Elektrotechnik & Informationstechnik (2008) 125/5: 190–196. DOI 10.1007/s00502-008-0537-9

Design, development and evaluation of online interactive simulation software for learning human genetics

A. Holzinger OVE, W. Emberger, S. Wassertheurer, L. Neal

Objective: In this paper, the authors describe the design, development and evaluation of specific simulation software for Cytogenetics training in order to demonstrate the usefulness of computer simulations for both teaching and learning of complex educational content.

Background: Simulations have a long tradition in medicine and can be very helpful for learning complex content, for example Cytogenetics, which is an integral part of diagnostics in dysmorphology, syndromology, prenatal and developmental diagnosis, reproductive medicine, neuropediatrics, hematology and oncology.

Methods and materials: The simulation software was developed as an Interactive Learning Object (ILO) in Java2, following a user-centered approach. The simulation was tested on various platforms (Windows, Linux, Mac-OSX, HP-UX) without any limitations; the evaluation was based on questionnaires and interviews amongst 600 students in 15 groups.

Conclusion: This simulation has proved its worth in daily teaching since 2002 and further demonstrates that computer simulations can be helpful for both teaching and learning of complex content in Cytogenetics.

Keywords: simulation; interactive learning; e-learning; simulation-based learning; object-orientated technology

Design, Entwicklung und Evaluierung einer internetfähigen Simulationssoftware für den Einsatz im Medizinstudium.

Zielsetzung: In dieser Arbeit beschreiben die Autoren Design, Entwicklung und Evaluierung einer internetfähigen Lernsoftware zur Karyotypisierung für den Einsatz im Medizinstudium. Dabei wird auch der Frage nachgegangen, ob Computersimulationen den hohen Anforderungen bei der Vermittlung komplexer Inhalte gerecht werden können. Es wird gezeigt, dass dieser Ansatz sowohl für Lernende als auch für Lehrende Mehrwerte bringt und den traditionellen Methoden in diesem Bereich überlegen ist, wenn die Simulation didaktisch richtig eingesetzt wird.

Hintergrund: Simulationen haben in der Medizin eine lange Tradition und erweisen sich insbesondere dann als hilfreich, wenn es darum geht, hochkomplexe Zusammenhänge verständlicher darzustellen, wie dieses Beispiel aus der Zytogenetik, einem Spezialgebiet der Humangenetik, zeigt. Die zytogenetische Diagnostik beschäftigt sich mit reproduzierbaren strukturellen und numerischen Veränderungen der menschlichen Chromosomen und kommt in der Dysmorphologie, Syndromologie, Pränatal- und Entwicklungsdiagnostik, Reproduktionsmedizin, Neuropädiatrie, Hämatologie und Onkologie zum Einsatz.

Material und Methoden: Die Simulationssoftware wurde als interaktives Lern-Objekt (ILO) entwickelt. Als Entwicklungsumgebung wurde die Java2-Plattform gewählt; die Entwicklung selbst erfolgte nach den Grundsätzen des User-Centered-Design. Die Software wurde Plattform-unabhängig ausgelegt und konnte auf verschiedenen Systemarchitekturen (Windows, Linux, MacOSX, HP-UX) erfolgreich und ohne Beschränkungen getestet werden. Die durchgeführte Evaluierung basierte auf Fragebögen und Interviews mit rund 600 Studenten in 15 Gruppen.

Schlussfolgerungen: Die Simulationssoftware hat ihre Alltagstauglichkeit seit 2002 im Lehrbetrieb unter Beweis gestellt. Anhand der Evaluierung konnte an diesem Beispiel gezeigt werden, dass diese Lernsoftware sehr gut geeignet ist, den diagnostisch-analytischen Prozess in der Zytogenetik anschaulicher zu vermitteln als die traditionelle papierbasierte Methode. Insbesondere wirkt sich die Einbettung in ein integratives Unterrichtskonzept positiv, sowohl auf Lernende als auch auf Lehrende, aus.

Schlüsselwörter: Simulation; interaktives Lernen; E-Learning; simulationsbasiertes Lernen; objektorientierte Architektur

Received March 31, 2008, accepted April 9, 2008

1. Introduction

Learning must be seen as an *active process of knowledge construction* (*Holzinger, 2002a*); consequently, e-learning environments ought to enable learners to perform in an interactive process by solving specific problems (*Alessi, Trollip, 1991*). Computer simulations are effective in many learning situations to support knowledge construction (*Alessi, Trollip, 1991*), (*Schank, 1994*). They are especially helpful in Medicine and Health Care (*Simon, 1972*), where they have been used as one of the earliest applications of computers in Medicine (*Wigton, 1987*).

The Human-Computer Interaction in such environments is usually realized through the visual representation and manipulation of objects in a Windows environment in order to simulate a realistic situation. The visual representation of objects must have a look and feel of their real-world counterparts. The behavior of these objects is modeled and simulated using object-oriented technology combined with specific

Holzinger, Andreas, Univ.-Doz. Ing. MMag. Dr., Institute for Medical Informatics, Statistics and Documentation, Research Unit HCI4MED, Medical University Graz, Universitätsplatz 3, 8010 Graz, Austria; **Emberger, Werner, Dr.,** Institute for Medical Biology and Human Genetics, Medical University Graz, Austria; **Wassertheurer, Sigi, Mag.,** Austrian Research Centers, Department for Biomedical Engineering, Austria; **Neal, Lisa, Ph.D.,** Tufts University School of Medicine, USA
(E-mail: andreas.holzinger@meduni-graz.at)

rules. The traditional approach has been that skills and concepts are learned through the manipulation and simulation of objects in the graphical user interface (*Jiang, Sittig, 1995; Wigton, 1987*).

In this paper we describe the design and technological development of a specific simulation software, which is referred to as CYTOTRAINER, since the specific topic where it is applied is Cytogenetics training. We describe our chosen medical topic and how and why it was chosen and we aggregate the challenges that arose to well known simulation categories.

2. Background and related work

As our medical topic we have chosen the field of Cytogenetics, wherein cytogenetic analysis is an integral part of diagnostics in dysmorphology, syndromology, prenatal and developmental diagnosis, reproductive medicine, neuropediatrics, hematology and oncology. Following the premise that learning is an active process of knowledge construction, our learning environment must enable learners to actively solving a problem in the context of authentic research in genetics (*Gelbart, Yarden, 2006*).

Traditionally, cytogenetic analysis is represented in the curriculum in a practical exercises using photographic material of microscope g-banded metaphases. Each student was given a photograph of a normal and a pathological metaphase and had to cut out the chromosomes manually to re-arrange them in the form of a karyotype (Fig. 1).

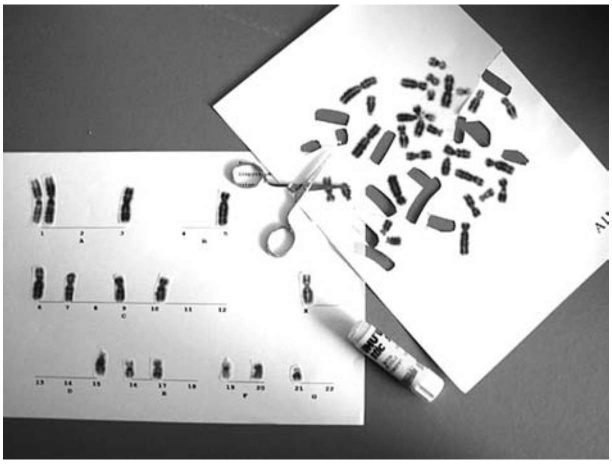

Fig. 1. The non-electronic conventional exercise: Using paper and scissors

In the routine analysis, this manual procedure has been replaced by computer based analysis tools that have been designed to fulfill the complex requirements of routine analysis. However, these sophisticated tools are complex and expensive, therefore definitely not of practical use for educational purposes. Consequently, the idea has emerged of developing an online simulation tool, with a strong focus on pedagogical and didactical benefit.

Basically, research has found a positive, functional relationship with learning outcomes arising from the use of simulations in medical education (*Weller, 2004*). From a psychological perspective, simulations can be seen as dynamic media and research on the impact of dynamic versus static media on learning performance has a long tradition and is consequently rich in empirical results: Past research revealed that the mode of presenting learning contents significantly affects learning processes and, as a result, learning performance (*Mayer, 2001*).

Often it was assumed that dynamic media might be the most successful method for presenting learning content about complex dynamic systems and that such dynamic media might significantly facilitate learning, however, past research found contradictory results (*Davies, 2002*; *Tversky, Morrison, Betrancourt, 2002*); this is possibly due to the fact that medical educational research is inconsistent and varies widely in methodological rigor and substantive focus (*Issenberg et al., 2005*). The existing community of research recommends a cautious use of such media (*Rieber, 1991; Schnotz, Grzondziel, 1999; Mayer et al., 2005; Mayer, 2005*). Dynamic media is only successful in facilitating learning in comparison to traditional static media such as texts or images, when they are able to (1) reduce the cognitive load, which is necessary to comprehend them, (2) serve to generate mental models of a concept and, consequently (3), offer visualizations that correspond to a meaningful mental model; most of all, dynamic media must be attuned to learners' experience, expertise, and previous knowledge. Therefore, material containing dynamic media must avoid information, animations, and elements, which are not necessary to comprehend a concept (*Holzinger, Kickmeier-Rust, Albert, 2008*). However, there is still a great deal of future research and development needed to advance both technological and psychological aspects in the area of simulation-based learning (*McGaghie et al., 2006; Cook, 2005*).

3. Methods and materials

3.1 Structural aspects

From the beginning of this project we identified a strong need for a strict classifying procedure for the produced digital content. Past experience made us aware of the importance of investing additional time in content persistence strategies. The pure amount of data involved when digitizing the entire curriculum brings with it inherent challenges.

Counter to what one might expect, the biggest problem is not saving the data per se, but rather retrieving it. How does one retrieve the exact piece of data being sought with a semantic description? Another issue is how to organize and coordinate the treatment of inhomogeneous types of learning material as well as the various authors. These issues are those of the content producer, but other arise for the student, in particular, what a student can expect from a certain piece of content, what kind of preparations are necessary to understand the offered learning material, and how will the student know if he or she has used the material effectively. To answer all these questions appropriately, we determined a rigid framework for the organization of the content. This framework is based on the IEEE LOM standard (*Holzinger, Kleinberger, Müller, 2001*). Consequently, the software was developed as a Learning Object (LO) which covers one atomic learning unit, corresponding to a 45 minute classroom session, whereby the LO is organized into four subsections:

- Content
- Metadata description
- Previous knowledge assessment (Q/A)
- Final knowledge assessment (Q/A)

Each content subsection contains the classical learning material, prepared for interactive online use. The metadata paragraph is key to the navigation and reuse of learning content. The content is classified by various properties. These metadata describe the content, format, purpose and structure of data and over the past years, the IEEE-LOM standard has dominated the metadata world in e-learning applications, however, with the advent of the Semantic Web, e-learning applications are beginning to evolve their metadata representation from these standards by adding semantic structure or by converting entirely to semantic representations of structure, consequently we are facing a shift in this respect (*Hend, Hugh, 2006*).

To provide reflection opportunities to the learner, self assessment features were incorporated in each learning object. Self assessment is of vital importance for learning and is very much appreciated by the students (*Holzinger, Motschnik-Pitrik, 2005*; *Ebner, Holzinger, 2007*).

The questions and answers are relevant to the exams, which are held conventionally rather than online. We distinguish among entrance and end term assessments, but in higher semesters the exit questions of prior courses will be linked to entrance conditions. To enable such a reuse of questions in an easy way, the above meta-data is needed to close the cycle.

3.2 Portability

For portability, we need to support software standards which are platform independent. Efforts to create platform independent software have resulted in HTML, XML and SOAP (*W3Org, 2003*) and JAVA (*Sun, 2003*), which are used to create high-quality platform independent material. The decision to use HTML and standard extensions was based on the wide availability and use of standard browsers.

Additional software was added to seamlessly embed additional interactive features through ''plug-ins'' (*Netscape, 2003*) to provide a set of standard functions to communicate with the browser. Popular candidates are FLASH and SHOCKWAVE (*Macromedia, 2003*) or JAVA (*Sun, 2003*).

We used JAVA2 technology. Special tags in the invocation signature of each applet avoid the use of a virtual machine with inappropriate features. Although the JAVA language is standardized, all tangible implementations need to be certified by SUN Microsystems due to the ongoing dispute between Microsoft and Sun Microsystems concerning Microsoft's implementation of JAVA. We tested our simulations on various certified JAVA2 platforms (Windows, Linux, Mac-OSX, HP-UX) without any limitations.

Furthermore, since the second major release of the JAVA2 platform (Version 1.3), the significant drawbacks have been abolished. The repaint code was revised and the SWING library is now an integrated part of the virtual machine.

3.3 Usability

What compatibility means in a technical sense can be compared to usability in a human sense (*Holzinger, 2002b*). Subsequently, our software was designed in a way that our target group can utilize it instantly with their current platform. We wanted to meet both optimal functional requirements and intuitive graphical user interface design known from standalone desktop computer systems. We reached that by a strong commitment to User-centered Design and incorporating basic usability engineering methods, such as rapid prototyping, during the design and development phase (*Holzinger, 2005; Holzinger, 2004*).

3.4 Software design

The requirements to online software are basically the same known from standard desktop software. However, the common problem with standard software is its size. If we want to reach a wide range of users, we need to keep in mind that some users would have low bandwidth and hence slow download.

Based on our experience within this project, we developed three rules for producing, what we call, browsable code:

1. Store nothing, compute everything! In online applications, CPU time is almost never the bottleneck of the system, but rather bandwidth. Therefore every dependent state should be calculated and not stored.
2. Design your software modularly! Due the fact that everything cannot be computed, some information and functionality need to be stored. To avoid huge executable files you have to design your software modularly; so that portions of code can be loaded when necessary (*Wilson, Kesselman, 2000*). An object oriented design (*Reed, 2002*) approach may help significantly.
3. Don't use large utility libraries which you don't know! When developing standard applications, there is no question about using predefined libraries from third parties (e.g. see (*Hibbard, 2002*)). In online applications size is everything. In most cases self implemented algorithms from standard literature (*Hibbard et al., 2002*) or free available codes (*Buss, 2001*) fulfill these requirements quickly and with minimum size. In addition, the effort is appreciable because there is normally no need for generalization.

Our experience has shown that by following these brief guidelines, slim (80–250 KB), yet full featured interactive online applications can be realized.

3.5 System implementation

In the field of simulation there are generally three main areas:

- Continuous Time Simulation
- Discrete Event Simulation
- Visual Simulation

Continuous Time Simulation (*Cellier, 1991*) describes the behavior of systems over time by the means of differential equations. These methods are normally used to describe smooth processes e.g. in flows within the human body or chemical reactions. To solve such systems of differential equations, algorithms for numerical integration and optimization have to be applied.

Discrete Event Simulation (*Buss, 2001; Schriber, 1991*) uses servers, queues, random numbers and event-lists to simulate ambulance logistics. In the field of Visual Simulation we do not simulate the process itself, we simulate the process' view and the user interacts with the provided view to control the process itself. These techniques are also known as Virtual Reality and Augmented Reality. To achieve this we need professional 2D/3D computer graphics to simulate virtual views of microscopy or endoscopy.

For our implementation we developed a long list of requirements which we summarize below:

- Platform independence
- Support for GUI
- Support for higher mathematics
- 2D and 3D interactive graphics support
- Object-oriented, slim and modular

It is difficult to find a single platform that meets all these requirements. We chose the JAVA2 platform licensed by Sun for a number of reasons. There is comprehensive platform support for both users and developers. There is one valid standard licensed by SUN Microsystems but different implementations form several companies and communities for all major and minor operating systems. The software use and even visual development tools are available free of charge. There is generic support for state of the art GUI (*Walrath, Campione, 1999*) higher mathematics (*Buss, 2001*) as well as 2D/3D graphics (*Selman, 2002*). The main advantage of JAVA, however, is that the language was designated for online applications from the beginning. Therefore it is very easy to produce slim code (*Bloch, 2001*) and the object-oriented design (*Reed, 2002*) as well as the generic class-loading mechanism (*Wilson, Kesselman, 2000*) allow a very high level of modularity.

3.5.1 Functional aspects

The simulator we developed provides both a full featured visual and functional reconstruction of a cytogenetic analysis process. The first step in a classical analytic process is the search for metaphases in a microscope. Therefore the simulator supplies a virtual microscopy window 100 where the user can move on the virtual slide to search for metaphases. Once a metaphase is found it has to be moved to the centre of the window (Fig. 2).

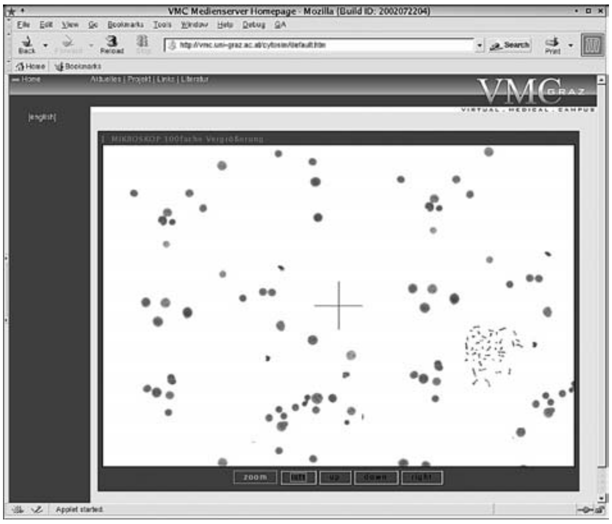

Fig. 2. The user can move on the virtual slide to search for metaphases

At this point the user reaches the next stage in analysis. It is necessary to count the chromosomes and assign them to a certain karyotype. To achieve this, the simulator makes it possible to zoom into the microscopy window by a factor of 1000 for detailed analysis (Fig. 3).

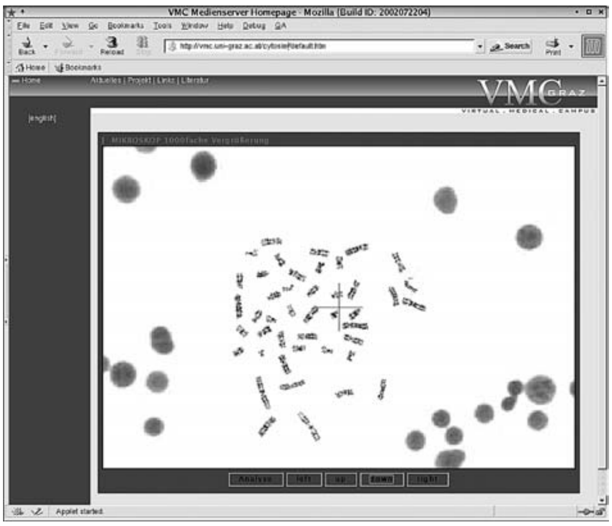

Fig. 3. Zooming into the microscopy window by a factor of 1000 for detailed analysis

After the user has counted the chromosomes, they may switch to the analysis section. The analysis window contains the capture area, the karyotyping area and the analysis line. In the capture area we can see the captured metaphase; when a chromosome is clicked it appears in the analysis line and the chromosome in the capture area is highlighted to indicate that it has been selected for analysis. In the analysis line, chromosomes can be rotated to the correct position using the rotate buttons. After the chromosome is rotated it can be moved to the karyotyping area by assigning the estimated position on the karyotype.

This procedure is done for all the chromosomes. When the karyotyping is done, all chromosomes that are in the wrong position are removed from the karyotype and the highlighting of the wrong chromosomes in the capture area disappears. The analysis has to be redone for the chromosomes which were in the wrong position. If the karyotype is finally correct, the karyotype formula has to be written in a dialog field. If the karyotype formula is correct, the link page appears (Fig. 4). This page refers to other LOs which contain cytogenetic basic information and clinical descriptions of chromosomal syndroms and naturally the final knowledge assessment.

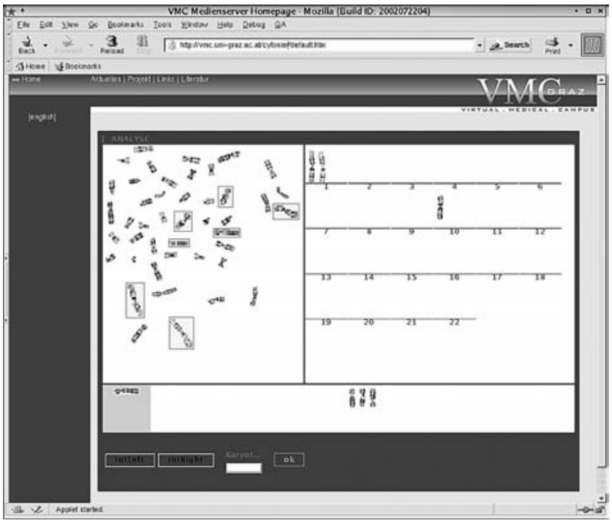

Fig. 4. If the karyotype formula is correct the link page appears

4. Lessons learned

4.1 Web simulation

In contrast to our solely client orientated, pure JAVA method, there are basically two other technical approaches to web simulation:

a) Server-side based solutions (*Straßburger et al., 1998; Straßburger, Schulze, 2001*); and
b) Client-side combined web/native solutions (*Schmid, 1997*).

The bottleneck of approach a) is that all computations are made on central servers. That means that both, the CPU capacity as well as network bandwidth have to be shared with all other users.

If there is immanent and heavy need for example of

1. Interactivity
2. Visualization data
3. Numerical solvers

the resources of central services are quickly overloaded. On the other hand, the PCs of the end users are only used as simple remote terminals.

In approach b) which is more similar to ours, the idea is to simply download the model and to keep the simulator local. That means that the browser being used holds the web based course information like in the learning management system but communicates via a plug-in with a locally installed simulation engine to solve the given exercise. The disadvantage of this method is that in addition to a plug-in, like in our approach, additional installed simulation engines are also needed on each client computer. This requires great effort regarding license fees and computer administration to keep the system going.

Our approach delivers the convenience of an *all-in-one offer for asynchronous learning.* Related work on similar object-oriented simulation technologies include e.g. (*Kilgore, 2002*; *Reichenthal, 2002*; *Jacobs, Lang, Verbraeck, 2002*).

4.2 Evaluation findings

Beginning with the Winter semester 2002 the Cytotrainer was used within Module 3 of the new curriculum of the Medical University in Graz with 600 students in 15 groups. Along with the practical use we carried out an evaluation. As method for this qualitative evaluation we used interviews and questionnaires.

4.2.1 Practical use/handling

Firstly, the use of the program proved to be relatively easy. No significant problems have been reported, neither by teachers nor by students, independently of previous computer experience. Secondly, the teachers reported consistently that it was easy to teach by using the Cytotrainer. One leaves the search of the chromosomes to the students and provides the individual background knowledge for each case solved by them. This sets up a supplementary clinical reality referral.

4.2.2 Advantages

The biggest advantage for our students was that the simulation exposes them to a considerably more realistic scenario. The classical method does not correspond to modern standards and is therefore only used for special question formulation. This circumstance is also obvious to the students because they cannot imagine that modern chromosome analysis could be carried out by cutting out photographs! However, the application of analysis software can be counterproductive when the bulk of time would be needed to teach the students the handling of the special programs instead of the content.

Furthermore, the Cytotrainer also offers the advantage that the exercises can be executed before or after instruction. Students reported that this gives them the opportunity to prepare for the class or to implement the basic knowledge offered during the class with a deeper study of the material on their own. Finally, we applied a component of fun, which increases the challenge to the student to complete the given tasks.

During our experiments, we were able to detect an increase in motivation (questionnaires in a one-shot before-after scenario). We were also able to determine a good acceptance. The majority of teachers and students were very satisfied.

It appears that the analysis of two or three mitosis is sufficient. Therefore, the time that would be needed for further analysis can be shortened considerably. The students who practiced before class were able to complete their tasks much faster. This fact is recognized by, and communicated to, other students, who in turn took advantage of the practice possibilities. The extra time available can be used for interesting question and answer periods and a deepening communication of knowledge.

4.2.3 Disadvantages and risks

The most obvious disadvantage was, naturally, that we were dependent on a corresponding computer infrastructure in contrast to the previous paper and scissors exercises (ref. to Fig. 2).

The slim architecture and the highly specific task resulted in the disadvantage of limited flexibility. The learning object is suitable for chromosome counting, however, it is not capable of doing anything else. A risk could therefore be in overestimating the technology.

Currently, it is only possible for the students to ask questions during the traditional classroom instruction. Independent work with the learning object, for example at home, proved to be less effective. The available cases display of course only an arbitrary selection of exemplary aberrations. They do not portray a representative spectrum of the patients. Each case only contains three mitoses. Due to the fact that the chromosomes are always rearranged, it is almost certain that the students will not be able to learn the position of the individual chromosomes by heart, but three mitoses is nevertheless quite a low number. These do not provide under any circumstances a sufficient number of mitoses for the assessment of a patient.

Without accompanying instruction, the students are hardly given the chance to gain information that helps them to gain a deeper understanding of the topic. The instruction must provide background information for both clinical genetics, as well as for chromosome preparation. Therefore, the teacher will by no means be replaced, but will continue to play an important role in the instruction of the student.

5. Conclusion

Our work provided insights in the design, development and handling of an Interactive Learning Object (ILO). We realized that 45 minutes of (didactical) time to work with such an object is the absolute maximum. This fills exactly one lecture-hour at an atomic level within the new curriculum.

From the software design perspective we localized three facts which are vital and we subsume these under rules:

1. Store nothing, compute everything;
2. Design the software modular;
3. Do not use large utility libraries, which you do not know.

Concerning the use of the simulation, we found that in our case there are 3 metaphases in different image qualities and different arrangements of its chromosomes viewable per virtual experiment. If we would have stored some relevant scenarios, each scenario would have images with a size of about 0.5 MB. Through randomized just in placement and online manipulations of the images, we could reduce the required amount of storage down to 150 KB. There are currently about 15 different virtual experiments available and steadily increasing. Currently, this would lead to size of about 1.5 MB. By partitioning the system into sections of code and separate experiments by the means of software generalization and appliance of selective class-loading mechanisms we got portions of 70–150 kB which we reload on demand by background threads. To realize the virtual views we needed some numerical procedures for image manipulation. Normally, one would use prepared libraries provided from the market. We decided to use the rich standard graphics programming interface and implemented the missing features manually. This maybe was not the fastest but most sustainable method for our users. Focusing on this resulted in only 70 KB of program code.

From the Human–Computer Interaction perspective we were able to prove that the handling was easy for both teachers and students. From the didactical-pedagogical perspective we could show that the actual learning object deals with the relation between the theory already learnt and its context within a medical environment. Experiments with 600 students in 15 groups showed a high acceptance and an increase in motivation using this method, mainly due to the fact that the Cytotrainer exposed them to a considerably more realistic scenario than the classical scenario using paper and scissors.

For the future, an interesting question is on how to extend the simulation software in an quasi-intelligent way, providing special guidance, since unguided inquiry is generally found to be an ineffective way of learning (*de Jong, 2006; Klahr, Nigam, 2004; Mayer, 2004; Holzinger, Kickmeier-Rust, Albert, 2008*). Psychological findings must be integrated in technological development on systemic level.

References

Alessi, S. M., Trollip, S. R. (1991): Computer-Based Instruction: Methods and Development, 2nd ed. Englewood Cliffs (NJ): Prentice Hall.

Bloch, J. (2001): Effective Java Programming Language Guide. Reading (MA): Addison Wesley.

Buss, A. (2001): Discrete Event Programming with Simkit. Simulation News Europe Issue, 32/33: 15–25.

Cellier, F. E. et al., (1991): Continuous System Simulation. Berlin: Springer.

Cook, D. A. (2005): The research we still are not doing: An agenda for the study of computer-based learning. Academic Medicine, 80 (6): 541–548.

Davies, C. H. J. (2002): Student engagement with simulations: a case study. Computers & Education, 39 (3): 271–282.

de Jong, T. (2006): Computer Simulations: Technological Advances in Inquiry Learning. Science, 312 (5773): 532–533.

Ebner, M., Holzinger, A. (2007): Successful Implementation of User-Centered Game Based Learning in Higher Education – an Example from Civil Engineering. Computers & Education, 49 (3): 873–890.

Gelbart, H., Yarden, A. (2006): Learning genetics through an authentic research simulation in bioinformatics. Journal of Biological Education, 40 (3): 107–112.

Hend, S. A.-K., Hugh, C. D. (2006): The evolution of metadata from standards to semantics in E-learning applications. Proc. of the 7th Conf. on Hypertext and hypermedia. Odense, Denmark, ACM.

Hibbard, W. (2002): Building 3-D User Interface Components Using a Visualization Library. Computer Graphics, 36 (1): 4–7.

Hibbard, W., Rueden, C., Emmerson, S., Rink, T., Glowacki, D., Whittaker, T., Fulker, D., Anderson, J. (2002): Java distributed objects for numerical visualization in VisAD. Communications of the ACM, 45: 4.

Holzinger, A. (2002a): Multimedia Basics, Vol. 2: Learning. Cognitive Fundamentals of multimedial Information Systems (www.basiswissen-multimedia.at). New Delhi: Laxmi.

Holzinger, A. (2002b): Multimedia Basics, Vol. 3: Design. Developmental Fundamentals of multimedial Information Systems. New Delhi: Laxmi.

Holzinger, A. (2004): Application of Rapid Prototyping to the User Interface Development for a Virtual Medical Campus. IEEE Software, 21 (1): 92–99.

Holzinger, A. (2005): Usability Engineering for Software Developers. Communications of the ACM, 48 (1): 71–74.

Holzinger, A., Motschnik-Pitrik, R. (2005): Considering the Human in Multimedia: Learner-Centered Design (LCD) & Person-Centered e-Learning (PCeL). In: Mittermeir, R. T. (ed): Innovative Concepts for Teaching Informatics. Vienna: Carl Ueberreuter, 102–112.

Holzinger, A., Kleinberger, T., Müller, P. (2001): Multimedia Learning Systems based on IEEE Learning Object Metadata (LOM). ED-Media World Conf. on Educational Multimedia, Hypermedia and Telecommunications, Tampere (Finland), AACE, 772–777.

Holzinger, A., Kickmeier-Rust, M., Albert, D. (2008): Dynamic Media in Computer Science Education; Content Complexity and Learning Performance: Is Less More? Educational Technology & Society, 11 (1): 279–290.

Issenberg, S. B., McGaghie, W. C., Petrusa, E. R., Gordon, D. L., Scalese, R. J. (2005): Features and uses of high-fidelity medical simulations that lead to effective learning: a BEME systematic review. Medical Teacher, 27 (1): 10–28.

Jacobs, P. H. M., Lang, N. A., Verbraeck, A. (2002): *D-SOL*. A Distributed Java-based Discrete Event Simulation Architecture. 2002 Winter Simulation Conf., San Diego (CA), Society for Computer Simulation International (SCS), 793–800.

Jiang, M. Z., Sittig, D. F. (1995): Developing interactive computer-based simulations: an object-oriented development methodology enhances computer-assisted instruction. Computer Methods and Programs in Biomedicine, 47 (3): 189–196.

Kilgore, R. A. (2002): Object-oriented simulation with Java, Sild and OpenSML.NET languages. 2002 Winter Simulation Conf., San Diego (CA), Society for Computer Simulation International (SCS), 227–235.

Klahr, D., Nigam, M. (2004): The equivalence of learning paths in early science instruction – Effects of direct instruction and discovery learning. Psychological Science, 15 (10): 661–667.

Macromedia (2003): Flash. Online available: http://www.macromedia.com/software/flash.

Mayer, R. E. (2001): Multimedia learning. Cambridge (UK): Cambridge University Press.

Mayer, R. E. (2004): Should there be a three-strikes rule against pure discovery learning? The case for guided methods of instruction. American Psychologist, 59 (1): 14–19.

Mayer, R. E. (ed) (2005): The Cambridge Handbook of Multimedia Learning, Cambridge (MA): Cambridge University Press.

Mayer, R. E., Hegarty, M., Mayer, S., Campbell, J. (2005): When Static Media Promote Active Learning: Annotated Illustrations Versus Narrated Animations in Multimedia Instruction. Journal of Experimental Psychology: Applied, 11 (4): 256–265.

McGaghie, W. C., Issenberg, S. B., Petrusa, E. R., Scalese, R. J. (2006): Effect of practice on standardised learning outcomes in simulation-based medical education. Medical Education, 40 (8): 792–797.

Netscape (2003): Netscape Gecko Plugin API Manual.

Reed, P. R. (2002): Developing Applications with Java and UML. Reading (MA): Addison Wesley.

Reichenthal, S. W. (2002): Re-introducing web-based simulation. 2002 Winter Simulation Conf., San Diego (CA), Society for Computer Simulation International (SCS), 847–856.

Rieber, L. P. (1991): Animation, Incidental Learning, and Continuing Motivation. Journal of Educational Psychology, 83 (3): 318–328.

Schank, R. C. (1994): Active learning through multimedia. IEEE Multimedia, 1 (1): 69–78.

Schmid, C. (1997): New Trends in Control Engineering Education. 2nd IFAC Workshop on New Trends in Design of Control Systems, Smolenice, 116–121.

Schnotz, W., Grzondziel, H. (1999): Individual and co-operative learning with interactive animated pictures. European Journal of Psychology of Education, 14: 245–265.

Schriber, T. J. (1991): An Introduction to Simulation using GPSS/H. New York: Wiley.

Selman, D. (2002): Java 3D Programming. Greenwhich (CT): Manning Publications.

Simon, W. (1972): Mathematical Techniques for Physiology and Medicine. New York: Academic Press.

Straßburger, S., Schulze, T. (2001): Verteilte und Web-basierte Simulation: Gemeinsamkeiten und Unterschiede. 15. Symp. Simulation, Paderborn, Frontiers in Simulation, 169–174.

Straßburger, S., Schulze, T., Klein, U., Henriksen, J. O. (1998): Internet-based Simulation using Off-the-Shelf Simulation Tools and HLA. 1998 Winter Simulation Conf., Washington D.C., Society for Computer Simulation International (SCS), 1669–1676.

Sun (2003): Sun Microsystems, Java. Online available: http://java.com/getjava/.

Tversky, B., Morrison, J. B., Betrancourt, M. (2002): Animation: can it facilitate? International Journal of Human-Computer Studies, 57 (4): 247–262.

W3Org (2003): World Wide Web Consortium: Online available: http://www.w3.org/.

Walrath, K., Campione, M. (1999): The JFC Swing Tutorial. Reading (MA): Addison Wesley.

Weller, J. M. (2004): Simulation in undergraduate medical education: bridging the gap between theory and practice. Medical Education, 38 (1): 32–38.

Wigton, R. S. (1987): The use of computer simulation in teaching clinical diagnosis. Computer Methods and Programs in Biomedicine, 25 (2): 111–114.

Wilson, S., Kesselman, J. (2000): Java Platform Performance. Reading (MA): Addison-Wesley.

Authors

Andreas Holzinger
is Associate Professor for Applied Information Processing at Graz University of Technology and Head of the Research Unit HCI4MED at the Institute for Medical Informatics, Statistics and Documentation of the Medical University Graz. He is chair of the WG Human-Computer Interaction and Usability Engineering (HCI&UE) of the Austrian Computer Society and founder of the Special Interest Groups HCI4MED and HCI4EDU. Andreas is an expert at the triple intersection of Education, Psychology and Informatics, consultant for several ministries of science in Europe, national expert and evaluator in the EC and member of several organizations. He was Visiting Professor at Middlesex University London, School of Computing Science in summer 2007, at Vienna University of Technology, Institute of Software Technology and Interactive Systems in winter 2005/06 and at Innsbruck University, Institute of Learning and Organization in winter 2004/05. To date 185 publications and 125 lectures/talks (March 2008). http://www.basiswissen-multimedia.at/.

Werner Emberger
is working at the Institute of Human Genetics of the Medical University Graz. He is involved in diagnostics, genetic counselling and research as well as medical education. Werner holds an MD from Graz University. He is member of the Austrian Society of Human Genetics.

Sigi Wassertheurer
is a mathematician with major research and development interests in physiological modelling and simulation. He is Head of the Cardiology and Haemodynamics Group of the Austrian Research Centers GmbH – ARC, Division of Biomedical Engineering. Sigi is active in various scientific societies including ASIM (Arbeitsgemeinschaft Simulation). Currently, he and his group focus on model-based non-invasive measurement techniques.

Lisa Neal
is an Adjunct Clinical Professor of Public Health and Family Medicine at Tufts University School of Medicine in Boston (MA), USA, where she teaches mainly about online consumer health. She is an e-learning consultant who has taught online and developed online courses for a wide range of topics and institutions. She is Editor-in-Chief of the eLearn Magazine of the Association of Computing Machinery (ACM). Lisa was awarded the 2003 U.S. Distance Learning Association (USDLA) Award for Most Outstanding Achievement by an Individual in Corporate/Business. Lisa holds a Ph.D. in Computer Science from Harvard University. Currently she blogs about education and health at http://lisaneal.com/.