Proceedings of the 8th International Conference on Applied Informatics
Eger, Hungary, January 27–30, 2010. Vol. 2. pp. 139–148.

Teaching Computing Across the Disciplines

"The different shades of grey"

Malgorzata S. Zuberek

Department of Computer Science, Memorial University, Canada
e-mail: malgosia@cs.mun.ca

Abstract

Computing has become an essential and pervasive problem-solving toolset, inextricably linked to other disciplines. Increasingly, solving real world problems requires an approach from outside the scope of one discipline. In order to meet challenges and opportunities of the global economy, computing education is undergoing continuous changes. Yet, the efforts of the last decade, in particular the attempts to "plant a larger crop of e-skilled graduates" were not terribly successful. Setting educational goals in a windstorm of rapidly changing landscape presents no doubt a colourful conundrum. A novel approach to computing education is now on the horizon, and the shift in teaching paradigm is fundamental. In the new model students shall be exposed to thinking 'outside the box', and graduates shall be prepared to tackle complex, real life problems right away, between, across and beyond disciplinary boundaries, as a reflective contextual practice. In this paper the nature of the new approach is discussed, its promise, fun and potential.

Keywords: Education, Innovation, Experimentation, Curriculum, Teaching Methodology, Cross-disciplinary Teaching, Problem-solving, Computational Thinking

Categories and Subject Descriptors: K [Computing milieux]; K.1 Computer Industry, K.3 [Computers and Education], K.4 Computers and Society, J [Computer applications]; J.7 [Computers and other systems]

1. Introduction

Contemporary computing offers intrinsic excitement which is difficult to match in other disciplines. The excitement comes from the intellectual challenge of solving problems and the engineering challenge of building things that work [3]. Indeed, breath taking news come up almost every day: the very first experiment with bacteria whose 'father' was the computer, the remotely controlled robot-submarines fighting the deadly oil leak at the bottom of Gulf of Mexico, the digital camera that erased entire photo industry almost ‘overnight’, and the increasingly more sophisticated cyber attacks that frighten security of national infrastructures, are just a few extreme examples.

The excitement is there, so are the jobs. And, e-jobs are among the most lucrative on the market. Despite all this, computing is not attracting enough students. Where have all the students gone? [Fig. 2] Clearly, there is a shortage of e-specialists and the forecast is not very promising [Fig. 1]. The job market is getting ‘hungry’. The trends are similar also in other countries, including the EU. Will the new approach to computing education be a panacea? Will it bring a renaissance to computing?

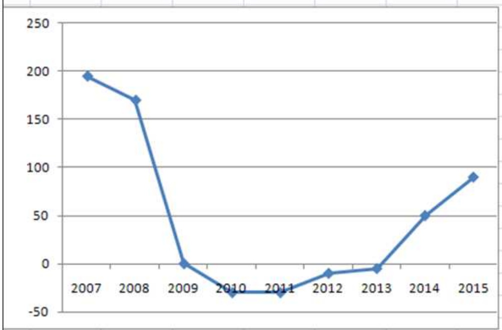

Figure 1: Recent trend and forecast in excess demand

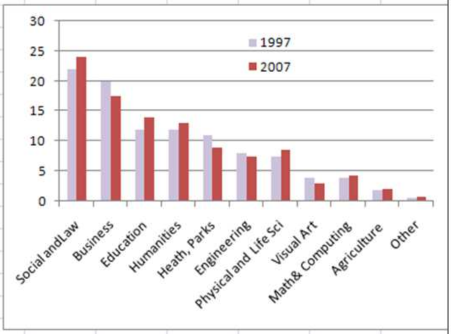

Figure 2: Change in University degrees by field of study

2. Computing in Perspective

Computing is a creative activity that draws on a wide variety of fields, such as natural science, mathematics, engineering, social sciences, business and arts. As such, computing embraces competences that lie at the boundaries of professional practice.

According to ACM/IEEE Computing Curricula [1], computing is a broad multidisciplinary field that extends well beyond Computer Science. Depending on a context, it may include partner disciplines, such as Computer Engineering, Software Engineering, Computational Science, Information Systems, Information Science, Information Technology, and other possibilities still exist. (In the European context, the field of Informatics perhaps would be the best match). In this paper the term computing is used in its broadest, as defined here sense.

The spectrum of applications involving computing is very widespread, like the Cyberspace is, and the potential of computing is unimaginable. This has well been captured in the anecdotal polemic of the two famous scientists (quoted below).

In 1971, American Mathematician Richard Wesley Hamming unveiled a ’bit’ of his wisdom by saying “The purpose of computing is insight, not numbers”.

A riposte came soon after from George Elmer Forsythe, a founder of the Computer Science Department at Stanford University who concluded: "The purpose of computing numbers is not yet in sight".

Nowadays our troubled students dream loudly: "If computers could only think computationally or at least show some sense of humour our graduation day will be in sight".

3. Why to teach non-majors about computing?

Computing is a critical multidimensional field that contributes to discovery, security and competitiveness in all research. According to the US forecasts, the most scientifically important and economically promising research frontiers in the 21st century will be conquered by those proficient with skills in advanced communication technologies and in scientific computing. Hence, studying computing sounds after all like a good investment.

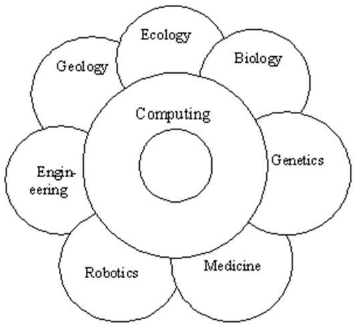

Figure 3: Rosette Commons

While most of today's graduates are equipped with body of the knowledge and skills from a single discipline, the real world rarely conforms to disciplinary boundaries. As such computing requires integration of multidisciplinary domains. Examples may be brought from virtually any area:

- Modern Biology, has transformed from an insular entity into a modern interdisciplinary science, which in turn requires interdisciplinary and cross-disciplinary training for future work force in life sciences [5].
- Medical doctors need to learn data mining techniques in order to extract information hidden in electronic health records, and computer scientists need to collaborate with radiologists while creating medical modelling and imaging software for hospitals, medical schools and research organization.
- Ethics specialists must combine their understanding of the totally new world created by computationally powered advances in biomedicine and bioscience

technology with the moral judgement when setting up socially acceptable boundaries.

- In laboratories Nanos, a tiny computer-chip operated machines, have to be programmed in order to perform life saving tasks, such as unclogging blood vessels, or destroying a tumour without opening the body.
- In liberal arts, software tools are widely used for creation of artistic masterpieces, like paintings, music, photography or movies. And, game development for multi-billion dollar industry involves much more than a massive programming endeavour.
- As well, financial markets could not operate without large scale computing technology. In this extremely complex and dynamic environment, business users continue to demand accurate modelling, rapid implementation and delivery without delay.
- And finally, the miracles like Google, YouTube and Facebook would not exist without specialists who possess knowledge and skills not only in digital communication and networking but also in artistic design and visual aesthetics. [Fig. 4]

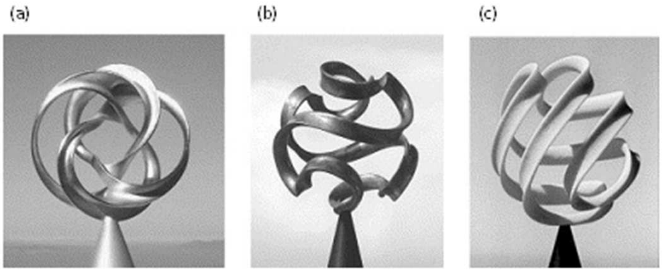

Figure 4: Human perception of beauty is not the same on screen and in nature [7]

4. Reconstructing the Teaching Paradigm

As computing technology continues to encroach upon almost every domain of our lives, computing education needs to integrate more than ever key computing concepts into non-computing disciplines. Great majority of college and university courses are taught nowadays from a specific disciplinary focus. While such model proved to be useful for introducing core concepts, there is a growing need to integrate information from several different disciplines. Many of tomorrow's careers demand not only depth of knowledge in a specific area, but also a general sense of vocabulary, tools and concepts across other disciplines.

The context in which university graduates function today has widened significantly. We witness emerge of numerous interdisciplinary programs in recent years.

New bio-programmes, like biophysics, biochemistry, biomedicine, bioinformatics, spring up under a common umbrella of super-discipline Life Sciences. Yet, most of these programmes are actually integrated studies that offer mixed-discipline major by allowing students to customize their academic programmes through selection among existing courses. The difficulties of offering academic programmes of a mixed nature are particularly acute at the undergraduate level, since the traditional organizational structure at universities is uni-disciplinary.

The new approach to computing education is much more radical. The idea is not only to introduce computing courses into majority of academic programmes, but to weave computing concepts into individual courses, where ever feasible (like mathematics, physics, chemistry, arts, business or law), at the junior level in particular. This also implies resurrecting and revamping Computer Science curriculum in high schools. It is a gigantic endeavour!

While the new trend to teach computing concepts in virtually all academic programmes is spreading out, some leading universities world-wide have already modified their general educational goals to address today's intellectual challenges. Increasingly, concepts and skills that are common to computing as a whole are being introduced to core requirements. But what are they? There is by no means a single view on this issue. The list of fundamental knowledge and skills most graduates should possess in order to become competent users of computational systems from our perspective is as follows:

- Ability to tackle effectively systemic problems and to make informed algorithmic and data structure choices
- Knowledge of how to systematically organize and manipulate data and reason about data (issues covered largely by Information Science), including familiarity with database systems and basic data mining techniques
- Familiarity with information privacy, security and professional ethics, in addition to effective communication and collaborative skills
- Competence in using existing software tools, as well as ability of developing simple computer programs using a modern multi-paradigm programming language such as, for example, Ruby or Python
- Understanding of computer modelling and simulation, as well as familiarity with visualization tools, in order to study further implication of the defined interactions.

This set of topics can also serve well as a core background for a Minor in Computer Science. When complemented with courses of cross-disciplinary nature, team-taught perhaps, and/or courses offered by other departments, the body of the knowledge and skills the graduates gain appears to be sufficient for collaborative computational work on interdisciplinary applications. Still, there is a debate regarding material in the last two points of the above list, in particular computers programming, as some teachers perceive this material to be too difficult for junior non-majors. I would argue against. Most students find computer programming very challenging, this is quite true, but modern programming languages are increasingly more user-friendly. Developing some programming skills helps not only to understand computation better, but it also moves a computer user onto entirely different level. Modelling and simulation, on the other hand, can be introduced in a gentle way and can make computing experience colourful and attractive. Modelling and simulation comprise knowledge that, when combined with visualization helps to develop an understanding of how parts of the system interact, and in this sense it plays an important role. Moreover, simulation is believed to be a laboratory of all future scientists.

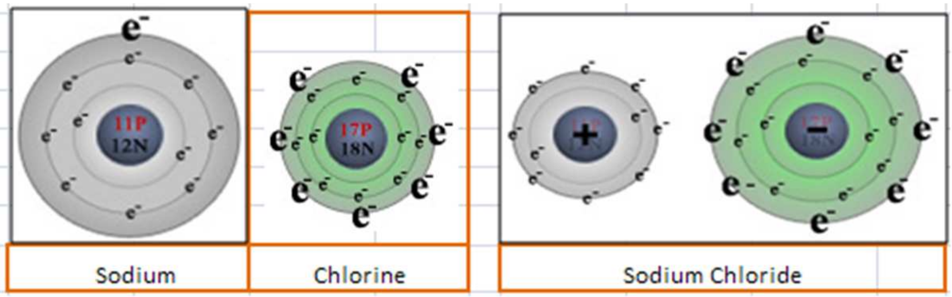

Figure 6: Teaching chemical reactions with animation software. [9]

Visualization has been an effective way of communicating both abstract and concrete ideas. It is also a fantastic teaching tool used in many disciplines, in particular in medicine. As an important element of computation, it brings many quantitative facts to life. Algorithm visualization, for example, is a handy, easy way to understand how algorithms work [Fig. 7, 8].

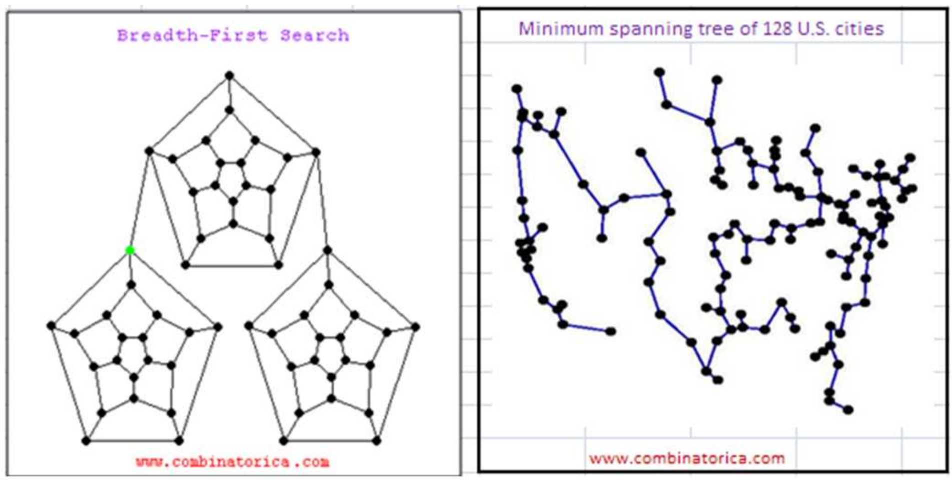

Figure 7: Understanding algorithm design and execution through visualization

5. Green Light for Computational Thinking

"*Computational thinking*" – the mindset that students need to acquire in order to work effectively with computational systems - has been identified a fundamental skill in contemporary times. In practical sense computational thinking translates into an understanding of computational processes, not in any programming language but conceptually. It may have many flavours. Computational thinking requires quantitative reasoning, analytical thinking, and multi-scale modelling. As such, it is very useful in problem decomposition, development of a strategy and investigation whether produced answers are correct. It also helps in mapping domain-specific terms into computing concepts [Fig. 5], as well in observing how viewing information in different ways leads to new interpretations.

Multilevel Abstraction	Biological Databases, Protein networks
Interaction, Recursion, Backtracking	Gene and Protein networks
Probability Models	Hidden Markov Models
Search Space and Complexity	Sequence alignment
Permutation and Significance	Over-representation of genes in pathways
Computer Graphics	Protein structure visualization
Simulation (and Modelling)	Homology Modelling, Mutation in genes
Information Visualization	Exploration of time series data
Optimization, Clustering	Gene expression profiling, Hierarchical clustering

Figure 5: Sample mapping between computational thinking and Bioinformatics [5]

Computational thinking has been declared a common trust at National Science Foundation in the US, and is on the move from K12 through college. It has been recognized a necessary prerequisite on which problem-solving skills can be built. However, it appears to be a strange misfit between the way logical reasoning is taught and the way our minds learn and work. Intensive efforts are underway to identify methods of teaching computational thinking to a broader range of undergraduates.

Computational thinking may also serve as a common language when people talk about computational problems. The techniques people use in thinking about problems are basically the same across disciplines. Anyone who can think in terms of some standard computational techniques should be able also to communicate easily with other people from distant disciplines.

6. Problem-Solving Skills is a Problem

Many employers, technology based and data intensive companies, lament about students' diminishing problem-solving skills. As Tenenberg/Fisher noted, teaching in higher education is governed by a central irony. Graduates at the work place cannot demonstrate any strength in theory and methods of the profession, nor in practical training or experience in the actual area of study. Teachers in defence (including myself) blame basic math and logical thinking skills that students are lacking. The third party, the students, feel that they are fully competent computer users, but they have actually only a superficial button-pressing comprehension. At the same time, it has been increasingly more difficult to convince students for doing any math in head or on paper. When challenged to think, they say, they don't have time to think. The painful truth is that they simply cannot think anymore and therefore cannot solve problems. Students opt to sit rather in biology class than in computer science class (with a genuine trust that biology is the only science in which multiplication means the same thing as division).

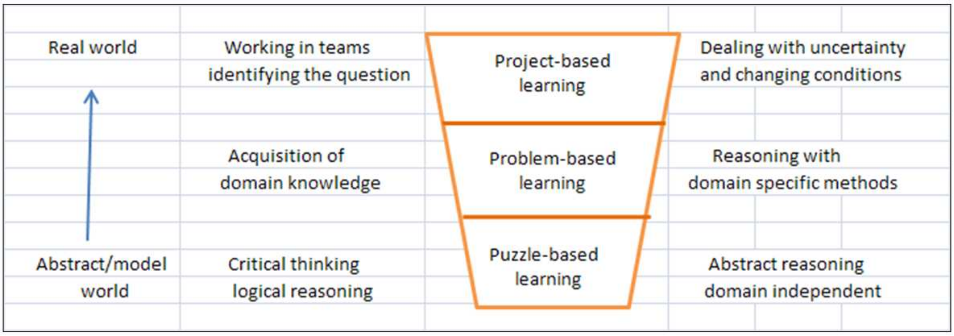

Figure 8: The process of building problem-solving skills. [2]

If we adopt the idea that the ultimate goal of the University education is to lay foundations for students to become effective problem solvers in the real word, then the process of developing problem-solving skills is critical. As described in [2], learning of problem-solving involves three skill categories, illustrated as a sequence in [Fig. 8]. In this continuum each layer builds upon the layer below it. When following such model computing curriculum must put more emphases on building strong problem-solving skills than on teaching inheritance.

7. Puzzle-based Learning

Puzzle-based learning has a very long tradition (Alcuin's puzzle, the popular "river crossing problem" (below), originates from the 8th century). Today puzzles are used not only for developing logical reasoning but also as a way to appeal to the "gamer generation", the generation that associates computing with three characteristics: small + simple + fun (device, problem complexity and rewards, respectively).

Since many real world problems can be perceived as large scale puzzles, it is believed that puzzle-based learning plays a crucial role not only in promoting mathematics, but also in developing problem solving skills. It also contributes to general domain-independent reasoning and critical thinking that can lay a foundation for future studies. Universal mathematical problem-solving principles, like generality, simplicity, learning and entertainment, can be successfully demonstrated to students through puzzles [2]. Puzzle-based learning is an experiment in-progress in several universities world-wide.

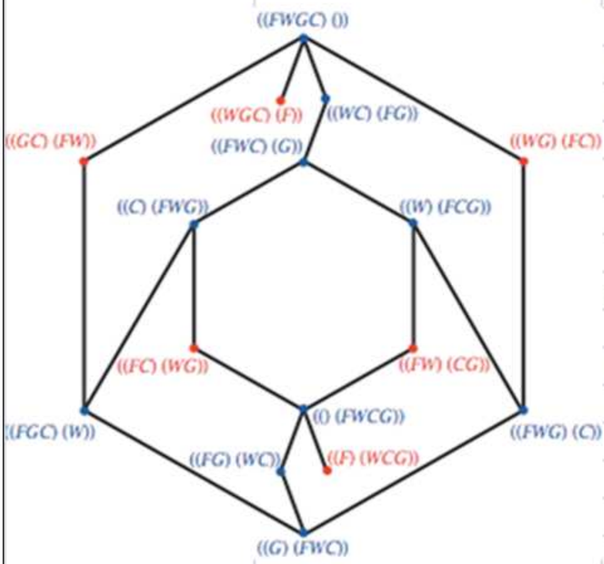

Figure 9: [4] The state transition graph of Alcuin's "the river crossing problem", Source [4]

Figure 10: The Rubik Cube Puzzle

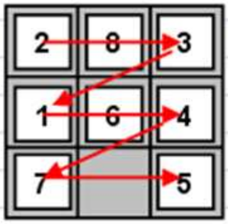

Figure 11: The "8puzzle problem" Source [8]

Well versed in mathematical abstraction, I fully subscribe to the new model of computing education, in particular to the strong emphasis on developing logical thinking and problem solving skills. In my view these skills are absolutely critical for students in order to progress in studies and in life. Development of such skills must start at early age. After all, a good start is a half of success. I only regret that the wakeup call came so late, as my past teaching experience could have been more colourful otherwise. Yet, endless human capacity to retrench, retool and try again is to be admired.

8. Concluding Remarks

- Today's market place needs more e-skilled graduates who are capable of solving challenging problems created by the cyber age
- The lack of problem solving skills by the university students (in-coming as well as out-coming) is alarming
- Governments and educational institutions must promote importance of computing and must financially support educational initiatives that involve computing

- Computing departments must redefine their educational objectives to provide graduates with higher level of competence in computational thinking and cross-disciplinary computing
- Computational thinking has been recognized a necessary prerequisite on which problem solving skills can be built
- Introducing computational thinking to academic programmes will require transformation of high school Computer Science
- Fundamental shift in teaching paradigm is taking place right now in many educational institutions
- There is also a lot going on in efforts to change the image of computing by presenting it as a colourful and enjoyable activity both in traditional and in ubiquitous context
- Contemporary computing requires integration of knowledge from multiple domains more than ever
- Integrating across several disciplines promotes not only computational skills but also insight and innovation
- When redesigning curricula to incorporate cross-disciplinary activities it could be a short curricular distance from "Your students need more breath" to "Your students don't know much" [6]
- "The different shades of grey" are not so different after all. And this is a great way to end.

References

[1] ACM/IEEE Computing Curricula 2008, ACM Press, New York, December 2008

[2] Falkner, N. at al: Puzzle-Based Learning, IEEE Computer Magazine, April 2010

[3] Garcia, D. (Moderator) Panel Session, SIGCSE'09, Chattanooga, USA

[4] Gibson, D. at al. Repairing the Student Pipeline to Computer Science, Global Challenge 2010

[5] Qin, H., Teaching Computational Thinking through Bioinformatics, SIGCSE'09, Chattanooga

[6] Vallino, J. Cutting Across the Disciplines, IEEE Computer Magazine, vol. 43, no. 4, April 2010

[7] `http://www.sciencedirect.com/science?_ob=ArticleURL`

[8] `http://www.8puzzle.com/8_puzzle_algorithm.html`

[9] `http://www.visionlearning.com/library/module_viewer.php`

[10] `http://www.combinatorica.com`

Malgorzata S. Zuberek
Department of Computer Science
Memorial University
St. John's, NL, Canada A1B 3X5