simulation control, virtual video wall, tele-pointer, avatar, and panels for annotation and control. Hardware and Voice-over-IP-lines support high quality audio-visual conferencing with little contingency. A module of the virtual lab module replicates video images from periphery by use of a video grabber card to avoid the use of multiple physical screens.

Heuristic Evaluation of a Virtual Lab in Meteorology

To determine the acceptance of tools, three evaluators observed how participants in two groups worked together on a difficult task in a networked virtual lab. The two groups worked independently at the same time from three offices in the same building. The session lasted three hours.

During the first one and a half hour, users were informed about the general character, importance, and process of the study. They attended a lecture for about 60 minutes about the subject they should work on. They filled out the questionnaire about personnel details, answered two questions about pre-knowledge, and estimated their computer skills. In the following half an hour we guided participants to their workplaces where they started to try out the lab interface. The evaluator instructed them about how to use the system. For the next hour they worked in the virtual lab. They filled out post-questionnaires at the end.

We connected client workstations (Multimedia Personal Computers with MS Windows 2000, Pentium 4 CPU, up to 2,4 Ghz, 512 MB RAM, network card, graphics adapter with 3D acceleration up to Nvidia GeForce3) over

100 MBit/s Intranet (see Figure 1). The headset was an upper standard fabricate. The H.323 audio / video (AV) conferencing units (Polyspan Viewstation SP) were connected via Intranet. The lecturer, an experienced researcher in the computational modeling of phenomena in meteorology, contributed the simulation program and visualization rules. Each group used an embedded virtual video wall in quartered video split mode (resolution 320 by 240, up to 15 frames per second) and G.722 audio. We recorded audio and video by way of integrating a silent endpoint in the conference.

The virtual experiment was an authentic research example from meteorology. It showed the model of an atmospheric convection. The model comprised two differently colored iso-surfaces, four movable colored cutting planes, and colored illuminated streamlines to represent humidity, air flow, temperature, and air pressure. The visualization sequence was independently generated for each group by a remotely controllable computer program on an Onyx 3800 computer with in total 16 processors in use that was in the same building. The simulation generated about two changes per second in the virtual lab. Each scene represented a step in time and comprised ca. 330,230 dynamically composed spheres and triangles at a display rate of 15 to 20 frames per second. The virtual lab contained all elements of the toolkit except virtual computer screens. Participants entered numerical commands in the steering console to change parameters of the simulation. The visualization was in the middle of the scene and surrounded by elements. From each side two of four slices were alternatively hidden to decrease clutter. Students had to identify patterns in movement over time. The list specifies sub tasks:

1. navigation by the use of the avatar, and pointing at models and pressing control panels by mouse,
2. interactive viewing by way of rotating the model by keyboard, and starting (or stopping) the animation,
3. chatting via text panels, voice, and video communication, and watching slides on a virtual projector.

Notes were made on paper. Of course, a study of learning outcome would have involved more, but we were just interested to see if users could perform the necessary basic tasks from a technical point of view.

After the test, we asked participants to fill out the usability questionnaire. Then they judged the quality of each of the 10 elements in checkboxes (“I liked / disliked”). They also indicated how often they had used a tool (Figure 2). After the session, participants received a gift and we raffled 25 Euro.

Six meteorology students volunteered, all were Caucasian. On average they were 24 years old, standard deviation 2. They knew each other and studied in nearly equal semesters (middle level in an undergraduate course). The youngest was 21, the oldest were 26. Three were male. Users were familiar with computers and new to the system. They were untrained to use visualization. They had 15 minutes to practice.

Two female participants were in the first group of three, one in the second group. Each participant had a one-page manual. Each group worked ca. 40 minutes in the virtual lab. The principal evaluator reminded participants to answer questions at the end of the session for the rest of the third hour. Students shared tasks and synchronized action when one tracked down interesting features or needed help. The lecturer supervised action.

Findings

Students accepted the virtual lab. They made progress in collaboratively using the system and developing theories on data characteristics. We collected and analyzed tapes and logs to verify their subjective assessments in post-questionnaires (developed with a psychologist). The participants used audio because it simplified ad-hoc coordination and communication. Participants used the simulation and the 3D model but missed axis labels and did not like to steer the simulation by keyboard because that was “not intuitive”. They had problems with the *visual* complexity of the model because they saw “too many information at once”. One participant asked about the “correct orientation” of the model (although there is none in 3D).

Most testers used the virtual slides projector at least once (“good to read important things once more”). Participants rated the video wall “unimportant for communication” and “distracting”, albeit it was “nice to see others”. The whiteboard was rated “superfluous”. Text chat “was ignored too often, audio [was] better and faster”. Users preferred to communicate via headset to exclude their physical surrounding. A few criticized that controls were occluded and that navigation took more time than in a 2D graphical user interface. Users appreciated use of the tele-pointer by way of pointing along a thought 2D plane because it was simpler than 3D-based navigation. Students liked 3D representations because they gave them a vivid, interactive view on a complex mathematical model. In contrast, participants did not need the video wall. They knew each other and had worked together. They had common ground and a trustful relationship which counterbalanced the need to have a direct visual link to peers.

There were also downsides. We restarted the graphics system (not the simulation) due to a deadlock between colliding screen repainting routines. Luckily, this happened just once, at the beginning of the session. More important was that students were unable to finish more than basic observations in time. The reason was probably our minimalist support for cooperative activities and workflow over time that did not make the conceptual level of collaboration and logic reasoning explicit enough. One solution is to give the tutor a more active, guiding role during the experiment. But this would limit 24/7 access. The alternative is to use software-based scaffolds that condition users to collaborate, plan, measure, and reason for.

Figure 1: Participant. Shown: AV-conferencing unit, virtual lab, headset.

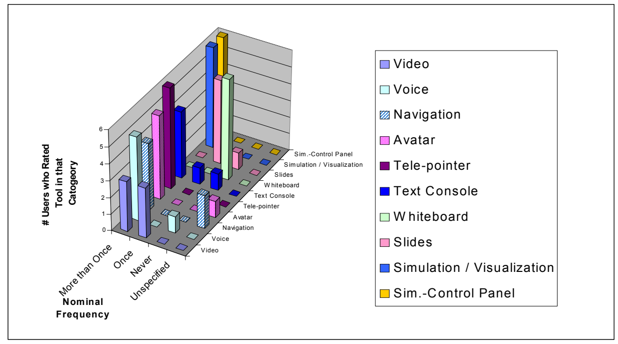

Figure 2: Tool use. Height shows how many users rated the tool (indicated by shade) according to the front labels.

In summary, the emphasis on visual design was right. Users appreciated views that supplemented knowledge from lectures and books and gave peers new ways to socialize with each other. Collaboration was mostly effective.

Our strategy to cover more than one quadrant of Kolb's Cycle was basically right, but stronger support for comprehension, reflection, and planning is required. Also, we will support automatic navigation, visualize collaborative action, and reduce clutter by way of distributing data along different channels of perception to improve usability. Most of these problems are solvable by a careful lab design and do not affect the design of our toolkit. In contrary, the changes can be adopted due to the modular, open structure of the development kit.

Conclusion

The paper has specified the development of software to create virtual labs (see www.learninglab.de/vase3). A controlled case study indicated successful use and room for improvement, mostly in lab design than in toolkit design. Researchers must develop standards for multi-modal user interfaces, scaffolds in collaborative virtual environments, and study learning motivation, efficiency and efficacy under field conditions.

Acknowledgements

We thank F. Block, K. Chmielewski, R. Einhorn, Th. Huk, L. W. Pettersson, C. Richter, and M. Steinke for their help in all stages of the project. VASE 3 was funded by the Ministry for Education and Research (BMBF), the Ministry for Research and Culture (MWK), and partially supported by WGLN.

References

Bly, S. (Ed.) (1998) Special Section on Collaboratories. *Interactions*, May + June (pp. 31-72).

Christian, C. Scientists' Role in Educational Content Development (2003). *Journal of Science Education and Technology 12*, 1 (pp. 31-37).

Churchill, E.F., Snowdon, D. & Munro, A. (Eds.). (2001) *Collaborative Virtual Environments: Digital Places and Spaces for Interaction*. London: Springer.

Dede, C., Salzman, M. C., Loftin, R. B. & Sprague, D. (1999) Multisensory Immersion as a Modeling Environment for Learning Complex Scientific Concepts. In W. Feurzeig and N. Roberts (Eds.), *Computer Modeling and Simulation in Science Education*. NY: Springer.

Duffy, T.M. & Jonassen, D. (Eds.). (1992) *Constructivism and the Technology of Instruction: a Conversation*. Englewood: Lawrence Erlbaum.

Emigh, D. (1998) Scientific Visualization in the Classroom. *Proc. ACM / IEEE Supercomputing '98,* (pp. 1-7).

Finholt, T. (2001) Collaboratories. *Annual Review of Information Science and Technology*, (pp. 73-108).

Fretz, E.B., Wu H.K., Zhang, B.H., Davis, E.A., Krajcik, J.S. & Soloway, E. (2002) An Investigation of Software Scaffolds Supporting Modeling Practices. *Research in Science Education 32*, (pp. 567-589).

Gomez, L.M. & Gordin, D.N. (1995) A Case Study of Open-Ended Scientific Inquiry in a Technology-Supported Classroom. *Proc. AI-Ed '95*, (pp. 17-24).

Jensen, N., Seipel, S., Nejdl, W. & Olbrich, S. (2003) CoVASE – Collaborative Visualization for Constructivist Learning. *CSCL Conference 2003*, (pp. 249-253).

Kolb, D. (1984) *Experiential Learning: Experience as the Source of Learning and Development*. Eaglewood Cliffs: Prentice-Hall.

Koschmann, T. (Ed.) (1995) *CSCL: Theory and Practice of an Emerging Paradigm*. Hillsdale, NJ: Lawrence Erlbaum.

Kouzes, R.T., Myers, J.D. & Wulf, W.A. (1996) Collaboratories: Doing Science on the Internet. *IEEE Computer 29*, (pp. 40-46).

Leigh, J., Johnson, A.E. & DeFanti, T.A. (1997) Issues in the Design of a Flexible Distributed Architecture for Supporting Persistence and Interoperability in Collaborative Virtual Environments. *Proc. ACM / IEEE Supercomputing '97*, (pp. 1-14).

Nielsen, J. (1994) *Usability Engineering*. San Francisco, CA: Morgan Kaufman.

Sonnenwald, D.H., Whitton, M.C. & Maglaughlin, K.L. (2003) Evaluating a Science Collaboratory: Results of a Controlled Experiment. *ACM Transactions on Human-Computer Interaction 10*, 2, (pp. 150-176).

Trindade, J., Fiolhais, C. & Almeida, L. (2002) Science Learning in Virtual Environments. *British Journal of Educational Technology 33*, 4, (pp. 471-488).

Whitelock, D., Romano, D., Jelfs, A. & Brna, P. (2000) Perfect Presence: What Does This Mean for the Design of Virtual Learning Environments? *Education and Information Technologies, 5*, 4, (pp. 277-289).

Youngblut, C. (1998) Educational Use of Virtual Reality Technology. *Tec. Report*. Inst. Defense Analyses, US.