InfoTraffic – Teaching Important Concepts of Computer Science and Math through Real-World Examples

Ruedi Arnold
Inst. for Pervasive Computing
ETH Zurich
8092 Zurich, Switzerland
rarnold@inf.ethz.ch

Marc Langheinrich
Inst. for Pervasive Computing
ETH Zurich
8092 Zurich, Switzerland
langhein@inf.ethz.ch

Werner Hartmann
Center for Comp. Education
PH Bern
3012 Bern, Switzerland
werner.hartmann@phbern.ch

ABSTRACT

The use of suitable examples is a key to teach abstract, theoretical concepts. Interactive computer software allows us to use such examples to create attractive learning environments that not only appeal to students, but also enhance knowledge transfer in class. However, developing such highly specialized systems is costly, resulting in only few of these tools being developed for higher education. This article[1] introduces InfoTraffic, a collection of new learning environments to support the introduction of *fundamental* concepts of computer science and mathematics in order to be of long-lived value. We describe the didactical concepts behind the interactive and concrete approach of InfoTraffic, and illustrate them through two of its learning environments – one targeted at propositional logic, the other at queueing theory.

Categories and Subject Descriptors

K.3.1 [**Computers and Education**]: Computer Uses in Education—*Computer-assisted instruction*; K.3.2 [**Computers and Education**]: Computer and Information Science Education—*Computer science education, Curriculum*; G.4 [**Mathematical Software**]: [User interfaces]

General Terms

Human Factors, Design, Experimentation, Theory

Keywords

learning environment, fundamental topics, abstract concepts, propositional logic, queueing theory

1. INTRODUCTION

Do you think of propositional logic when you see traffic lights at a crossing? Or do you reason about what the throughput of a street might be while being caught up in a traffic jam? Most people do not, we presume. But traffic and its control can serve as an excellent example to illustrate some fundamental, abstract concepts of computer science and mathematics.

It is generally accepted today that teaching is particularly efficient whenever students can establish a connection between the subject to learn, and their own experiences in everyday life. This helps especially with introducing abstract and formal topics, and leads to an approach of presenting well-known examples that lead to the more general case. The same idea lies behind the well-established teaching technique of *advance organizers* [2], where the main concepts of new topics are taught *concisely* at the beginning of a lesson, before moving on to the actual details. Advance organizers are particularly valuable if they use an analogy to knowledge or experience from everyday life. This way, learners can not only connect new topics to prior knowledge, but also be less afraid of abstract topics.

Our idea is to go one step further, and not only connect abstract topics to prior real-world knowledge, but to also allow the exploration and immersion into abstract concepts through real-world *applications*. Not only does this enable students to build a bridge form prior knowledge and experience to a new and abstract topic, but they also get to know and experience a practical application for it. InfoTraffic [1] is a collection of learning environments (including teaching materials) for introducing important concepts of computer science and mathematics with the help of traffic control applications. This paper introduces two such environments, LogicTraffic and QueueTraffic, which teach propositional logic and queueing theory, respectively, through real-world application scenarios in the traffic domain. They represent the first set of InfoTraffic applications and have already been successfully used in several high-school classes in Switzerland.

In the remainder of this paper, we will first motivate the choice of fundamental topics for the LogicTraffic and QueueTraffic learning environments – propositional logic and queueing theory – before describing the individual applications in more detail. We then discuss the didactical concepts underlying the set of InfoTraffic learning environments in general, and conclude with reporting on a small set of initial uses of LogicTraffic and QueueTraffic within the context of computer science and math classes in Swiss high-schools.

2. FUNDAMENTAL TOPICS

The two topics we take into account in this article are

SIGCSE'07, March 7–10, 2007, Covington, Kentucky, USA.
Copyright 2007 ACM 1-59593-361-1/07/0003 ...$5.00.

propositional logic and queueing theory. They both appear many times in our daily life and play major roles in computer science. Creating learning environments for such fundamental topics ensures the longevity and broad applicability of the devised systems and their accompanying teaching material, thus justifying their higher development efforts when compared to static, more abstract teaching material.

Even though queueing theory is far from being a staple knowledge among the general population, its effects are generally well-known: We have to wait in queues quite everywhere we go and what ever we do, be it in on the road, in supermarkets, at the post office, or at our doctor. How do queues work? Why do they occur? Or maybe more importantly: How can they be avoided? While not everybody might care for answers to these questions, some people are faced with them every day: the manager of a supermarket deciding how many cash registers to open (or how many salespersons to hire) on which day and daytime; the hospital staff trying to provide expedient treatment at peak times with fewer doctors and nurses; or airport control faced with bad weather and the start of travel season.

Queueing theory is very important in scientific disciplines that require some form of dynamic capacity planning, prediction, or analysis, e.g., network communications, but also logistics. While an in-depth understanding of queues requires a substantial amount of higher mathematics (e.g. calculus and probability theory), an intuitive understanding and approach might be sufficient in many cases, e.g., for directing traffic around an accident, or when staffing point-of-sales during special sales. Having a general grasp of the underlying problem can also ease further comprehension of more advanced models and levels of abstraction.

Logic is even more omnipresent in our everyday life. We all claim to think rationally and argue consistently. Modern information and communication technology would not be possible without logic, both from the point of the underlying infrastructure, as well as for its users. Internet search engines, for example, generally use conjunctions of search terms. Yet even though logic is very much ubiquitous in our discussions and actions, its notions and concepts are often only intuitively clear. Recent studies show that many search engine users end up formulating boolean queries that are syntactically wrong. Similar problems arise with many database-based query systems, such as internet auctions, libraries, or online shops.

Logic is of central importance in science. Logic and mathematics have always been tightly coupled, as it provides mathematics with formal notations and thus with a toolbox for mathematical proofs. Electrical engineering is also very closely tied to logic, as modern semiconductor circuits are essentially nothing more but boolean formulas cast in hardware. The same is true for computer science, where logic is a basic building block throughout. Denning gives five "windows of computing mechanics" in [7], and applications of logic can easily be found for all of them. Two examples for illustration: program verification in the window "computation" builds heavily on logic and the same holds for artificial intelligence in the window "automation".

Despite their obvious importance for scientific disciplines and real-world situations, both logic and queueing theory are rarely taught at high school. One of the main reasons for this is most likely the lack of tangible examples and applicability in the most common (i.e., theoretical) approaches: their abstract and formal presentation renders logic and queueing theory hardly appealing to students (and teachers as well, we fear). Given the importance of such fundamental topics, more practical approaches are therefore needed in order to make such theories more attractive and motivating for students.

3. THE INFOTRAFFIC LEARNING ENVIRONMENTS

The goal of InfoTraffic [1] is to provide a collection of learning environments for teaching fundamental concepts of computer science and mathematics. So far, two programs have been released, together with corresponding teaching material: LogicTraffic and QueueTraffic. They teach the concepts of propositional logic and queueing theory, respectively, and do this in the context of real-world applications in the traffic domain. All of the software described below is available online along with teaching material at [1].

3.1 LogicTraffic and Safe Intersections

The LogicTraffic application (figure 1) offers an introduction to propositional logic (PL) using the example of traffic lights control at an intersection. The main task for the student is to find a formula in PL that makes a given intersection "safe", i.e., which prevents collisions through appropriate signaling on the corresponding traffic lights. With each lane corresponding to a variable, and *true* and *false* indicating a green and red light, respectively, a "safe" formula is one that avoids any two crossing lanes to simultaneously have their traffic lights on green.

The learning targets of LogicTraffic include the basics of PL, and cover concepts such as variables, truth values, logical operators, formulas, equivalence of formulas, and normal forms. By using the intersection/traffic light metaphor, students can immediately connect the concept of PL to real-world situations, and the use of interactivity allows direct feedback (i.e., animated collisions, see figure 2) whenever conflicting assignments are made.

The user interface of LogicTraffic consists of five parts (see figure 1). The central part shows a particular *traffic situation*, representing an intersection with a number of intersecting lanes and their corresponding traffic lights. A *truth table* to the right shows all possible traffic light settings for the intersection, allowing students to manually designate certain combinations as "safe" or "unsafe". Below the truth ta-

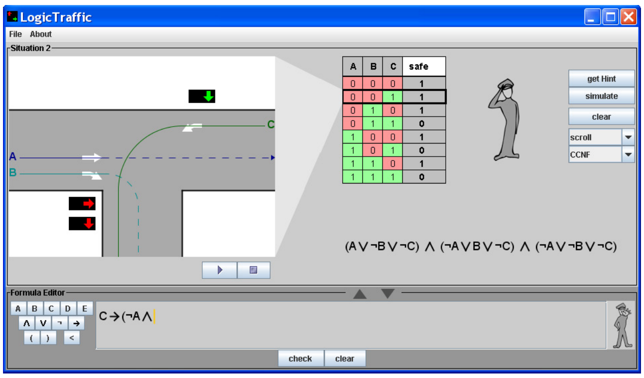

Figure 1: The learning environment LogicTraffic.

ble, a *corresponding PL formula* is shown that summarizes the truth table above, according to whatever combination of "safe" and "unsafe" values the student entered. Instead of manually filling out the truth table, students can also use the *formula editor* at the bottom to directly enter a PL formula and initialize the truth table accordingly. Last not least, a set of *controls* on the right and below the situation allows students to simulate each line of the truth table in the traffic situation window, thus gaining a first-hand understanding of potentially conflicting traffic light combinations.

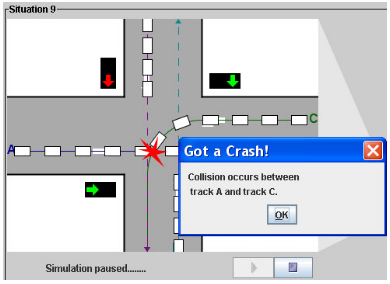

Figure 2: An animated collision in LogicTraffic.

LogicTraffic allows a stepwise introduction to the basic concepts of PL. In a first steps, students can control the traffic lights manually (by clicking on them) and thus gradually filling the truth table. For each possible configuration of traffic lights, the student declares whether the resulting traffic control is safe or not. Using the simulation controls then allows a visual validation of the entered configuration. LogicTraffic comes with a set of intersections of growing complexity, which can be loaded and solved in succession. While a simple intersection with two lanes can still be configured in the manual fashion described above, students quickly realize that increasingly complex situations lead to unmanageably large truth tables, hence a transition to equivalent PL formulas can be motivated. Consequently in the next step, students try to control a given intersection only with the help of a PL formula, without using the truth table. In a last step, students then try to find more compact descriptions of those formulas that describe safe intersection. The controls allow students to transform a given formula into a range of representation, e.g., into canonical disjunctive normal form, which in turn leads to the investigation of the relationship between normal forms and the truth table. A final step thus might involve devising an algorithm that allows for the construction of a formula in canonical disjunctive normal form, given a corresponding truth table.

3.2 QueueTraffic and Traffic Jams

QueueTraffic offers a gentle introduction to queueing theory by letting students simulate and analyze traffic at an intersection. Important system parameters can easily be changed and their impact directly analyzed through simulation and statistical summaries. The learning goals of QueueTraffic include the basics of queueing theory and cover concepts such as throughput, arrival rate, average waiting time, and poisson distribution. The main idea is that students gain an intuitive understanding of basic queueing theory through experimentation with a range of different intersections, by changing parameters like the arrival rate of cars

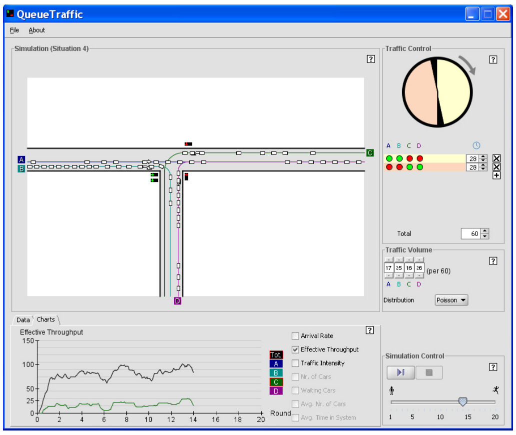

Figure 3: The learning environment QueueTraffic.

for individual streets and the timing of the individual traffic lights, and observing the resulting system behavior.

The user interface of the programm consists of four parts (see figure 3). The central part of the window is again taken up by a view of the *traffic situation*, representing an intersection with its lanes and traffic lights. The *traffic control* panel on the right allows students to define certain phases, i.e., time slots when certain lanes show a green light. Each phase is represented by a combination of red and green signal settings and a duration. All phases together make up a single round of the simulation – the circular diagram in the top right shows the relative length of each phase within a round (the example in figure 3 shows two rounds with equal duration). Also, students can define the arrival rate of cars for each of the lanes. A set of *simulation controls* below allows students to start and stop the simulation, as well as controlling its execution speed. The *data and chart tabs* at the bottom left display statistical information about the current system settings in numeric and graphical format. During simulation, these figures are constantly updated.

In their first steps with QueueTraffic, students are asked to simply define a set of phases, run the simulation, and observe what happens. They can see how cars appear in the system and leave again, and whether queues are built up or not. In a consecutive step, students are taught some basic queueing concepts, such as arrival rate or throughput, and are asked to analyze the given situation and parameters accordingly. Finally, students should try to configure a given system based on their knowledge about queues so that no traffic jam occurs, or so that waiting times are minimized. As in the LogicTraffic application, different predefined intersections can be loaded in order to gradually increase the difficulty level throughout the exercises.

4. DIDACTICAL CONCEPTS

There are a number of important questions to be addressed when implementing a learning environment. We shortly discuss some of them in this section, and point out how they apply to the InfoTraffic learning environments. Specifically, we want to examine topic selection, representation styles, interactivity, and the role of examples in teaching.

4.1 Teaching Fundamental Topics

As we have argued in section 2 above, the development of an interactive learning environment such as InfoTraffic only makes sense and is worth the effort if it covers fundamental topics that ensure its longevity and widespread applicability. Schwill [10] gives a set of criteria for fundamental ideas of computer science, based on Bruner [5]. We summarize them here as follows: A fundamental idea with respect to some domain is a schema for thinking, acting, describing, or explaining which is applicable in different areas, may be demonstrated and taught on every intellectual level, can be clearly observed in the historical development and will be relevant in the longer term, and is related to everyday language and thinking. In other words, fundamental ideas guarantee the selection of content which is cognitively demanding, relevant, and long-lived. Our choice of topics for our initial InfoTraffic learning environments – propositional logic and queueing theory – has been specifically influenced by this line of reasoning.

4.2 Using Different Representations

Abstraction is a useful and powerful method to teach complex concepts, yet it is often demanding and requires significant levels of experience. While we learn over time to accept and understand explanations given to us just in pictures or as text, we started out as children learning from concrete objects and experiences only. Even though after finishing school we are well accustomed to conceptualizing events and procedures solely in our imagination, we still appreciate concrete relevant examples from our everyday life.

Both propositional logic and queueing theory are typically taught as abstract and formal topics. Obviously, knowledge of the underlying formalisms helps to gain a deeper understanding of their respective implications, yet for introducing these topics, less abstract representations might be more advantageous. Does it really help if our first knowledge of logic is that "1" stands for "true", "$\wedge$" for "and", and "$1 \wedge 0 = 0$" holds? Do we really understand the relevance of queueing theory if it is introduced as an M/M/1[2] system? Once students are lost and behind, it is obviously difficult to get their attention back. Finding representations that are more appealing and realistic than abstract concepts plays a major role in the process of teaching complex topics.

Bruner [6] introduced the theory of three basic modes of instruction. They can be seen as one possibility to move away from abstraction and mere formalism in teaching. We summarize those three modes here, and extend them with a fourth *virtual-enactive* mode as introduced in [8]:

Symbolic: Acquire knowledge through symbols (e.g., text or signs). Symbolic representations are concise and are especially suitable if one has already an appropriate intuitive perception. Most people do neither need to climb a tree nor see a picture of a tree to imagine what the idea "tree" means.

Iconic: Facts are represented as pictures. Concrete objects, events or procedures are understandable as visualizations. A hotel brochure or a city map often suffices to get the picture and orient oneself.

Enactive: Learn through your own doing. This is especially the case for children, as they learn through their own actions, grouping of objects, and through observation. Kids need no manual for a tricycle.

Virtual-Enactive: Through manipulation in a software environment enactive processes are simulated. Popular examples are learning environments where students can control virtual robots on their screens.

[2]M/M/1 stands for the Kendall notation of a queuing system and refers to a system with Markovian (exponential) arrival and service time distribution with a single server.

These four basic modes of representations are repeatedly employed throughout the InfoTraffic learning environments. In LogicTraffic, formulas and truth tables offer a symbolic representation, the static picture of a traffic situation corresponds to an iconic view, and through mouse-clicks on traffic lights and animation of the situation a virtual-enactive mode is achieved. In QueueTraffic, numbers (parameters) and data sets give a symbolic representation of the situation, as do the graphical statistics, the static situation visualization, and the phase clock. The traffic situation is again iconic, and through simulation, a virtual-enactive mode is reached.

4.3 Providing Interactivity

Good educational software is usually characterized by a high degree of interactivity. We use a model by Schulmeister [9] that defines six levels of increasing human-computer interaction. Level one means no interaction at all, only the display of information. Level two lets users navigate through the representation of information. Level three offers multiple representations of the content. On level four, the user can modify parameters of the representation. Additionally, on level five, the user can manipulate the content itself. Level six means the user can create and manipulate objects and watch the system react, i.e. gets feedback.

The set of InfoTraffic learning environments have been developed with a high level of interactivity in mind. According to Schulmeister's definitions above, both LogicTraffic and QueueTraffic reach an interactivity level of four, as both allow students to modify the simulation parameters. LogicTraffic additionally offers students to directly manipulate its formulas (i.e., its contents), which corresponds to an interactivity level of five. It further gives feedback about the current safety state, thus reaching level six.

Note that only few computer-based learning environments satisfy the demand for a high degree of interactivity. One reason is the high cost of development. As Berg [3] notes: "Highly interactive software using simulation strategies is almost non-existent in higher education. Clearly the cost of developing such software is a barrier." As we have already pointed out above, educational software that focuses on fundamental ideas of a domain, and consequently addresses different cognitive levels, has the potential to amortize the high cost of development, as the pedagogical concept underlying the software will be of long-lived value.

4.4 Extending the Rule-eg-Rule Technique

The sole existence of an attractive learning environment does of course not improve teaching quality. It is imperative that any such tool is properly incorporated into the classroom, e.g., by providing corresponding teaching materials that outline exercises, preparatory units, and lecture notes. The InfoTraffic learning environments not only come with a set of accompanying teaching materials (see [1]), but also introduce a novel lecture organization, which we call the *eg-rule-eg-rule* technique. Common lectures are often organized based on the *rule-eg-rule* technique, see figure 4. According to Bligh [4] the first *rule* stands for a concise statement, display, and re-expression of the topic. Bligh describes the *eg*-part (example) as elaboration (detail, illustration, reasons and explanations, relate, examples) and feedback. The second *rule* is then a summary with recapitulation and restatement.

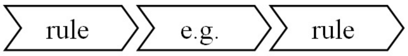

Figure 4: The widely used rule-eg-rule technique for teaching.

For abstract topics, it is often not suitable to begin with a rule, i.e. with a concise statement displaying and summarizing the main concepts in an condensed form. For such cases, we therefore suggest to prefix the initial overview with an introductory example, e.g., based on an everyday life experience. Our extended *rule-eg-rule* technique is thus as depicted in figure 5.

By beginning with a real-world example, students get a first concrete impression of the topic and can relate it to their prior knowledge. This introduction is then followed by Bligh's first *rule*, i.e., a concise statement of the abstract theories and concepts, another elaboration, and the final summary. This *eg-rule-eg-rule* technique can be seen as an extension of an *advance organizer* (cf. section 1), where the main topic is not only summarized, but made concrete by the introduction of an example from everyday life experience.

One main idea behind the learning environments of InfoTraffic was to use a suitable example for the initial *eg*-part that is intuitively understandable and allows different appealing representations. All InfoTraffic learning environments share the common theme of traffic control and its simulation, and thus provide an immediately comprehensible introduction to the individual abstract concepts they teach.

5. CONCLUSIONS AND OUTLOOK

With LogicTraffic and QueueTraffic, we presented two learning environments that teach propositional logic and queuing theory, respectively, through real-world examples. They are part of the InfoTraffic [1] initiative, which attempts to simplify teaching fundamental, abstract topics of computer science and mathematics through highly interactive simulation software based on everyday experiences.

The InfoTraffic learning environments center around the common scheme of traffic control and its simulation, which we found to be a suitable vehicle (sic) to demonstrate a range of fundamental, abstract concepts of computer science and mathematics. Both the topics of propositional logic and queueing theory can be well-motivated with the use of an intersection metaphor. We are currently working on a subsequent teaching environment within InfoTraffic, which will focus on the concepts of dynamic systems and graph theory, again based on examples from traffic control.

We have repeatedly used both LogicTraffic and QueueTraffic at various different Swiss high-schools. Informal student feedback was encouraging, with many rating the programs to be appealing, clear, and concise. During classes, we often observed students using the programs even when they

Figure 5: The eg-rule-eg-rule technique as an extension of the rule-eg-rule technique.

had no more exercises to do, just experimenting on their own. We also presented the two programs in teacher training courses, where many teachers found them useful and deemed them very suitable environments for teaching the respective topics. Teachers especially appreciated the fact that both environments did not require any installation (except for a pre-installed Java runtime environment, JRE), and that both software programs come with a detailed set of teaching materials. While we plan to conduct more formal evaluations, these initial findings have convinced us that the InfoTraffic learning environments work well at school and are met with much interest from both teachers and students.

The development of highly specialized learning environments can be a demanding, tedious, and time-consuming task. It is not trivial to come up with good examples and to find suitable representations and graphical user interfaces that offer high interactivity. However, for *fundamental* concepts, the longevity and broad applicability across a variety of subjects can justify this effort, particularly if the material is widely publicized, well maintained, and readily available.

6. ACKNOWLEDGEMENTS

The authors thank Nicolas Born and Anna-Nina Simonetto for their valuable work and fruitful discussions. Their gratitude also goes to Friedemann Mattern for his continuing support of the InfoTraffic project.

7. REFERENCES

[1] Learning environment InfoTraffic, online available along with teaching material. http://swisseduc.ch/compscience/infotraffic, 2006.

[2] D. P. Ausubel. The use of advance organizers in the learning and retention of meaningful verbal material. *Journal of Educational Psychology*, 51:267–272, 1960.

[3] G. A. Berg. The Big Questions. *International Journal on E-Learning*, 1(2):5–6, 2002.

[4] D. A. Bligh. *What is the Use of Lectures?* Penguin Books, Harmondsworth, England, 1972.

[5] J. S. Bruner. *The Process of Education.* Harvard University Press, 1960.

[6] J. S. Bruner, R. R. Oliver, and P. M. Greenfield. *Studies in Cognitive Growth.* John Wiley and Sons, New York, 1966.

[7] P. J. Denning. Great principles of computing. *Communications of the ACM*, 46(11):15–20, 2003.

[8] W. Hartmann, M. Naef, and R. Reichert. *Informatikunterricht planen und durchfuehren.* Springer, Berlin, 2006.

[9] R. Schulmeister. Taxonomy of Multimedia Component Interactivity. A Contribution to the Current Metadata Debate. *Studies in Communication Sciences. Studi di scienze della communicazione.*, 3(1):61–80, 2003.

[10] A. Schwill. Fundamental ideas of computer science. *EATCS-Bulletin*, 53:274–295, 1994.