Introduction to Business Informatics with Greenfoot Using the Example of Airport Baggage Handling

Detlef Rick, Julia Ludwig, Sebastian Meyer, Carsten Rehder, Ingrid Schirmer
Department of Informatics
Universität Hamburg
Vogt-Köln-Straße 30, D-22527 Hamburg, Germany
{rick, ludwig, meyer, rehder, schirmer}@informatik.uni-hamburg.de

ABSTRACT

Complexity being an essential part of our everyday and occupational life, the following question arises: Which fundamental skills do pupils need to develop, in order to be well prepared for handling future complex technological, and social, systems? We believe that general education should address this issue by cultivating computational—or rather “informatical”—thinking, i.e. an informatical view on the world.

The educational programming environment Greenfoot consolidates the strengths of traditional microworlds with Java’s scalability [1], thus enabling the provision of rather complex systems, most notably simulations of material or traffic streams. In Greenfoot, these systems can be interactively explored and manipulated. Thus Greenfoot is not only an excellent learning environment for introductory programming courses but it is also particularly well suited to address more application-oriented issues in computer science.

In a one week school project pupils in their eighth year were presented with a Greenfoot simulation of a highly simplified airport baggage handling system. Using the example of baggage handling, they got involved not only with programming but also with applied computing, namely with issues related to the field of Business Informatics (BI, German Wirtschaftsinformatik). Our aim in including applied computing issues in the school project was to present a broad image of computer science.

Keywords
business informatics and general education, applied computing, complex systems, simulation, Greenfoot

1. COMPLEX SYSTEMS IN A COMPLEX WORLD

Increasing complexity, in its messiness and its beauty, is an important part of our everyday life. Complex technological, biological, and social systems are embedded in, and interact with, other systems, thus forming even more complex and constantly changing systems. These typically cannot be described in linear terms: “Such systems are very difficult to understand, and often impossible to analyze with traditional precollege mathematics.” [2]

Information technology plays a central role in the study of systems, in that today it is closely interwoven with virtually all kinds of systems. For example, complexity in organisations is driven by, and tackled with, complex information systems. Information systems are an integral part of their context, and vice versa.

At the same time information technology gives us the means of interpreting and understanding complex systems. Computers allow us to construct our own virtual worlds, and thus to explore and share our understanding of systems and the real world’s complexity. Computational devices and artifacts change the way we make sense of a complex world, not only by helping us to “tackle problems we would not [have dared] take on before the age of computing” [7], but also by presenting themselves as “objects-to-think-with” [6]. These devices give us examples of complex systems, affecting our way of thinking and our imagination.

We believe that general education should prepare pupils for the challenges of future complex technological, and social, systems. Computer science (CS) courses at primary and secondary schools should cultivate skills summarised under the term “Computational Thinking” [7] by including applied computing issues such as are found in the field of Business Informatics.

2. BUSINESS INFORMATICS IN CS EDUCATION

Business Informatics (BI), which is a well established discipline in Germany and other European countries, is an example of a discipline based upon applied computing. Akin to Information Systems, BI draws on computational concepts to approach business and organisational problems arising from, and tackled by, the use of information systems. In BI the study of systems, the design of solutions and computational or more generally speaking, “informatical” thinking converge.

At the Department of Informatics, University of Hamburg, during a seminar project, a group of BI, CS and CS education students carried out a one week CS school project for a school class in their eighth year. We included lessons of applied computing, namely problems belonging to the field of BI, for several reasons.

First of all, we wanted to present a broad image of computer science, and thus rouse the pupils’ interest in CS. In general, contextual approaches are considered promising in order to increase participation in CS—in particular female participation [4].

Another reason was that we wanted to teach object-orientation, i.e. the paradigm as a mental tool and, to a lesser extent, the programming. Nygaard suggested that “teaching object-orientation must start with a sufficiently complex ex-
ample” [5]. Business informatics, being concerned with complex socio-technical systems, provides us with many such examples not all of which are suitable for educational purposes. We chose baggage handling at airports as an example.

Finally, one more reason for us to include BI lessons in the school project was that the material prepared for the school project has served an additional purpose. It is frequently on display at our department’s open days. Since we have multiple bachelor and master programmes, we wished to present the diverse perspectives on computing by using a single consistent example.

3. THE SCHOOL PROJECT

For our one week CS school project we used the educational Java programming environment Greenfoot. Greenfoot being “a meta framework for micro worlds” [1] allowed us to develop our own “micro world.” As stated above, we developed a Greenfoot simulation of a simplified airport baggage handling system, the Greenfoot Baggage Handling System, GBHS.

3.1 Greenfoot

Greenfoot is an educational Java programming environment which allows for interactive object instantiation and manipulation. Greenfoot is neither a microworld, nor is it merely a programming environment. It is, beyond that, a 2D-graphical simulation framework for discrete event simulations, within which it is easy to develop 2D games and simulations [1, 3].

Greenfoot is particularly well suited for an introductory course on CS, which is intended to cover some issues related to BI, because interactivity permits the exploration and manipulation of scenarios without programming. Furthermore, due to the use of Java as a programming language, Greenfoot scales well and allows for “sufficiently complex examples.” Finally, since Greenfoot is a simulation framework, it allows for the implementation of animated simulations of material and traffic streams.

3.2 Baggage handling

For the school project we implemented a Greenfoot simulation of a highly simplified airport baggage handling system (BHS). For our purpose, many features of a real BHS were not considered. However, we modelled animated passengers who check in their luggage and go to their flights, because we wanted the people to feature in the scenario (Figure 1). Using the example of baggage handling, both computational concepts and business informatics concepts were covered during the school project. An example for a BI issue is the non-technical difficulties and perils of commissioning or replacement of complex information systems. The initial difficulties with the BHS at Heathrow Airport’s Terminal 5 in 2008 give a vivid example.

Figure 1: Greenfoot simulation of an airport baggage handling system

The first exercise was to figure out the problem and to rearrange the layout of the treadmill. This was easily done by drag-and-drop. In a later lesson the pupils designed their own GBHS, obeying some constraints like costs and passage through time. Afterwards some of them told us that they enjoyed these lessons because it was so easy to manipulate the objects in Greenfoot and because they liked to compare with others. While rearranging the GBHS the pupils had to keep track of their changes using Greenfoot’s object inspector.

In contrast, one of the programming exercises was to implement some conditionals such that the luggage was routed correctly by the GBHS.

4. CONCLUSION AND PERSPECTIVE

Our first experiences with Greenfoot simulations of complex systems like the GBHS have been promising. Further simulation scenarios will have to be developed in order to explore the possibilities of including Business Informatics issues in Computer Science courses at school.

5. REFERENCES