Java Teachware – The Java Remote Control Tool and its Applications[*]

Thomas T. Fuhrmann[a], Christoph Kuhmünch[b], Gunther Schöppe[a],
Gerald Pätzold[a], Dieter W. Heermann[a], and Wolfgang Effelsberg[b]

[a] Institut für Theoretische Physik[†]
Universität Heidelberg
Philosophenweg 19
D-69 120 Heidelberg

and

Interdisziplinäres Zentrum für wissenschaftliches Rechnen
der Universität Heidelberg
Germany

[b] Lehrstuhl Praktische Informatik IV[‡]
Universität Mannheim
L15,16
D-68 131 Mannheim
Germany

Abstract

This paper describes the technical aspects of the Interactive Home Learning Project which provides electronic material for higher education. The target user group are students with an Internet connection via the universities' ISDN- or modem-interface. Two learning modes will be investigated: In the first mode, the material will be provided asynchronously through web-servers, while in the second mode, students with an ISDN-connection at home can attend lectures synchronously. In both modes we want to include animations and simulations in form of Java applets as a new teaching technique. We present our *Java Remote Control Tool* which allows to control distributed Java applets.

1 Introduction

Within the last years the Internet became more and more widely available. Especially many students got access through modem- or ISDN-interfaces provided by their university. The Interactive HomeLearning Project (IHL) sponsored by the Bundesministerium für Bildung und Forschung (BMBF) aims at those students. The goal is to offer the students educational material via the Internet. Apart from supplying traditional material like hand-outs and slides, we

[*] This project is supported by the BMBF (Bundesministerium für Forschung und Technologie)

[†] URL `http://wwwcp.tphys.uni-heidelberg.de/`

[‡] URL `http://www.informatik.uni-mannheim.de/informatik/pi4/`

especially concentrate on four new forms of educational paradigms: animation, simulation, audio and video. While there already exist several commercial solutions to provide audio and video streams asynchronously through web-servers[1], no methods exist to accomplish the synchronous transmission.

Since the TeleTeaching Project of the universities of Mannheim and Heidelberg uses the Multicast Backbone (MBone) to transmit lectures via the Internet IHL has to provide methods to transmit these data streams through ISDN-channels. Additional material (text, pre-recorded video, animations and simulations) is integrated in form of HTML and Java.

The paper is structured as follows: Section 2 gives an overview of the TeleTeaching Project of the universities of Mannheim and Heidelberg. Several scenarios we plan to implement will be described shortly. Section 3 explains the Interactive HomeLearning scenario in more detail. In section 4 some example teachware applications are presented and in section 5 we describe our Java Remote Controller Tool which supports the implementation of distributed simulations and animations in Java. In the last section conclusions and an outlook to our future work are given.

2 The TeleTeaching Project of the universities of Mannheim and Heidelberg

Mannheim and Heidelberg are only about 25 km apart from each other. Furthermore the universities of the two cities offer complementary kinds of lectures; e.g. Mannheim has several professorial chairs for computer science but offers no lectures in physics. Heidelberg on the other hand is well known for its department of physics. Because of this situation both universities were interested in working together.

In January 1996 the universities of Mannheim and Heidelberg started a joint TeleTeaching project. In this project we planned to implement three different TeleTeaching scenarios.

The first scenario we implemented was the *"Remote Lecture Room Scenario"* in which whole lectures are transmitted via the Internet. Since summer-term 1996 lectures are exchanged between the two cities on a regular basis. The MBone tools (vic and vat) [Hui95, **?**] are used to transmit audio and video to the respectively remote side. Furthermore a video signal showing the students on the remote side is transferred to the lecturer's control monitor and the shared whiteboard-tool is used to show slides. Up to now the MBone tools have only been employed in the unicast mode which means that no other sides can join the lectures.

The second scenario which is also already implemented is called the *"TeleSeminar Scenario"*. During the winter-term 1996/97 a joint seminar on "Digital cash in the Internet" took place in Mannheim, Freiburg and Karlsruhe. While speakers from Mannheim and Karlsruhe investigated into the technical aspects of digital money students from Freiburg gave talks about the social and economical consequences. Again, the MBone tools were used to transmit the talks but this time in multicast mode. In contrast to the "Remote Lecture Room Scenario" the seminar was much more interactive and a discussion between the three sides took place after each talk.

The third scenario which has not yet been implemented is called the *"Interactive HomeLearning Scenario"*. Here the target group are typically students running Windows 95 or Linux on their PC with an Internet connection via ISDN. This scenario has the most widespread distribution of members. It can be divided into two phases:

- In the asynchronous learning phase production and consumption of the educational material takes place asynchronously i.e. at different times. During the last two terms we recorded several lectures. Besides from educational hypertext, interactive animations, and simulations, this material will be offered to the students in the World Wide Web.

[1] Two examples for commercial solutions are VDO[AG97] which is used by the Deutsche Tagesschau and Real-Audio[Net97].

- In the synchronous learning phase, production and consumption of the lectures take place synchronously, i.e. at the same time. In this phase we plan to transmit lectures live to the students at home. The required Internet connection will be based on ISDN.

The organizational, pedagogical and technical aspects of the first two scenarios are discussed in more detail in [EGE97]. In the following section we will hence concentrate on the technical aspects of the HomeLearning scenario.

3 The Interactive HomeLearning Project

The goal of the IHL project is to establish the geographical and temporal independence of all the participants of a lecture or seminar. The teacher and students are no longer restricted to a common place like the lecture hall — or, as in case of today's video conference to specially equipped rooms — but will be able to participate while sitting in front of a Workstation at a CIP Pool or a PC at home.

Additionally, students can attend the lectures not only online while they are held, but anytime afterwards as well, since hand-outs, slides and selected video sequences will be gathered in a special multimedia database which is accessible through a web-server. A further didactic surplus of the multimedia processing of the lectures is achieved by supplementing simulations and visualizations of the lectures' contents.

These advantages of utmost independence predestine this method for adult education and the collaboration of geographically separated research groups.

In difference to the TeleTeaching and TeleSeminar scenarios — where expensive multimedia equipment was required — here costs are minimized, because a PC already belongs to the basic equipment of today's students anyway. This is somewhat similar to the old Telekolleg where a TV set was prerequisite. But in contrast to the TV version, IHL allows for active participation in form of questions during lectures or special tutorial programs.

The project is divided into two phases of 12 months each, the asynchronous and the synchronous phase. In both phases it is planned to work on specific elements of the whole scenario in order to find concise instructions for implementation.

4 Java Teachware

Throughout the lectures we make use of simulations. Those elements of interactive learning are realized as Java applets, since Java has a lot of advantages for the design of the interactive elements: First of all, Java simulations are platform-independent. Secondly, text, video, and Java applets can be integrated into a multimedia learning unit accessible by every standard web-browser. Furthermore, Java provides easy-to-handle means for graphics and network access. The latter is a great advantage for the Java remote control program described in chapter 5. Altogether students can now make hands-on experience during the lectures and explain their questions directly at the object.

The following subsections illustrate this approach by giving some examples developed by students during their practical programming courses.

1. In the computational physics lecture, students can easily test different sets of parameters. This can amount to a feeling for the physics behind it. Learning becomes experience as an experiment is performed.

2. The examples from the computer networks lecture are aimed to illustrate algorithms and protocols. Students can either view a pre-defined example for illustration, modify certain settings or create an entirely new sequence.

4.1 Molecular Dynamic Simulation of a Polymer Chain

The interest in the physics of polymers was pushed by the need of new materials in all fields of everyday life. Prominent examples for this are plastic bags made out of polyethylene or compact discs which mainly consist out of Bisphenol-A-Polycarbonate (BPA-PC). The properties of these materials were investigated by experiments and for some a theory has been established. Nevertheless computer simulations can give a more detailed picture in this context. Polymers can be modeled for computational purposes in a variety of ways. Depending on the kind of question and the degree of abstraction, one has the basic choice between a lattice model and a model in continuum. If one wants to stay as close as possible to the chemically realistic chain it may be of advantage to remain in the continuum (i.e. real space).

The Java application by Thilo Stöferle simulates a single polymer chain in real space using the united atom model[Bis79] with molecular dynamics[2]. In this model linear polymers are represented by chains of mass points. The masses are connected by springs. Moreover angle forces, torsion forces and van der Waals forces between the masses are used. The program reads configurations in some formats and can change the physical parameters even during the simulation which is very useful to demonstrate special effects. As important physical parameters, e.g. potential and kinetic energy, are calculated and plotted during the simulation, it can be used to demonstrate the effect of different forces and the time development of certain observables. The on-line visualization of the polymer chain enables by a hands-on training an intuitive understanding on the interplay of the different parameters of the model and leads to a understanding of the behavior of a polymer chain.

4.2 Lava Lamp Simulation

This applet is inspired by a "lava lamp" which consists of a vertically orientated cylinder filled with two unmiscible fluids of different color and specific weight[3]. The cylinder can be heated from below by an electric light bulb. As long as the lamp is switched off, the fluid with the somewhat higher specific weight forms a layer below the one with lower mass density. When the bulb gets switched on, the lower fluid layer is heated and buoyancy comes into play. Bubbles form and rise upwards the cylinder. During this passage they cool again and finally sink down. An ever changing dynamics of rising and sinking bubbles and more complex fluid structures can be observed[Hoh77, Bin93].

However, the applet does not display such a lamp, but is devoted to the numerical solution and visualization of one of the simplest systems of mathematical equations which can capture such physical behavior. It has been programmed by our students Hans Boie and Thorsten Koch. Central to the applet is a graphical display (a mapping of the simulation domain) in which the two fluids are shown as regions of different color. Buttons to start and stop the simulation as well as for re-initialization are provided. There is also a pop-up window in which the underlying equations are printed. This becomes important if the user begins to play around with other components of the GUI which allow to manipulate quite a number of the free parameters in the equations. Also the size of the simulation domain as well as the graphical appearance and some details in the numerical solution scheme may be customized.

We called this the "lava lamp applet" since many people have already seen such a lamp and easily comprehend or remember the underlying physical principles. The equations used, however, have a much wider applicability, and serve as model equations in many branches of condensed matter physics. A student who explores this applet and the behavior of the solution (i.e. the appearance of the generated graphics) therefore gets an intuitive understanding of this often used paradigm.

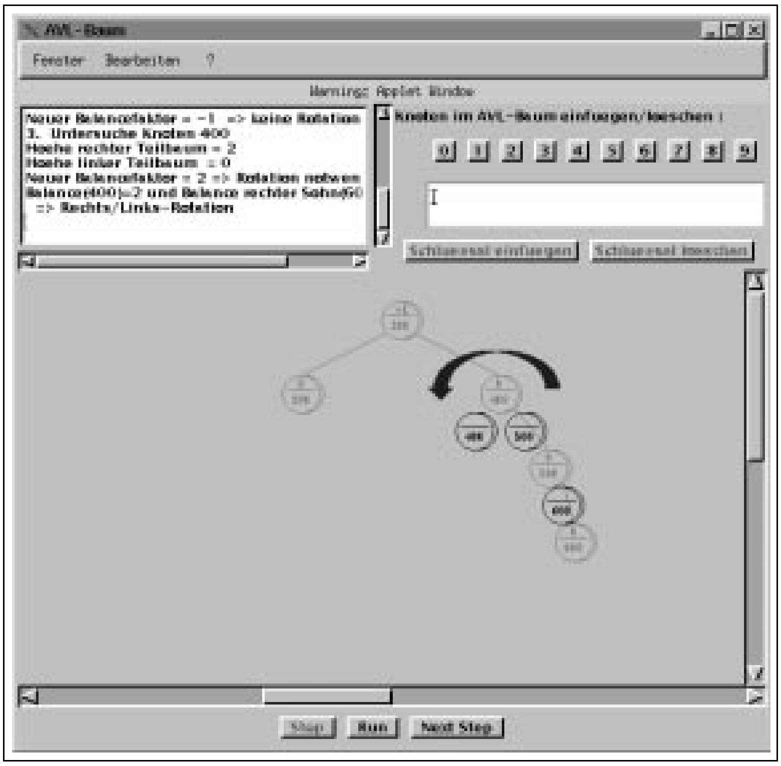
[2] URL http://wwwcp.tphys.uni-heidelberg.de/java/md/

[3] URL http://wwwcp.tphys.uni-heidelberg.de/java/lava/

4.3 Simulation of AVL-Trees

An AVL-tree is an elementary data-structure usually introduced to students of computer science during the first basic courses. The basic concept is a height balanced binary tree. After each insertion or deletion of an element rotation-algorithms are used to rebalance the tree[OW93].

The applet written by Axel Semling concentrates on the visualization of the rotation operations[4]. It supports two steps in the learning process of a student: In the first step the algorithms are explained in that way that an example tree is build by the applet itsself. In the second step the student can freely build a tree by inserting and deleting elements. The applet animates the rotations after each insertion/deletion. While it is the goal of step one to explain the mechanisms of the AVL-tree, the second step shall give the student the opportunity to get more acquainted to the concepts of the AVL-tree.



```
Neuer Balancefaktor = -1  => keine Rotation
3.  Untersuche Knoten 400
Hoehe rechter Teilbaum = 2
Hoehe linker Teilbaum  = 0
Neuer Balancefaktor = 2 => Rotation notwen
Balance(400)=2 und Balance rechter Sohn(60
  => Rechts/Links-Rotation
```

Figure 1: The AVL-Tree-Simulation

It is planned to implement a third step of the learning process by enhancing the applet with a test mode. In the test mode the student is asked to insert or delete elements and then has to rebalance the tree by executing rotation operations. The goal of step three is to test the success of the learning process.

5 The Java Remote Control Tool

Despite all these advantages of the use of Java applets for simulation in TeleTeaching, we are confronted with the problem of distributively controlling the applets. A lecturer will not only want to use audio, video and whiteboard techniques, but also to include Java applets in the way they are presented in the web. Students should be able to view and eventually control these applets synchronously.

[4] URL http://www.informatik.uni-mannheim.de/ cjk/java/avl/

5.1 Java versus alternative approaches

A first solution might be to use shared X techniques, where a central application server runs the required simulation and several distributed X servers display the user X-Windows interface. But this does not prove practicable for several reasons: no general availability of the X-environment, bandwidth problems, and the impossibility of handling different levels of access restrictions. We therefore chose an approach directly based on Java:

- Java enabled web browsers are much more common than X-Windows servers. Many students who would otherwise be excluded from the new educational means can hence join the interactive lectures.
- Customized Java solutions consume comparatively little bandwidth. Students with low-bandwidth connections such as dial-up connections are thus not excluded from sharing the simulation examples. There might even remain enough bandwidth for audio and video purposes. (While cheap networking with high bandwidth is not available, bandwidth should always be a point of concern. Experience shows that e.g. mouse movement in the X-environment can consume up to 20KBit/s!)
- Java allows to easily handle different users with different access restrictions who simultaneously control the application via a network connection. Unrestricted interference of students' and lecturers' actions would otherwise thoroughly spoil the application's functionality.

In addition to that Java enables us to split the programming work of the simulation applets and assign subproblems to students as useful practical exercises. It is then a great advancement in motivation for the students if they see their applets being integrated into the common interactive home learning scheme.

5.2 Conceptual outline

According to the specifications outlined above we decided for a solution that adds means for distributed control to the GUI of an existing applet. The main features are:

- Different users sign up for participating interactively (e.g. automatically by invoking a web-page that contains the applet).
- They then receive all GUI interaction commands via network which forces all the applets to behave coherently. (The applets will not run identically since processing speed on different machines might be different. Methods for synchronization will be explained below.)
- Floor control has to be implemented: If a user wants to control the applet her/himself s/he can apply for being granted the control (e.g. by pressing a control button).
- A master program (on the server providing the applet) checks the request and eventually hands the floor control over to that user. The user who possessed the control before looses the control over the applet.

Since the "apply for control"-button is placed in a separate window, the GUI of the applet need not be changed. This entirely conserves the web-pages' layout allowing for an easy migration towards distributed control.

The web-server providing the applet's class code is in charge of distributing the GUI events among the applet instances and grants the right for controlling the applet to the different users.

5.3 Technical outline

Since changing specifications is one of the biggest source for software malfunction the changes in the existing applets should be as small as possible. This goal was met by a solution that in fact requires no changes in the applets at all.

We defined a class `DistributedControl` that besides its constructor only needs one method, the `handleEvent` method. Applets that are to be controlled distributively will instantiate one object of that class and send every event to that object. Processing of that event will then take place in the `DistributedControlWindow` that will be explained in a moment.

As a result of this minimal interface construction it is even possible to implement distributive control into classes for which no source code is available. This can be done by overriding `handleEvent` for that class, invoking `DistributedControl.handleEvent` and then calling `super`.

On instantiating a `DistributedControl` object a `DistributedControlThread` is started, the purpose of which is to hold contact to the control server via a socket connection.

Another feature of the `DistributedControl` class is finally that if it is called as application it will work as control server, i.e. it suffices to place the class files on the web-server and start `DistributedControl` there. Coordination of all the parties joining will then be handled automatically. This feature is achieved by additionally supplying a `main` method for `DistributedControl`.

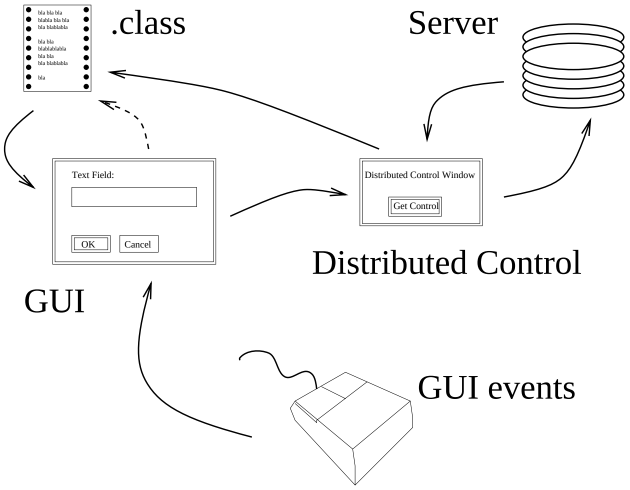


Figure 2: Instead of handing GUI events directly to the applet's code (dashed line) events are redirected to the `DistributedControlWindow` that holds a socket connection to the applet's web-server. If the applet is active this event is distributed. If the applet was not granted the floor control events will be blocked by disabling the GUI.

The `DistributedControlWindow` acts as local control unit. At first it blocks all GUI events for the applet by disabling its GUI. It then pops up a window with the 'get-control'-button and a status display. If the user presses the button a request for floor control is sent to the server. The server then notifies the applet that holds the floor control to release the control. Finally the floor control is handed to the applet that applied for the control and its GUI elements are enabled.

A time-out mechanism is required to cope with network breakdowns. If the applet holding the floor control does not release it on request the server will close that connection. Unlike in the normal case, here events might be lost leaving this applet in a possibly non-coherent state. All other applets, however, remain coherent. But since the server logs all events of a session allowing for 'coming late to lecture' a reconnect is always possible.

5.4 Synchronization

Event handling has thoroughly changed between Java 1.0.2 and Java 1.1. We are hence forced to re-implement the scheme described above. We use this chance to extend our concept towards synchronized distributed control as follows.

Many simulations not only rely on user input but also on random numbers to simulate effects. Together with different processing speeds on different platforms this leads to diverging applet states. This need not be a principal problem. In fact mostly the solution described above suffices completely.

For the cases where really identically running applets are required we introduced a `broadcast` method that broadcasts objects e.g. random numbers, user input, GUI events. Instead of writing `z = Math.random()` we write `z = (Double) distributedControl.broadcast( new Double( Math.random()))`. (In order to save bandwidth we had rather broadcast a Random object than the random numbers themselves. But this example is only for demonstration.) The broadcast itself is handled as described in figure 2. However implementation has become easier due to RMI.

Synchronization is now also easily handled by inserting a `distributedControl.broadcast( null )`. Hence properly programmed applets i.e. applets that do not use local events or random numbers but only the broadcasted objects will now always run identically.

6 Conclusions and Outlook

In this paper we described interactive HomeLearning which is a module of the TeleTeaching project of the universities Mannheim and Heidelberg. The idea of using Java applications in lectures leads us to the need for a tool to remotely control them. We developed the *Java Remote Control Tool* which allows to control distributed Java applets. Main advantages are the low bandwidth need and the ability to handle different access restrictions. Furthermore, our implementation of the Java Remote Controller Tool can easily be adapted to already existing Java applets. Upgrade to Java 1.1 gives us the chance to extend our implementation towards full synchronization.

7 Acknowledgments

The authors are grateful for the support by A. Linke and T. Hapke and thank Thilo Stöferle, Hans Boie, Thorsten Koch, and Axel Semling for their commitment during the software practicals that lead to the development of the Java simulations. This work was supported by the Bundesmisterium für Forschung und Technologie.

References

[AG97] VDO Adolf Schindler AG. http://www.vdo.com, 1997.

[Dor95] T. Dorcey. Cu-seeme desktop vidoe conferencing software. *Connexions*, March 1995.

[EGE97] Andreas Eckert, Werner Geyer, and Wolfgang Effelsberg. A distance learning system for higher education on telecommunications and multmedia - a compund organisational, pedagogical and technical approach. *accepted at Ed-Media 97*, 1997. accepted at Ed-Media 97.

[Hui95] Christian Huitema. *Routing in the Internet.* Prentice-Hall PTR, Englewood Cliffs, NJ 07632, 1 edition, 1995.

[Net97] Progressive Networks. Real-audio/ video homepage. http://www.real.com/, 1997.

[OW93] T. Ottmann and P. Widmayer. *Algorithmen und Datenstrukturen.* BI-Verlag, Mannheim, 2 edition, 1993. Umfassendes Werk zu Datenstrukturen.

[PMSS97] Peter Parnes, Mattias Mattsson, Kare Synnes, and Dick Schefström. mtunnel: a multicast tunneling system with a user based quality-of-service model. *IDMS*, March 1997. eingerreicht auf der IDMS.

[Bis79] M. Bishop, M. H. Kalos, and H. L. Frisch. *J. Chem. Phys.* **70**, 1299, 1979

[Hoh77] P. C. Hohenberg and B. I. Galperin. *Rev. Mod. Phys.* **49**, 435 (1977)

[Bin93] J.J. Binney, N.J. Dowrick, A.J. Fisher, and M.E.J. Newman. *The theory of critical phemomena* (Clarendon Press, Oxford, 1993)