

# Problem-based E-Learning in Practice: Digital Laboratories Provide Pathways from E-Science to High Schools

Hans Hinterberger  
Information Technology and Education, ETH Zurich, Switzerland  
hinterbe@inf.ethz.ch

**Abstract:** Our digital age requires competent data management and critical information appraisal. This paper claims that computer science (CS) should be given the same respect in education that mathematics, physics and chemistry enjoy. Turning the computer into a digital laboratory by combining problem based learning (PBL) with E-tutoring allows students to learn CS skills by independently creating information out of raw interdisciplinary data. This technique allows a transfer of CS curriculum from the first year of university to high schools, which could free up university time for discipline-specific CS learning.

## 1. All enabling sciences deserve prime time in education

Mathematics is called the "queen and handmaiden of science" partly because it is the most fundamental *enabling science*. Consequently, during the course of their compulsory education students are trained to master mathematical concepts which increase in complexity with each level of schooling. Nobody questions the necessity of requiring basic mathematical competences. But why do we not require a structured education in computer science (CS) even though mathematics and CS enjoy a privileged partnership to the degree that some would call CS the "princess and handmaiden of science and everybody else" because of its ubiquitous *enabling technology*? Perhaps this gap has led to the apparent lack of pedagogical strategy for CS; a deficit which is even stranger because none of us can avoid becoming involved in the intricacies of the digital world.

CS has become as indispensable as mathematics and deserves to be treated with the same respect in education.

Mathematics allow us to formulate *abstractions* and find *relationships*. These capabilities are not reserved for mathematicians; anyone can apply them to everything from the little challenges of daily life to complex problem-solving on the job. We argue that *critical thinking* is to CS what abstraction is to mathematics, and the counterpart to finding relationships is the *generation of information from raw data*, which is the basis of critical thinking. Mathematical skills by themselves do not provide competence in vital computer-supported technologies. To effectively meet the challenges of the 21<sup>st</sup> Century everyone needs at least some data and information processing skills.

But how can these skills be taught so that even students with an aversion to working with computers can be engaged?

Initial inquiries during a current project to define a strategy for CS in education in Switzerland have shown that worldwide there are apparently no solutions which could seriously serve as best practice examples for systematic CS education from elementary school to university. This is a serious cause for concern, considering that the well-being of entire economies is at stake. As recently as 2009, for example, the U.S. National Science Foundation issued a call for research proposals under the title: "CISE Pathways to Revitalized Undergraduate Computing Education (CPATH)". The projects ought to "[...] *transform undergraduate computing education on a national scale*, to meet the challenges and opportunities of a world where computing is essential to U.S. leadership." Similarly, in Europe a 2010 Education Ministerial Meeting of the Organization for Economic Co-operation and Development (OECD) with the theme "Investing in Human and Social Capital: New Challenges" included the agenda items "*Matching skills to new needs*" and "*Equipping effective teachers for the 21st century*".

This paper makes a case for a competence-oriented approach to teaching computer skills that allows the curriculum to be moved from the first year of university to the senior or even junior years of high school. The line of reasoning is based on 12 years of teaching bachelor-level introductory CS for the natural sciences at ETH Zurich (Hinterberger, 2010a and 2010b)<sup>1</sup>. What I learned during this time is that computing cannot be dealt with in isolation; my teaching efforts only bore fruit after I integrated CS topics with state-of-the-art, practical research.

Section 2 describes our E-Learning-based pedagogical approach, and Section 3 discusses how professional practices in E-Science influenced our choice of course content. Section 4 shows how students can get insight into the complex processes of *generating information from raw data*. In Section 5 we argue that a scientifically founded introduction to CS should already happen at secondary school and show that this would free up substantial resources at the university level. Section 6 reviews some issues that often lead to disappointing E-Learning experiences and points to practices that may be more satisfying.

## 2. Digital laboratories help to teach CS

There are many ways to teach basic CS skills, but which strategies are more sensible? Over the years we observed that efforts to define a body of course content for CS in general education ended up time and again in discussions about what CS is, ad hoc designs of curricula and uncoordinated definitions of course content without a comprehensive strategy. We argue that it is more effective to focus attention on *competences* which students can develop in an active learning environment. In other words, it is useful to distinguish between treating the computer as *medium* or as a *tool* on one hand and as a *subject matter* on the other.

When learning mathematics, physics and chemistry, students traditionally engage in practical activities. Nobody would expect to become proficient in applying basic scientific concepts only by reading or hearing about them. Don't we all remember that these topics first became interesting when we were able to test our skills with real world phenomena and problems that actually made sense in the context of our daily lives? Furthermore, the physics and chemistry laboratories were perhaps the most interesting rooms in our schools. Schools also have so-called computer labs that typically host a collection of networked PCs. Yet these labs can only become effective if, after starting a computer, students find themselves in a "digital laboratory" in which they can experiment with the CS-concepts that they must learn. CS, with its invitation to actively create, lends itself to a *constructivist* learning approach, in particular *problem-based learning* (PBL), a teaching method devoid of lecturing.

### Problem-based learning

PBL, a time-proven pedagogical method based on active self-teaching, is increasingly used when competence must be acquired. Princeton University, for example, introduced PBL to tackle problems they faced with their introductory science courses. In effect they moved the teaching from the lecture hall to the laboratory where students spend their time solving realistic problems from different domains under the guidance of tutors. (<http://www.princeton.edu/pr/pwb/07/0312/1a.shtml>).

The expert knowledge of the PBL instructor is conveyed through tutoring (face-to-face instruction) of one student or a small group of students. Tutorial learning is completely active for the students. Learners work with the tutor as long as it takes for them to completely understand the concepts. In this way, a lesson unit comprises a constant amount of material but the time a student uses to absorb the material can be adjusted to fit the student's capabilities. The traditional lecturing environment lacks this flexibility.

Ted Oppenheimer observed: "In study after study, whenever tutoring is matched against some competing pedagogy, including technology, tutoring wins handily" (Oppenheimer 2003).

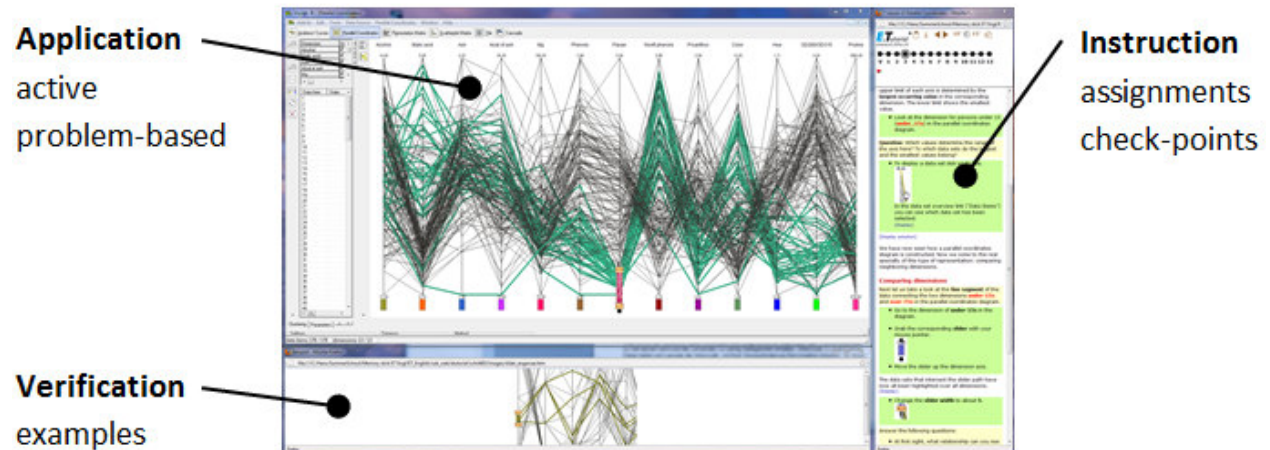
### The computer as tutor in a digital laboratory

In its original form PBL asks learners to find a solution to a physical problem in a physical laboratory. When PBL is transported to a digital laboratory, students can profit from the computers' ability to carry out several activities apparently simultaneously. Acting as a *virtual tutor*, a computer can give a learner an assignment for a new task in

---

[1]Part of this paper appeared first in this publication in German.

one window (an instruction window) while the student still sorts the entries of a table in another window (an application window). One can imitate a human tutor even better with a third window (a verification window) in which learners can call for confirmation that they are on the right track and get help when the need arises. Figure 1 shows a typical screen layout of three windows used for E-tutored PBL.



**Figure 1:** Computer-supported tutoring is implemented with three separate but logically connected windows. In the *application window* learners actively solve a given problem with real software applications. The *instruction window* lists brief assignments with which the virtual tutor guides students through a realistic problem-solving process. The set of instructions is sprinkled with check points that include guidance aids which can be called up in the form of examples shown in the *verification window*.

In 2000 at the Department of Computer Science of ETH Zurich we began to develop computer-supported tutorials, called *E.Tutorials*<sup>®</sup> to teach introductory CS for the natural sciences. Faessler et. al. (2006) is one of several publications reporting our experiences. We noticed that *E.Tutorials*<sup>®</sup> positively influence the teaching in two ways: a) the CS concepts presented are learned in more depth and b) the problem-based learning environment connects these concepts with other subjects.

Tutorial E-Learning in a digital laboratory is an ideal form for teaching computer systems, software, data, and information skills because it provides a learning environment that motivates students (Faessler et. al. 2006). An additional benefit *E.Tutorials*<sup>®</sup> offer is individualized instruction during asynchronous learning, which equalizes differences in previously acquired knowledge and performance speed that hinder classroom teaching. Trial *E.Tutorials*<sup>®</sup> are available on the website *Computer based Tutoring & Assessment* ([www.cta.ethz.ch](http://www.cta.ethz.ch)).

### 3. E-Science provides form and content for PBL

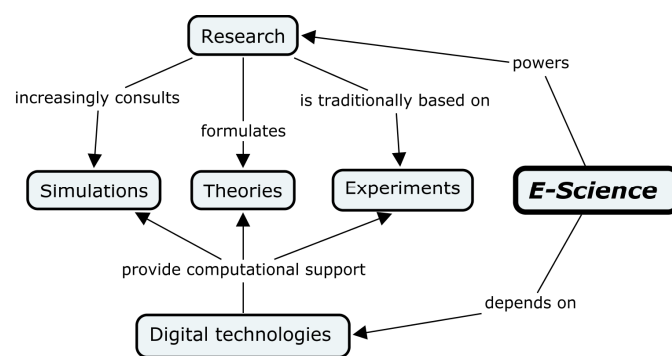
The analogy between CS and more traditional sciences helped us to find the PBL teaching methodology. The similarity also carries into the discussion about which topics to teach. For example, mathematicians give their students interesting engineering problems because they call for an understanding of the underlying physical principles which in turn lead to the relevant mathematical theories. Correspondingly, when the computer is the subject matter, it is best to embed the target CS concepts in a realistic, practical context so the skills and theory are self-evidently worth learning.

PBL stands or falls with the choice of problem. It succeeds with interesting, realistic problems; it fails with artificial problems for which learners cannot muster motivation. At a research university there is no excuse for artificial problems because instructors can readily find interesting and realistic projects based on actual work carried out by themselves or by their colleagues. Section 4 illustrates the kind of case studies we developed. But first we look at how contemporary research practices can guide curriculum development. Like the mathematicians using engineering

problems to cultivate understanding of mathematical theories, we found it helpful to look at the concepts behind modern methods of scientific research when considering content for our CS courses. As we looked for models for the design of our courses we were attracted to E-Science, a new scientific paradigm focusing on data-intensive systems and scientific communication.

### E-Science

An essential component of research has always been experimentation, an activity that continuously profits from technological inventions. When early electronic computing devices opened the way for numerical methods in mathematics, researchers began to use simulations to complement physical experiments. The digital revolution, triggered by the commercial proliferation of microprocessors, gave birth to further computational methods and data-intensive systems which support all areas of research with digital computers, leading to a flood of complex data and a completely new paradigm called "E-Science" or, to adopt a more neutral position, "the Fourth Paradigm" (Hey et. al., 2009). The elements defining this term and their relation to each other are illustrated with the concept map in Figure 2.



**Figure 2:** The four paradigms of research. Empirical research based on experiments and simulations help researchers find results that are the basis to formulate and verify theories. E-Science comprises data-intensive systems and scientific communication to provide a comprehensive digital infrastructure for all research activities.

E-Science describes the new scientific practices arising from the capabilities of what we call the *information workplace*: the digital infrastructure and desktop computing tools that are now ubiquitous. E-Science offers platforms for many scientific tasks, including

- data acquisition
- data storage
- data management
- analysis (including visualization, mathematical modeling, simulation)
- programming
- collaboration
- publication of results
- sharing of raw data

The shift to PBL creates a shift in CS learning paradigms: as the E-learning platform moves data and data processing to the center of attention, discussions about programming languages and software paradigms become secondary. In our opinion, using E-Science as a base for planning a PBL curriculum for CS offers three particularly interesting themes: (a) the *digital infrastructure*, (b) *complex processes* and (c) *enriched E-Learning*.

- (a) **Digital Infrastructure.** Direct practice on real problems with current software builds portable, integrated data-handling skills.
- (b) **Complex processes.** An ability to evaluate the interplay between cause and effect and an understanding of how data systems function should be a part of today's general education. Step-by-step E-guidance through

multi-phase analytic procedures, multi-platform collaboration and public data mining leads to mastery of the complex skill sets required in many walks of life.

- (c) **Enriched E-Learning.** Flexible E-tutored platforms can be used in many modes of teaching, adding value to classrooms and distance learning from basic to advanced levels.

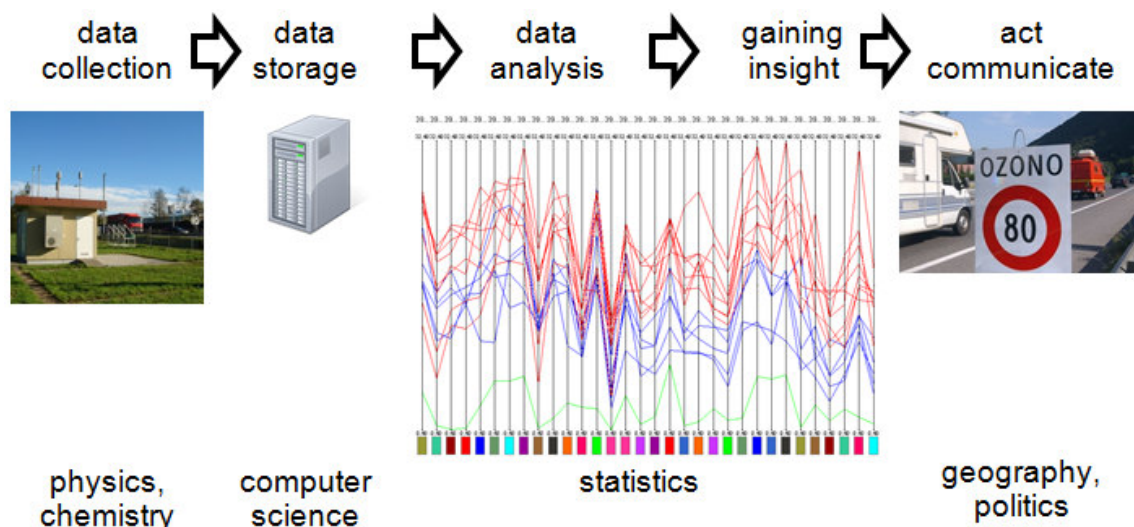
#### 4. Gaining insight into complex processes: a fundamental CS-competence

Not only can computers realistically model complex, interdisciplinary working environments for educational purposes, they can also help students learn how to manage this complexity, especially when instructors make use of data collections created in research environments and provided in an *open data* context. This ensures a feeling of reality and demonstrates the role that collections of raw data play in decision making. In this way, teachers additionally raise the learners' awareness of the value of intellectual property in the form of data collections. Therefore, today's schools should, in addition to printed literature and E-Learning materials, also provide access to real data collections.

"Subprime-loans triggered the financial crisis in 2008"; "The prices for medication are at the root of the exploding health care costs"; "A powerful thunderstorm was responsible for the crash of Air-France flight AF 447". Reports of this kind throw us into a sea of information and usually we accept these "facts" although we may have doubts or difficulties in understanding the material. Who could competently analyze and evaluate all the reports and arguments? Most of the time we rely on gut feeling, but would it not be illuminating to evaluate the original information directly? To learn how raw data can be transformed ("cooked") into useful information, students retrace the processes underlying a given argument so that they can independently derive the information on their own. This skill improves their ability to evaluate arguments and to present their own information convincingly.

##### Practical example of E-learning: speed limits

One of our E.Tutorials<sup>®</sup> investigates how public authorities arrive at the decision to restrict the maximum speed allowed on certain highways to 80 km/h. The process which controls this regulation originates with the regional concentrations of pollutants in the air, particularly the gas ozone. The necessary measurements are collected centrally, processed and stored on a server accessible over the internet, allowing students to transfer the raw data to a computer for analysis as shown in Fig. 3. This procedure, however, requires knowledge of internet protocols, familiarity with spreadsheet program functions, and comprehension of statistical methods of data analysis. When the findings gained through the data analyses are evaluated together with geographical information students can understand the political regulations.



**Figure 3:** From data to information by example of the analysis of air quality data: The computer spans at least *six domains*, from atmospheric physics to politics. The entire process can be managed with an information workplace as defined in Section 3. (Diagram by B. Scheuner).

Case studies of this kind not only provide an interesting context for CS concepts, they also clearly illustrate the interdisciplinary character of today's decision making processes (see Fig. 3). The varied applications of the computer not only make CS a core discipline, they also demand that CS be recognized as a cross-domain discipline.

## **5. Basic CS curriculum can be shifted to secondary schools**

The definition of CS curricula is a persistent challenge. Schools are looking to universities for advice and leadership. Responding to this call, the Chair of Information Technology and Education at ETH Zurich started a project in 2008 to establish which conditions must be met to transfer the basic CS curriculum of a research university to high schools. This effort is supported financially by the Rector of ETH. Specifically, the subject matter from the first two university semesters was transferred to the junior and senior years of a high school (Raemibuehl Gymnasium, Zurich) including:

- Communicating and publishing over the internet
- Programming simulations with spreadsheets
- Macro programming with VBA
- Introduction to programming with Java

The first experiences of both the university and the gymnasium have been positive (Faessler 2009). We attribute this successful transfer in large measure to our efforts to introduce instruction-process analyses in order to verify that the concepts taught are actually "absorbed" by motivated students (Dahinden & Faessler, 2011).

We have argued above that the enabling technology provided by CS is so ubiquitous that the digital skills it requires must be taught beginning at least in secondary school so that anyone pursuing a professional career is equipped with 21<sup>st</sup> century competence. Unfortunately, the reality is that students entering university still lack sufficient competence to effectively make use of information and communication technology (ICT) applications and still can't program a computer. Again, this seems odd because it would be unthinkable to introduce basic mathematics, statistics, physics, etc. only in university.

This deplorable situation is further aggravated by the fact that today's natural science students desperately need more time to become familiar with the methods and techniques of E-Science. Spending this time on the basic elements of CS is irresponsible.

High schools in Switzerland—and certainly in many other countries—are richly equipped with computers that are typically underused because CS is not a mandatory subject of the core curriculum. Many teachers, however, would be happy to use these devices if they were given the mandate to do so and be provided with sound teaching materials. Training in basic CS competences fits well into teaching programs at the secondary level (comparable to algebra). As a consequence, high school classes would become more interesting and teachers at the next level would be given more time to teach specialized knowledge.

### **The bonus of freeing up university time**

The educational gain of shifting basic CS skills to the secondary school curriculum can be quantified in dollars. Real total tuition cost figures for university students in Europe are difficult to come by, yet they are comparable to the American numbers. Thus to examine the situation in Switzerland, we take \$36,390, the yearly tuition fee (in 2008) of the Massachusetts Institute of Technology (MIT). To measure the amount of education, we use the European Credit Transfer System (ECTS) credit points (CP), a standard which has been established across Europe. Students are expected to complete 60 ECTS CP per year (a total of total 180 CP for a bachelor degree, 90 CP for a Master). One ECTS CP therefore corresponds to \$ 606 in tuition cost.

Now we can calculate the value of the time freed in a Swiss university when we move the introductory CS courses from the bachelor program to the high school. Natural science degree programs at ETH Zurich include on average 5 CP in introductory CS lectures. Now, if universities are freed from these courses they can invest the time gained in discipline-specific course work rather than elementary CS classes. Such a transfer both enriches high schools and results in a potential gain in quality across all university degree programs in Switzerland that can be expressed by a sum of close to \$ 100 million, figuring 5CP at \$ 606 per CP, times 32'700 students who finished high school in Switzerland in 2008 (Statistics Switzerland).

We remind the reader that this impressive sum already exists; it economically quantifies learning content which can be moved within the educational system to bring about a multi-million dollar gain in educational quality.

## 6. Conclusion: prospects and risks revisited

Those who plan to use computer supported teaching in high schools or universities face some known risks:

The computer ...

- ... brings with it an operating expense which is often underestimated. As a consequence, the financial and personnel support necessary for a functional, stable digital infrastructure is often missing.
- ... can lead to a digital divide where teachers are split into two camps: those who accept the digital challenge, and those who decline it.
- ... makes educational content "faceless" because the instructor remains hidden behind a computer delivery tool, challenging his skills to connect with the students.
- ... can overwhelm learners if they do not get appropriate, goal oriented support for complex sequences of operations.
- ... can under challenge learners if the device is used only superficially or as a medium for entertainment.
- ... tempts learners to disclose personal and confidential data in social networks, social structures that also lead to "faux relationships over the internet" at the expense of interpersonal relationships.

These risks are balanced by the potential advantages of well-designed and executed computer-based instruction:

### Prospects

The computer ...

- ... challenges teachers to formulate problems in a timely and realistic context so that schools can offer instruction appropriate for a knowledge based, digital society.
- ... combined with suitable E-Learning methods allows a high flexibility so differences in levels of knowledge and learning speed can be equalized.
- ... in connection with the "Open Educational Resources" movement can further democratize intellectual properties, demonstrating the benefits of E-citizenship in many walks of life.
- ... provides a digital laboratory for research practices adopted from the E-Sciences can be explored hands-on, as a complement to lectures in theory, improving transfer and retention.
- ... unites the digital links of the processes of knowledge acquisition, making them visible and workable, thus improving critical thinking skills.
- ... invites students to creative action.

The need for a comprehensive education in CS is self-evident. Our experiences suggest that CS skills are best taught by focusing students on practical, realistic information tasks and tutoring CS concepts and methods in that context. The center of attention should not be on the technology, rather on continuous evaluation of the quality and meaning of the teaching and learning experience. Universities and high schools should work together to build a coherent pedagogical infrastructure for CS similar to the mathematics curriculum that we take for granted.

## References

Dahinden, M. & Faessler, L. (2011), *Monitoring Blended Learning Environments Based on Performance Data*. IADIS International Conference e-Learning, Rome Italy.

Faessler, L., Hinterberger H., Dahinden M., Wyss M. (2006), *Evaluating student motivation in constructivistic, problem-based introductory computer science courses*. AACE World Conference on E-Learning, Hawaii.

Faessler, L. (2009), *Einsatz von ETH-Lernmaterialien im gymnasialen ICT-Unterricht im Rahmen der informatischen Grundbildung*, Technical Report 641, Departement Informatik, ETH Zürich.



Hey, T., Tansley, S., Tolle, K., eds. (2009), *The Fourth Paradigm, Data-Intensive Scientific Discovery*. Microsoft Research.

Hinterberger, H. (2010a), *Making Informatics Work for Everyone—teaching Computer Competences for the Natural Sciences at ETH Zurich*. Technical Report 686, Department of Computer Science, ETH Zurich.

Hinterberger, H. (2010b), Das Potential des Computers im Schulunterricht in P. Aerni, F. Oser (Hrsg.) *Forschung verändert Schule: Neue Erkenntnisse aus den empirischen Wissenschaften für Didaktik, Erziehung und Politik* , Seismo Verlag.

Oppenheimer, T. (2003), *The Flickering Mind: The False Promise of Technology in the Classroom and How Learning Can Be Saved*, Random House.

## **Acknowledgements**

The author is indebted to Sarah Shephard from the Educational Development and Technology Unit (LET) of ETH Zurich and to Abbott Chrisman for their thoughtful and helpful comments and suggestions.