

# A Framework for Web-based E-Learning of Discrete Event Simulation Concepts

Bernd Page<sup>†</sup>  
University of Hamburg

Wolfgang Kreutzer<sup>††</sup>  
University of Canterbury, New Zealand

## Abstract

This paper describes selected e-learning materials developed as part of a cooperative simulation-supported learning initiative for university courses on discrete event simulation. These materials include Java applets for teaching general simulation concepts, as well as a web tutorial and an Internet-hosted simulation laboratory which teaches how to use a simulation framework in Java.

## 1 Motivation and Overview

E-Learning uses electronic media to improve teaching and learning. In the context of computer simulation it supports learning *about* computer simulation as well as *making use* of simulation technology to improve teaching and learning in other disciplines. Since it aims to improve learning *about* modelling concepts we refer to the first case as *simulation-focussed* learning [see Neu2005], where simulation methodology and relevant software are introduced, explained and practised. Simulation-focussed learning helps students to understand computer simulation methodology and its practical application to problems in specific application domains; such as logistics. In the second case, which we call *simulation-based* learning, computer simulation (often in combination with visualisation and animation) augments presentations of concepts in domains such as biology, medicine, business, logistics or computing. Here models and their simulation help to *explain* complex processes, illustrate abstract knowledge, uncover connections or interdependencies, and encourage experiential learning and situated applications of knowledge in the relevant field. Aircraft simulators or models of medical procedures are examples for applications which have long been used to train pilots or doctors. A more in-depth discussion of the relationship between simulation and e-learning can be found in [Neu2005].

In this paper we emphasize *simulation-focussed learning* of core simulation ideas; i.e. we look at digital e-learning materials that include computer simulations for assisting the teaching of simulation courses. In this context we

---

<sup>†</sup> Professor at the Department for Informatics, University of Hamburg, Hamburg, Germany

<sup>††</sup> Associate Professor at the Department for Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand

specifically wish to support *blended learning*; i.e. the use of classical teaching evens and tools (e.g. conventional lectures and laboratories, textbooks, etc.) alongside innovative communications media (e.g. web-based learning tools). This goal reaches beyond simple uses of information technology to facilitate self study or distance education.

The teaching materials presented in this paper include Java applets for illustrating general simulation concepts (e.g. how event lists work), a web-based tutorial, and a simulation laboratory for teaching the use of the DESMO-J simulation framework (“Discrete Event Simulation in Java”, see [Pag2005]). These components have been developed as part of a cooperative e-learning project by the Universities of Hamburg (Germany) and Canterbury (New Zealand). In Germany the project was sponsored by the ELCH-program<sup>1</sup> and by the German Federal Ministry for Education and Research (BMBF) in its program for Scientific-Technological Collaboration with New Zealand (in 2004-2005<sup>2</sup>).

## 2 A Java Framework for Discrete Event Simulation

The DESMO-J simulation framework has been under active development at the University of Hamburg for many years (see [Pag03]; [Pag05], Chap. 10), One of its main motivations has been the need for a suitable tool for teaching the programming language aspects of discrete event simulation to Computer Science and Information Systems students. Although DESMO-J cannot be considered an e-learning component in a strict sense, it has proved very useful as a simulation development tool with a strong emphasis on didactical aspects. Our experiences in using DESMO-J for teaching purposes has further strengthened our conviction that understanding of basic simulation concepts must be based on practical work in designing and implementing simulation models, using familiar tools such as *UML* and *Java*. This demand becomes even more relevant for students with little “real world” experience but sound Computer Science background.

The DESMO-J framework supports this learning style well and has been successfully used in courses on discrete event simulation methodology and its applications at the University of Hamburg, the University of Canterbury and elsewhere.

DESMO-J is based on Java. Although we gain some technical advantage (see [Pag2005], Chap. 6.2.3) by using Java as a simulation tool’s base, this decision was largely motivated by Java’s widespread adoption as a first programming language in undergraduate Computer Science curricula. DESMO-J offers a

---

<sup>1</sup> Financed by the “E-Learning and Multimedia in College” project of the federal State of Hamburg’s Department of Science and Technology, Project-number uELCH A 0006

<sup>2</sup> Project- number NZL 00/02

comprehensive simulation framework and not simply a simulation language. In contrast to other tools of this kind, whose users often can not venture beyond what can be expressed in a rather simplistic graphical interface, this means that a skilled practitioner may need to dig deeper, down to DESMO-J's programming layer and its class library. To assist learning the relevant skills we provide both a web-based tutorial (see Section 3.1) and a web-based laboratory (see Section 3.2). These e-learning components are meant to complement more conventional learning materials; e.g. a textbook such as *The Java Simulation Handbook* (see [Page2005]), which was written as part of the project. In line with the goal of a *blended learning* experience, they also encourage and support students' active participation and experimentation.

Some simulation knowledge is needed to effectively use both tutorial and laboratory. This will typically be acquired in lectures, or through reading relevant textbooks and lecture materials. Experience has shown, however, that true understanding of theory is not possible without application, an insight which corroborates the basis of cognitivist educational theories (e.g. as discussed in [Neu2005]). DESMO-J is an excellent tool for supporting the discovery-driven learning styles these theories recommend, particularly if a student already has good Java programming skills.

To build their first models, students must understand DESMO-J's core concepts and the framework's architecture. Here they encounter some of discrete event simulation's most basic ideas: event lists, event classes and methods, different types of statistical distributions, model instrumentation etc. Customizing a model makes use of *Java*. Since we assume that students are already competent Java programmers, they can concentrate solely on learning the relevant simulation concepts, which will be new to them. A further advantage of DESMO-J in this context is that it supports both event-based and process-based model descriptions and allows experimentation with these alternative world views within the same software (see Chapter 5.2 of [Page2005]). Students who have used DESMO-J in such a discovery-driven style will generally develop a deeper understanding of key discrete event simulation concepts than users of other modelling tools.

### **3 Some E-Learning Tools for Simulation Courses**

In this section we present two web-based tools that are part of the e-learning materials which support blended learning in a traditional lecture-based course on discrete event simulation.

#### **3.1 A Web Tutorial for the DESMO-J Simulation Framework**

The web tutorial introduces DESMO-J to students who are familiar with object-oriented programming in Java and have a basic understanding of discrete event simulation. *Figure 1* shows how this tutorial is split into a number of separate

sections. The ones on the left-hand side deal with general issues of DESMO-J's usage. This includes a brief overview of its functionality (*DESMO-J in a nutshell*), a demonstration of what a simulation with DESMO-J looks like (*DESMO-J at work*), and a concise introduction to building a DESMO-J model (*DESMO-J distilled*). The right-hand side sections offer more in-depth discussions of different parts of the framework (i.e. statistical data collectors, stochastic distributions, advanced modelling features) and extensions for special application domains (e.g. harbour logistics).



Figure 1: Entry page to the DESMO-J Tutorial

DESMO-J supports both process-oriented and the event-oriented modelling styles, and the tutorial offers a step-by-step introduction to how to use either. For more detail a student can always refer to the API (i.e. “application programmer interface”) documentation for each class and each package within the DESMO-J framework.

### 3.2 An Internet Laboratory for DESMO-J

DESMO-J's Internet laboratory is another integral part of the DESMO-J e-learning experience. It offers a wide range of DESMO-J models, each of which includes problem descriptions and all functionality needed to encourage a student's independent exploration. The laboratory has three main components:

- a *model presentation* component
- an *extension* component, with additional models and exercises
- an *administration* component, with user and rights management features

The main goal of the laboratory is to enable a student to work with examples, learn about simulation techniques, and learn how to apply them in DESMO-J. This style of learning is supported by the *model presentation component*, which allows access to a number of each model's aspects; i.e. model description or problem specification, areas of application, images, external links to additional information, UML 2 activity diagrams (see [Kna2005]), details for using DESMO-J classes and methods, source code in DESMO-J or Java, reports and traces (as generated by DESMO-J runs), and further comments.

The *extension component* permits learners to add their own models to the laboratory, without any need for knowing the its internal structure. Layers of access to models or model components accommodate the needs of university teachers who wish to deploy graded exercises. A *Role-Based-Access-Control-Concept* (RBAC) has been developed to administer these rights. It allows individual identification and passwords with different access rights for different user groups.

The teaching materials stored in this laboratory contain many DESMO-J models. These can be used for discovery-led learning of various modelling and simulation concepts, in the style suggested by constructivist learning theory (see [Neu2005]). Since uploading new models is rather straightforward, this collection can easily be extended. Currently it contains models that illustrate different application domains (e.g. harbour logistics, production systems) and different modelling concepts (e.g. random experiments without time dependency, event oriented simulations, process oriented simulations, different patterns of process synchronization etc.). Since it is implemented in different versions and uses different modelling viewpoints, the model "featherships in the Baltic sea" (see [Page2005], Chap. 5.2, 5.3, and 10.5) offers students a wide ranging overview of DESMO-J's functionality in relatively compact form. It can also be used as a guideline for implementing a student's own models.

## 4 Using Applets to Teach Key Simulation Concepts

DESMO-J's E-learning package distinguishes between *simulation* and *documentation* applets. While both the web tutorial and the Internet laboratory are stand-alone modules, *applets* are "...an embedded Java application, which executes in an applet viewer, such as a web browser" (see [Nie00], p. 654). By embedding such Java applets in web pages, e-learning software augments and enriches presentations with highly visual and interactive demonstrations and explanations. The DESMO-J tutorial, for example, contains a web page where learners can interact with a "live" simulation.

### 4.1 Simulation Applets

*Simulation applets* allow students to configure models, run simulations, and analyse the results directly from within a web page. Such applets can be implemented completely inside the DESMO-J framework itself. *Figure 2* shows the results of an applet-based simulation of a simplified model of loading trucks in a container terminal.

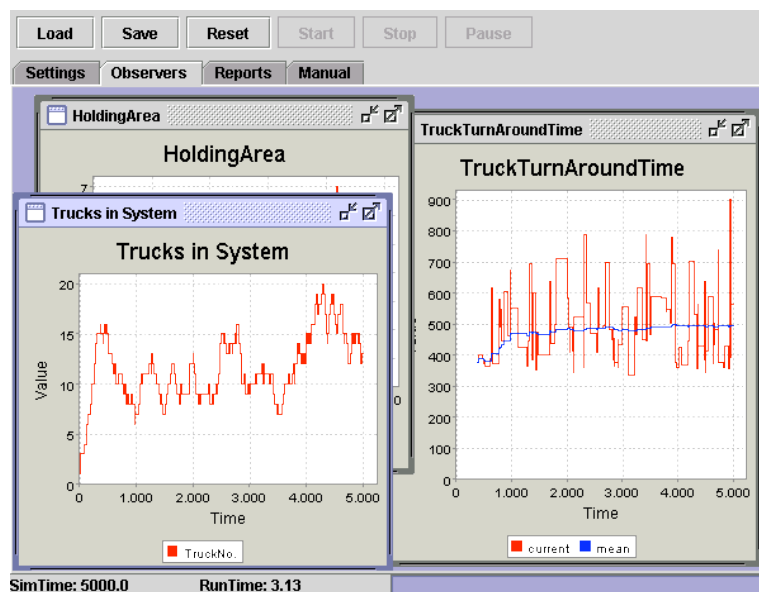


Figure 2: Java applet showing a simulation of trucks in a container terminal

Simulation applets can be used for reference models as well as for exercises in running a simulation and the subsequent statistical analysis of simulation results.

DESMO-J supports implementation of simulation applets by providing a `gui.ExperimentStarterApplet` class. Once we insert one of its

instances into a web page, the page's visitors can control linked simulations through a graphical interface. For example, they can change a model's input parameters, its initial state, and its output formats. No direct changes to model code are required. This is an important prerequisite to make experimentation with models accessible to a wide range of users and purposes.

## 4.2 Documentation Applets

By using images, sound and text, *documentation applets* explain key simulation concepts to students. Website visitors can view step-by-step animations, which can be interrupted and rewound at any time. Each animation step is accompanied by explanatory text, which can also be read to a visitor, in order to ensure minimal distraction while following the visual presentation. We will look at two such documentation applets in more detail.

slide 4 of 17

Call Event Routine

```

eventRoutine(<entity #2>) {
...
<entity #2>.schedule(<event #4>, new SimTime(1.8));
<entity #4>.scheduleAfter(<entity #2>, <event #5>);
}

```

1.100 scheduler

*current* *event list*

1.100	1.150	1.500	20.000
<entity #2>	<entity #1>	<entity #3>	--
<event #1>	<event #3>	<event #2>	<end event>

The routine for the current event is called.  
Two new events are to be scheduled in this routine.

◀ Previous      Play      Next ▶

Figure 3: A Java animation which demonstrates how event lists work

The “Scheduler-Demo” applet shown in *figure 3* explains how a discrete event simulator executes simulations. Its particular goal is to help students understand how *simulation event lists* are processed. To this end it shows how DESMO-J places (i.e. *schedules*) events on its agenda and how this list is then used to determine what event or process should next gain control. The intricacies of this procedure are fairly abstract and often not easily understood by a beginner. The combination of verbal description and visual animation improves comprehension. Since it is available on a web page, the applet can be accessed and watched by a student at any time and as many times as desired.

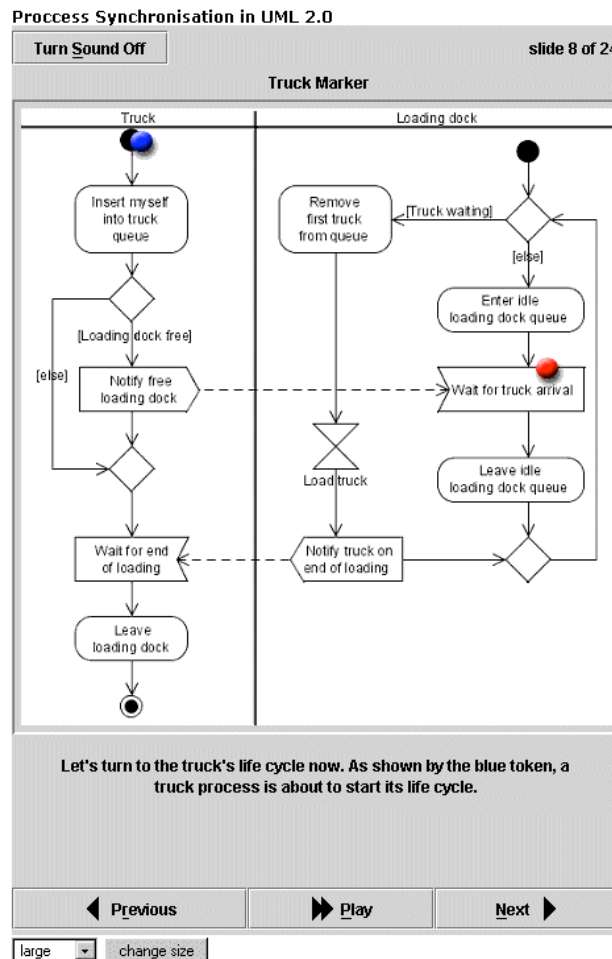


Figure 4: An applet which shows process synchronizations with UML diagrams



Another applet, called “*Process-Synchronisation in UML 2*” (see *figure 4*), uses a simple queuing scenario to demonstrate patterns of synchronisation in process-oriented models. Within this context two *activity diagrams*, one for client and one for server processes, are presented, using the well established *Unified Modelling Language (UML-2)* notation. To facilitate this, we have used the *UML Stereotype* mechanism to implement several extensions for describing discrete event simulations within UML 2 diagrams. Details of these extensions as well as their motivation have been published elsewhere (see [Kna05]). Animating these diagrams uses *Petri net* style movements of tokens to demonstrate changes in process states. Simulation processes’ actions, like *activate*, *passivate* and *hold*, which are hard to describe textually, can now be explained in a more motivating and lively manner.

Other Java-applets developed in this e-learning project are described in more detail in [Mey05].

## 5 Conclusions

E-learning technologies are well suited to augment and improve classical teaching styles. From the viewpoint of educational theory, such *blended learning* stimulates autonomous knowledge construction within a learning community. Teaching becomes independent of temporal and physical constraints and can take better account of individual strengths and weaknesses. Students can learn anywhere, at any time, and at their own speed. We have presented some examples of e-learning materials based on Java and web-technologies. These learning components have successfully been used to support knowledge acquisition in discrete event simulation courses.

Computer simulations can help greatly to clearly present abstract ideas and processes whose details and implications would otherwise be much harder to understand. On the other hand, e-learning also supports the teaching of simulation techniques, such as designing simulation models with UML and implementing them with a Java tool such as DESMO-J.

The symbiosis of e-learning and simulation brings many advantages to both fields. The DESMO-J web tutorial and its web-based simulation laboratory are well placed to deliver this promise.

## References

- [Coe02] O. Coenen. *E-Learning-Architektur für universitäre Lehr- und Lernprozesse*. Lohmar, Köln, Josef Eul Publ., 2002.
- [Kna05] N. Knaak, B. Page. *UML as a Modelling Language in Discrete Event Simulation*. Proc. EMS 2005, Riga, June 2005.
- [Mey05] R. Meyer, A. Möller, B. Page. *Modellierungswerkzeuge zur Simulation umweltökonomischer Systeme – Interaktives Modul im Rahmen der universitären Präsenzlehre der Angewandten Informatik und der betrieblichen Fortbildung*. Final Project Report, E-Learning Consortium Hamburg (ELCH)-Project, University of Hamburg, Dep. of Computer Science, Hamburg 2005.
- [Neu03] G. Neumann. *Simulation in a problem-based e-learning environment for logistics*. In: Y. Merkurjev, A. Bruzzone, G. Merkurjeva, L. Novitsky, and E. Williams, editors, Proceedings of HMS 2003 – The International Workshop on Harbour, Maritime & Multimodal Logistics Modelling and Simulation, pages 268–276, Riga, 2003.
- [Neu05] G. Neumann, B. Page, G. Kiesel, R. Meyer, W. Kreutzer. *E-Learning and Simulation*. In: [Pag05], Chap. 14, pp. 401-432.
- [Nie00] P. Niemeyer, J. Knudsen. *Learning Java*. O'Reilly, Sepastopol, 2000.
- [Pag03] B. Page, E. Neufeld. *Extending an object oriented Discrete Event Simulation Framework in Java for Harbour Logistics*. Intern. Workshop on Harbour, Maritime & Multimodal Logistics Modelling and Simulation – HMS 2003, Riga, Latvia, Sept. 2003, pp. 79-85.
- [Pag05] B. Page, W. Kreutzer. *The Java Simulation Handbook. Simulating Discrete Event Systems with UML and Java*. Shaker Publ., Aachen, Germany 2005
- [Sch03] R. Schulmeister. *Lernplattformen für das virtuelle Lernen*. München, Wien, Oldenbourg Verlag, 2003