Czech Technical University in Prague

ETH Zürich

Informatics Europe

ECSS 2010

6^{th} Annual
European Computer Science Summit

Proceedings

Prague, 12^{th} – 13^{th} October 2010

Message from the Chairs

Dear colleagues,

Welcome to the 6th annual European Computer Science Summit 2010!

We would like to take this opportunity to welcome you to the beautiful city of *Prague, Czech Republic*. We appreciate that you could take a few days off your busy schedule to meet together and discuss the future of our discipline and the challenges that lie ahead.

This year's Summit takes place in an amazing city, where ubiquitous reminders of ancient history blend with the modern world and glimpses of the future. One can notice an analogy with our field, Computer Science and Informatics. Our disciplines are, too, based on solid and time-proven principles, yet their nature is dynamic and ever-changing.

It is also the first time that the Summit takes place outside of Western Europe. Let us accept this as yet another confirmation that the division of Europe to a „western“ and „eastern“ part is no longer political, just geographical.

Finally, we would like to express our thanks to the Keynote Speakers who have accepted our invitation to share some of their thoughts and insigts in this forum. We would also like to thank all authors who have submitted their papers or abstracts. We would like to thank our sponsors, whose generous contributions and support have greatly enhanced the financial aspects of this event. Finally, we would like to thank to everyone who have aided to the organization of this event.

We wish you a pleasant stay and a memorable experience, both professionally and socially.

Pavel Tvrdík, Bertrand Meyer

Chairs and Committees

Co-chairs

Pavel Tvrdík, CTU Prague, Czech Republic

Bertrand Meyer, ETH Zürich, Switzerland

Program Committee

Frédéric Benhamou, Université de Nantes, France

Christine Choppy, Université Paris 13, France

Gregor Engels, Universität Paderborn, Germany

Victor Gergel, University of Nizhny-Novgorod, Russia

Hans-Ulrich Heiss, TU Berlin, Germany

Bertrand Meyer, ETH Zürich, Switzerland

Enrico Nardelli, Università di Roma "Tor Vergata" and GRIN

Roumen Nikolov, University of Sofia, Bulgaria

Antoine Petit, INRIA, Paris - Rocquencourt, France

Jørgen Staunstrup, IT U. of Copenhagen, Denmark

Letizia Tanca, Politecnico di Milano, Italy

Jan van Leeuwen, Universiteit Utrecht, the Netherlands

Pre-Summit Workshop Chairs

Department Chair Workshop: **Jørgen Staunstrup**, IT U. of Copenhagen, Denmark

EQANIE Workshop: **Hans-Ulrich Heiss**, TU Berlin, Germany

Organizational Committee

Josef Hlaváč, CTU Prague

Jindra Vojíková, CTU Prague

Cristina Pereira, Informatics Europe

Contents

Keynote Abstracts

Franco Accordino, European Comission, Belgium

“Digital Science and Its Impact on Scientific Society”

Today’s scientific landscape is characterized by two inter-related trends. On the one hand, the availability of advanced computing and data infrastructures enables, more than ever, to collect and process data throughout the scientific discovery process, enabling researchers to build more and more accurate "digital" models and to perform detailed simulations, for example, of the innermost properties of nature. On the other hand, the widespread use of “participative” web paradigms in the scientific process enables researchers to share data, models, software tools, papers and (re)views. It stimulates creativity and opens up new perspectives for global multi-disciplinary networking and collaborations. These trends, which we can call the “Digital Science”, are driving a cultural change in the way scientific knowledge is produced, disseminated and ultimately transformed into value. The change affects research, higher education, innovation, but also societal issues such as trust, reputation, accreditation and ethics. ICT is at the hearth of these transformations, not only as a discipline genuinely depending on scientific advances, but also as a fundamental technology which is enabling a new era of discoveries and mindset changing in all sciences. The presentation will introduce the policy context set by the “Digital Agenda”, with particular emphasis on ICT research, education and innovation aspects, as well as the related activities currently developed by the European Commission in the field of science-driven ICT research (e.g. FET young researchers), e-Infrastructures (e.g. open access) and ICT scientific societies.

Carlo Ghezzi, Politecnico di Milano, Italy

“European Funding for Long-Term Research: Facts and a Personal Perspective”

The talk focuses on the current efforts undertaken at the EU level to fund long term research through the European Research Council initiative. After a brief presentation of why the ERC exists, what it does, and what has been achieved so far, I will provide some personal insights gained both as an ERC grantee and as an evaluator.

Sabine Kruspig, European Patent Office, Germany

“Computer-Implemented Inventions at the European Patent Office”

The protection of software by intellectual property law has been a contentious issue for years. The European Patent Office provides a uniform patent application procedure for patent protection in up to 40 European countries. Enforcement of patents is regulated under national law. An overview will be given on patent protection under the European Patent Convention (EPC) and Patent Cooperation Treaty (PCT) for inventions implemented by means of computer programs (CII- computer implemented invention). Nowadays, the quickly developing field of computer technology influences a plethora of technological fields. The borderline for patentable inventions will be explained by way of a few examples from system software and application software.

Leslie Lamport, Microsoft Research, USA

“What is Computation?”

We depend on computer systems that are not dependable. No one knows how to solve this problem, and I don’t pretend to. But a better understanding of what those systems do just might help a little. The tool that people have been using for centuries to understand such things is math. I will use very simple math to explain what computation is.

Jan Sedivy, former IBM Czech Republic & Google, Czech Republic

“Voice: The New UI for Mobile Devices”

First, the speech recognition history will be reviewed very shortly. Then the basic techniques will be explained highlighting the implementation problems on mobile and embedded devices. The central part of the talk will present the recent advances in speech recognition deployment along with the most successful applications. At the end some of the future challenges in speech recognition as the major UI channel will be discussed.

Alfred Spector, Google, USA

“Google Research: Approach, Recent Results, and Challenges”

Google organizes its research and engineering teams in a hybrid fashion – one where we tend to blend research and development. We feel this non-traditional approach to research provides many benefits, particularly in areas of computer science that relate to scale in usage, processing, and data. It seems to be helping us to make good progress on numerous hard problems in systems design, voice recognition, translation of human languages, and more. Nonetheless, every advance still seems to uncover yet more research challenges. In this talk, I will illustrate some of our most recent advances and then discuss the next layer of opportunities that they have uncovered.

Andrey Terekhov, Saint-Petersburg State University, Russia

“Characteristics of Russian Outsourcing”

In the report initial conditions of Russian outsourcing at the beginning of 90th are described. Outsourcing was only one of few opportunities to survive. Russia had strong R& D background in Academy of Sciences and main universities. At the same time Western companies had huge lack of IT-specialists so it was strong reason for outsourcing, difficult scientific intensive tasks started to be transferred to Russia. There are some examples of such first activities. Further establishing of Russian IT-market, industry and IT-association RUSSOFT are described. Important role of RUSSOFT association is shown in examples. The way of Russian IT-business from mainly service-outsourcing to future world leader (including development of own software products) is illustrated.

Moshe Vardi, Rice University, USA

“The Tragedy of the Computing-Research Commons”

The tragedy of the commons refers to a situation in which multiple individuals, acting independently, and solely and rationally consulting their own self-interest, will ultimately deplete a shared limited resource even when it is clear that it is not in anyone’s long-term interest for this to happen. The computing-research community consists of multiple individuals, acting independently and solely and rationally consulting their own self-interest. I will argue that this results in outcomes that are not in the long interest of the community. As examples, I will discuss topics such as publication culture, open access, hypercriticality, theory vs. practice, and debate culture. I will argue that our community needs to take a more active role in regulating itself.

Characteristics of Russian outsourcing

Valentin Makarov, President of RUSSOFT Association

Andrey Terekhov, Head of Software Engineering Chair of Saint-Petersburg State University, Professor, CEO of Lanit-Tercom

1. Initial history

At the end of 1980-s the so-called "Perestroyka" began in Russia. Now many people argue whether it was possible to perform it softer, but «history does not know subjunctive mode». Transition from socialism to capitalism has passed very quickly with huge asperities for the population. There were real threats of starvation and of a civil war. Inflation exceeded 1000 % a year (for example, in 1992 inflation has constituted 2600 %). Gaydar's reforms have filled empty shelves of shops by imported goods, but at such prices which were inaccessible to the majority of the population. In these conditions the availability of convertible currency which were not subject to the Russian inflation provided certain stability. The family of three persons could live well enough on the income of $200 a month in the beginning of 1990-s.

In these conditions each Russian was searching for means of survival. Some people became "chelnoks" (suitcase traders), some traded food, others opened small businesses (repair of clothes or of cars, hairdressing salons etc.). At the same time in Russia there was a very large layer of engineers, teachers and scientists (about 300 institutes of the Academy of Sciences, more than 1000 universities, thousands of applied-research institutes aimed at implementation of scientific results in production). Certainly, not all of them were equally strong but at least one third of institutes of Academy of Sciences and universities worked at the level of World standards. Thus, there were several thousands of high level experts as a heritage of the USSR. Despite difficulties many high schools saved their professors and continued to graduate qualified engineers. Not all of them have been required by the Russian economy at this time. As it is known in the USSR more than half of researches (and some sources say that even more than 75 %) have been connected with defense. New authorities of young Russia have almost completely ceased financing of military contracts. Brain drain became the main problem. Tens of thousands of best specialists of productive ages have left Russia for other countries (basically for the USA). However not all. Many physicists, chemists, engineers and technicians have been retrained into IT-professionals and were successfully working. At that time Americans spent a lump of money for conversion training of Russian nuclear physicists and rocket specialists to prevent them from leaving to North Korea, Iran or still somewhere. They became excellent programmers.

Fortunately, a reverse situation established in the West (mainly in the USA). There were a lot of job positions in IT without sufficient amount of engineers. Green card program expanded in the USA, but American authorities simply did not want and could not increase number of immigrants immensely. Thus, it was quite natural to transmit a part of works to other countries where the labor force was qualified enough, but the cost was lower. Basically the idea of outsourcing has arisen in the USA in 30-s' of the last century. For example, for a company with 10-15 people it is too expensive to keep a qualified accountant in the staff, it is much cheaper to transmit the accounts' department to some professional organization which could well serve ten such small organizations with benefit for itself.

Presently it is usual to outsource all works that are not connected directly to the core activity of the organization (telecommunication, security, cleaning, IT-system administration etc.). The first successful examples of outsourcing to Russia were just of the same field – development of some tools, carrying out of the calculations which were not critical to the client, preprocessing of the data etc. Outsourcing was used typically as a means for minimizing expenses. However many Western companies quickly realized that there was a large quantity of almost jobless highly skilled experts in Russia in many different areas. Therefore some companies began to risk and to outsource their core processes to Russia, especially when they had failed to do it themselves before. Not to be unfounded, we will give examples from Lanit-Tercom's history.

In 1992 Italians (ITALTEL, Milan) have ordered to us the development of SETTOP - device which provides Internet access using the home TV instead of the display. Lanit-Tercom has successfully solved this task. A.Terekhov was invited to Milan for signing the next contract, and Italians showed him with a pride the huge box filled by our boards. It appeared that their purpose was not to get SETTOP but the ATM-switch. They were afraid of sharing commercial secrets with us, therefore forced us working "blindly". A.Terekhov explained they have done wildly inefficient job. For example, on each board there was a microprocessor M68000 with a few additional chips, which was absolutely unnecessary for switches. They agreed, apologized and then ordered development of ATM-switch from the scratch . We completed this task. It seems like it was the first ATM-switch in Europe, it is in use till now in PEAN network. We have developed and realized both hardware and the switch software, but what is even more important, we also developed software for management of heterogeneous communication networks. As far as we know, nobody except us has ever developed ATM tools in full in Russia.

One more example. At the end of 1992 A.N.Terekhov got a letter from US through one former Soviet citizen from Leningrad (and at that time already American citizen). The letter was from the American company Seer Technologies with the invitation to take part in the project on reengineering of legacy systems. Actually, this letter has not been addressed to any definite company as they had already made some unsuccessful attempts in the USA and have been almost despaired. The American business idea looked very attractive: there are tons (it is their expression) of programs written on Cobol, PL1, Adabas Natural and other out-of-date languages in the World. These programs have been successfully maintained for more than 10-20 years in crucial areas (defense, finance, public health services, etc.), but their support was becoming more and more expensive from year to year. With the time authors of programs did not work in the given firm any more, Universities did not prepare experts in old languages, nobody supported technologies used when this software was developed. The worst thing was that the documentation did not reflect a real state of affairs, because of numerous changes made in software for many years of operation. Thus, a unique source of the real information on the program was the source code (the program text). And as Americans always were famous for the adherence to "a brute force" method it was not rare to have an application with 4-5 million lines, consisting of modules with 40-60 thousand lines on 10-15 different languages.

On the other hand, even the understanding of the program operation logic and especially its convertion into the modern platforms demanded deep and detailed analysis (data flow graph, control flow graph , sequence of windows of interaction with the user graph, databases etc.). As it is known the number of paths in the graph grows exponentially according to the number of nodes in this graph, i.e. the analysis written «straight forward» will need to work very long.

Our long-term experience of compiler development and the strong mathematical background of our employees helped to solve this task. Reengineering gave us mass of absolutely new tasks and solutions. Some tens of master thesises and three PhD thesises were proved on this topic. The final product – RescueWare was admitted by Gartner Group as the best in the field of Legacy Understanding and Legacy Transformation in 2000 and 2001.

We think that almost every successful Russian enterprise has set of such stories of entrance to the outsourcing world.

2. Start of the Russian IT market and of the IT industry

In the USSR the software market (as well as the markets of other industrial products) simply did not exist. Software was developed under the state contracts, accumulated in the State bodies and centrally distributed. There was no software market, so there was no advanced industry of software development. At the very beginning of “Perestroyka” there existed so-called “cooperatives” which basically were selling personal computers. It was not meaningful to sell software as it was impudently copied, the piracy was prospering (however as well as in other kinds of activity). We know from the pioneers of Russian's software products' development how difficult it was to get money from first corporate customers. But piracy is not a feature of Russia alone. When in 1989 A.Terekhov has arrived to Bombay to sell a program called JEC of a dialog task input and text editing for ES EVM (clone IBM/360), developed at Leningrad university, he has seen with surprise this program already installed practically on all computers.

In programming itself we were not worse than our Western colleagues, besides in the mathematics and knowledge of effective algorithms – even much stronger. But we were amateurs, not professionals in the organization of development process, in such fields as version control, configuration management, quality assurance, project management. We were gaining all this knowledge at completing works for our Western partners. Budgeting seemed especially difficult – we simply could not understand how it was possible to estimate precise cost of the product development. In Soviet period we had the fixed budget for many years forward, even the most good experts could not receive more than 1,5 standard monthly salaries, for the shifted completion dates we could be scolded, but not fired. Contracts with Americans were the real school for us and for our teams.

The total turnover of all Russian software companies working for export hardly exceeded 100 million dollars per year by 2000. Russian economy was basically recovering by this time, many branches of industry felt quite successful, inflation was limited enough though still exceeded 10 % a year. Success in putting the country in order basically was associated with Putin's name. Well, he is not a true democrat, many liberals criticize him for some decisions, especially often discussing Khodorkovsky's case, but we can’t but say that in our opinion, essential economic progress in Russia began after 2000. In new conditions of relative stability and economic growth in Russia we faced a challenge of forming our software industry’s image, increasing its visibility from outside the country, finding possible ways of lobbying laws, important for our industry. Usually the role of the industry voice for the society and for the authorities is being played by a professional association because it’s quite difficult for one even very large enterprise to represent the industry as a whole.

3. RUSSOFT Association

As it often happens, our association was born almost occasionally. In 1999 the city administration of Saint-Petersburg organized a delegation to San Francisco for visiting Silicon Valley. The delegation of Saint-Petersburg included 4 representatives of IT Organizations and the deputy-chairman of the International Relations committee of the Saint-Petersburg's Government – Mr.Valentin Makarov. The General Consulate of Russia in San Francisco and Russian trade mission have helped with organization of this visit.

We have visited head-quarters of Oracle, Sun Microsystems and Hewlett Packard and met representatives of these organizations at level of vice-presidents. Naturally we praised our native qualifications and asked for serious orders. It was before the beginning of 2000th - the period on the threshold of Internet-Bubble. The Indian companies have grown to very vast scales by this time, therefore vice-presidents of all three mentioned corporations told us that they were interested in teams of 500 persons and more and they could not accept working with such small organizations as ours.

Sitting in our hotel round a pool in the evening, we sadly discussed visit results. Then Andrey Terekhov and Valentin Makarov have suggested creating a consortium of the several Saint-Petersburg companies to have possibility to expose "500 bayonets". We did not think long about the name of a consortium. Just in the eve of meetings with the US IT-big bosses we had visited the ancient Russian fortress called Fort-Ross in California and spent there a playful christening, having bathed in ice waters of Pacific Ocean. This fortress has been constructed by Russians about 200 years ago for providing Alaska (it was Russian then) with foodstuff. Later on, answering the questions of journalists why we had chosen such American name for Russian association, we said that we had exactly the same problem as 200 years ago – to create a front-point a for providing Russians with orders on software development.

We have held series of meetings with managers of St.-Petersburg enterprises after returning from America, our idea has been supported completely. Valentin Makarov has left the City government and became the President of our consortium. After that in different regions of Russia similar associations of software developers began to be formed, we were sending them samples of necessary documents gave advices about the work organization, answered numerous questions. Historically St.-Petersburg organization became the leader among Russian associations. The association of Moscow developers was organized essentially later. According to the Moscow traditions they immediately took name RUSSOFT, claiming the All-Russia scale of the association. Two years passed and the Moscow association did not reach our scale. That's why we merged at 2004 and fixed RUSSOFT as the name of the All-Russia association, but with the headquarters in St.-Petersburg and with Valentin Makarov as a president of united Association. Andrey Terekhov was the first Chairman of RUSSOFT Board and bears a proud name of the founding father of this organization.

Now there are 70 companies in RUSSOFT, not only from Russia, but also from Ukraine and Belarus. Sometimes hot discussions take places between the governments of these three countries, but we do not remember any scandal, but even small disagreements between businesses. Both in Ukraine, and in Belarus there are associations similar to RUSSOFT and we support close friendly relations with them, organize joint conferences, road-shows and other activities. Annually RUSSOFT conducts survey on the software development export industry in Russia which covers such questions as estimation of export values (depending on different models of business and industry sectors), geography of clients, vertical markets, situation with staff, software companies' satisfaction of the government policies, quality of regional infrastructure etc. We use data of well-known consulting companies like IDC, Gartner, Forrester Research, NeoIT and others for marketing surveys.

Nothing was saved from the initial idea – collecting contracts under the umbrella of an Association. It is difficult to convince the Customer that splitting of its precious problem into almost independent pieces is possible, difficult to adjust cooperation, it is almost impossible to distribute responsibility and profit fairly among members of an Association. With years coming the tasks of association were essentially changing. The «club» atmosphere comes upon foreground. Directors of the IT enterprises gather regularly and discuss problems and ways of their solution in informal conditions. The important part of Association activity is lobbying of the IT-industry interests. 20-30 years ago “lobbying” was the abusive word in our country applied in newspapers for accusation of defects of capitalist society. Now lobbying is perceived absolutely normally, we often deliver speeches in State Duma, we keep in touch with many officers from the Government, we push both to make business conditions better for the industry.

There are first successes. In 2007 the unified social tax (which is the biggest tax paid by the employer in our industry as the salary forms over 60 % of all expenses) was diminished from 26% down to 14% for export oriented software companies. For businesses which deal with natural resources the salary makes 3-5 % from general expenses, so 26% from the salary is absolutely insignificant loss for them. But it is dramatically significant for enterprises developing software for export. Taking into consideration that our basic competitors in India, China, Brazil, Ireland and other countries have no such tax at all or it’s rate is very low, it is obvious that the Russian enterprises are losing the international competitiveness “a priori” because of this tax alone. We managed to achieve decrease in this tax for our enterprises for two years (2008-2009). We have also been struggling for simplification in customs' regulation, of rules of VAT collection, on export promotion measures, etc.

Nevertheless one part of the initial task was kept intact. We mean creation of a positive image of Russia in other countries as the important constituent of the marketing strategy. 15-20 years ago it was very easy to hear a question from the potential customer, whether bears walk along Nevsky Avenue (the main avenue of Saint-Petersburg). If you open any Western newspaper the word "Russia" can be met only in a context of «Russian mafia», «crime», etc. When the entire World was trying to solve the Y2K Problem, Associated Press has published the big interview with Professor Andrey Terekhov, but has supplied his interview with a photo of the tired peasant in a padded jacket, sitting at desktop with bottle of vodka. There were a lot of such cases. RUSSOFT has spent many years of regular operations promoting true state of affairs and success stories because we tried to break this situation. In our opinion, today the Russian image is essentially better, than it was 10-15 years ago. Though relapses of “cold war” have an effect from time to time.

Our favorite success story is the ACM-ICPC International Programming Contest sponsored by IBM. In the first 20 years of existence of this competition American universities had won many times. But since 1996 when Russia began to take part in them, it became difficult for Americans to enter even into the first ten. Six times Russian universities won the absolute first place, practically every year they occupy 4-6 prize-winning places from the first ten. And these are universities not only from Saint-Petersburg or Moscow, but also from mid-sized Russian remote cities. We consider this fact shows level of Russian education in the field of IT in general.

4. Samples of Russian outsource success stories

To begin with the report on the “50 Global Emerging Outsourcing Cities' study” produced by Global Services and Tolons in 2009. St.-Petersburg has won the first place among attractive places for establishing R&D. It also entered into the Top 5 on engineering services and in Top 10 on products' development. Moscow is number 3 on R&D and also is included into the Top 5 on games development services.

Besides, it is necessary to consider that 6-8 largest companies from Russia, Ukraine and Belarus are always presented in Top 100 of different ratings in different nominations (fig. 1).

Almost all world IT leaders organized development centers in Russia proving that Russia is one of world's leaders in the field of research and development:

1. Intel – over 1000 employees in Nizhny Novgorod, Sarov, Moscow, Saint-Petersburg and Novosibirsk
2. Sun – over 300 employees in St.-Petersburg
3. EMC – over 300 employees in St.-Petersburg
4. IBM – large development center in Moscow
5. Motorola – over 300 employees in St.-Petersburg
6. T-Systems – over 300 employees in St.-Petersburg

HP and Google opened their development centers in Saint-Petersburg in last 1-2 years despite crisis and programmers' salaries growth.

fig.1

Our companies took strong positions in the vertical markets.

1. Finances. Luxoft and Exigen Services successfully work with the Deutsche Bank, DataArt – with smaller financial institutions.

2. Oil extracting, processing and trade in oil products. EPAM Systems carries out large orders for Halliburton.

3. Industrial production and design. More than 700 Luxoft employees are working under direct orders from Boeing

4. Lanit-Tercom works for many years in the field of creation of the medical equipment (for Laerdal), in development of software for devices for foodstuff quality check (FOSS), in reengineering of Legacy Software (Relativity Technologies).

It is possible to describe hundreds of such examples, we have mentioned only the most known (or most familiar to us).

Russian IT offshoring market has grown from $1.5 billion to $2.5 billion before the crisis in 2006-2008. Especially it is worth to underline that R&D's part of export has reached $1 billion of the total $2.5 billion in 2008. Many Russian companies which met problems in competing with Indian companies on the cost basis in traditional services have chosen a niche of high-end software and technology development. It allows them to effectively use advantages of the educational scheme and of R&D experience of Russia.

5. Russian IT-Industry Today

The first 10-15 years of outsourcing activities were very important for Russia, especially in the field of project management and quality assurance. However today we understand that outsourcing cannot provide explosive growth to our industry, there is no intellectual property accumulation, it is necessary to struggle at least for moral rights on intellectual property when you work on a contractual basis (possibility of the publication on work topics, a mention of authorship of development, etc.)

Many Russian companies, who have stored enough experience in product development for Western customers, started to develop own products, to master the vertical markets. While preparing this article the authors studied success stories of Russian companies but faced an old problem. There are products which are being sold in million copies. We precisely know that they are developed not only in Russia, but namely in St.-Petersburg. But their authors are forbidden to mention Russian roots as are afraid that the disclosure of that information would damage sales. Nevertheless, we will give some examples.

1. Parallels (Sergey Belousov) – one of world leaders in area of resources virtualization.
2. Aelita Software (Ratmir Timashev, Andrey Baronov) – Microsoft networks administration and network security. Has been sold more than for $100 million to Quest.
3. Speech Technology Center (Michael Hitrov) - is engaged in the field of the voice integration. They have successfully decrypted records of the black-box of the submarine "Kursk" which lain at the bottom of the sea for more than a year. They have also the World's largest application in the field of speech recognition.
4. SPb Software – one of the leaders on Windows Mobile applications development, while Novosibirsk company Vito Technologies has leading positions in iPhone applications development for Education.
5. Lanit-Tercom company created technology of designing of the chips optimized for the required problem, and now is promoting its solution in the field of stereovision (two webcams without any additional equipment).

It is important to notice that 5 from 10 leading world service providers on software products development (Global Services&NeoIT, 2009) are Russian companies. In our opinion, this leadership in product development is strongly related to the educational and scientific background of Soviet and

Russian engineers. It seems that owing to the superiority of our education system and of engineering school, our developers and experts in algorithms are able to smoothly adapt to changing conditions and to find solutions for the most difficult technical (and also — administrative, financial and other) problems better than anybody else. When we added understanding of market and of its requirements to this ability, we have quickly reached leading positions – first in application developers and then - in software products developers.

Throughout a number of years the growth of export of software from Russia exceeds growth of export of software development services. There is a maturation of capital venture activity supported by the Government. The State owned Russian Venture Company (RVC) together with Bank for foreign trade (VTB) initiated creation of a network of regional venture funds and also of «seed-money» funds. Large foreign corporations (Microsoft, Cisco, HP, IBM) and foreign venture funds («Almaz Capital","Siguler Guff","Drapper, Fisher&Jurvetson) actively cooperate with these funds. Associations of business angels have been created in Moscow and St Petersburg first, and then in mid-sized cities.

According to the annual survey organized by RUSSOFT in 2010, business has reacted immediately. Already more than 20 % of respondents plan to involve investments in 2010 and as much — in 2011. Market consolidation became an important tendency during the crisis – it was noted by more than 60 % of respondents. These tendencies coupled with development of the system of venture financing instruments should lead to a series of new World leaders of the software industry in Russia very soon. It will be unexpected even for Russians at first. And then it should become norm. Finally we will easily forget that time when potential of Russian developers was used for «driving nails», competing in prices with colleagues from developing countries.

SUPERCOMPUTING EDUCATION: THE THIRD PILLAR OF HPC

Voevodin Vladimir (Moscow State University, Russia)

Gergel Victor (Nizhni Novgorod State University. Russia)

Abstract

The paper raises a question on the necessity of implementing the supercomputing education in the leading research and technology universities. The report shows the experience of Russia in forming the national system of training high-qualified specialists in the sphere of the supercomputing technologies, which might be useful in solving the problems of supercomputing education in other countries. This experience can also be taken into account in organizing the large-scale educational programs in other crucial areas of information technologies.

1. Supercomputing Education – Why?

The supercomputing technologies are referred today to the most important directions of the scientific technical development of many leading countries of the world, including Russia. The potential of the supercomputer industry makes it possible to solve many fundamental and applied scientific and technical problems which require large-scale computations.

Supercomputers are being developed incredibly fast, which impacts the indices of performance and scalability of the modern computing systems. Let us refer to the Top500 List of the most powerful computer systems of the world (www.top500.org), the newest version of the list was released at the conference ISC-2010 at the end of May. Supercomputer Cray Jaguar being the leader of the list has the performance of 1,76 Pflops (1 Pflop = 10^{15} of floating point operations) and has 224162 cores working in parallel. The performance is vast but it will be available to a user only in case all the 224162 cores are used effectively. How to create a parallel algorithm of problem solving that could scale up to such values? How to distribute the data among more than 26000 compute nodes of Cray Jaguar to minimize the data exchange? Is it possible to create an algorithm with no data exchange at all? The questions are not simple and without special training it is difficult to find the answers to these questions.

The second position of Top500 is given to the Chinese computer Nebulae with the performance of 1,27 Pflops. In contrast to Cray Jaguar Nebulae uses the graphic processors Nvidia Tesla C2050, which are the co-processors to the traditional Intel Xeon X5650 processors. How to formulate an algorithm of problem solving so that it would be possible to reflect the SIMD nature of the graphic processors and at the same time to hide the data transmission time between the major processors and co-processors? If it is not done, the user will get only small percentage share of the performance. The Nabulae supercomputer performance shown on the Linpack benchmark – only 43% of the maximum possible performance - is a very persuasive confirmation that it is not an easy question. It should also be noted that these results were shown on the benchmark that is extremely easy, regular and has high computational power. During 30 years of using this benchmark we have learnt everything about it. What results will be shown with the real applications?

We can continue this analysis for all the systems of Top500 and for all the representatives of the supercomputer industry. The number of processors and cores in computing systems is steadily growing, which in combination with the heterogeneity sets non-trivial issues of how to use them. If we extrapolate the tendency of supercomputer development shown in Top500, in 2018 we can expect the emergence of the systems of exaflop performance that will have hundreds of millions cores working in parallel. Realizing the complexity of working with such large-scale systems the computational community is making attempts to understand what software will be able to work on the systems of the future, where the main bottlenecks are and how to overcome them. It is exactly with this purpose that J. Dongarra and P. Beckman initiated the project - International Exascale Software Project (www.exascale.org) - where the expert community discusses and tries to understand which changes in software should be introduced and which software components should become the point of applying the forces of professionals.

Supercomputers are rapidly developing. IESP discusses the software stack of the computers of the future. But there raises the question: who will work on these computers? To be more exact, who will be able to work on them? Who will be able to develop a method, create an algorithm and write an efficient program for these computers? Very few ones. And we should not have any illusions that it is relevant only for the systems of the highest performance range containing hundreds of millions cores. The situation will be similar for the “ordinary” desktop systems of the performance tens of thousands times lower and containing “only” about 104-105 cores. It is necessary to start working purposefully at developing the system of supercomputing education in order to prepare the specialists for the realities of the future superparallel computer world. Only three components together – hardware, software and supercomputing education – will create a steady basis for the development of the entire high performance computing area.

2. Supercomputing Education – When?

In recent years we can observe a paradoxical situation in the supercomputing area. On the one hand we see the stupendous development of the computing systems. Computer capability has been developing in accordance with the Moor's law for quite a while and doubled the performance within 18 months. That is why every 11-12 years the supercomputer performance increases by 3 orders, which leads to the transfer from Mega to Giga, from Giga to Tera, from Tera to Peta and so on and so forth. As a result, today we have the incredible characteristics of computing systems that are stated in the current version of the Top500 List of the most powerful computers of the world.

On the other hand, what is the progress in the development of the parallel programming technologies or in the methods of problem solving? What happened in these areas that would be comparable-scaled to the happenings of the hardware area? Almost nothing. The pace of computer development remains invariably fast while no adequate actions on developing the methods of working on such computers have been taken. Many years of inactivity in this sphere led to the huge gap between the possibilities of the modern supercomputers and the practice of their real usage. As a result, we spend a lot of effort on the struggle with MPI, we try to scale the non-scalable methods, we re-write the programs constantly moving them from one computing platform to another.

And the reason for the current situation is the absence of the corresponding supercomputing education. We should have started to actively implement it 10-15 years ago. This delay should be urgently compensated by active actions.

3. Supercomputing Education – How?

Having realized the significance of the supercomputing education, it is extremely important to solve the issue of what should be taught in the sphere of the supercomputing technologies. The analysis of the state of things shows that unfortunately in many universities this education is narrowed to studying only some technological aspects. But it is absolutely clear that a specialist in the sphere of the supercomputing technologies must have the knowledge and skills in a wide range of computer science issues. Among the required knowledge we find informational structure of programs and algorithms, architecture of parallel computing systems, models of computations and methods of complexity analysis, parallel numerical methods, parallel programming (languages, development environments, libraries) and many others. Moreover, it should be clear that creating parallel software is impossible without professional knowledge and skills in “usual” sequential programming. It is also very important to understand that in most cases the development of efficient supercomputer programs for solving most complex scientific technical problems is impossible without having the highest qualification in mathematics.

How to provide the training in such a wide range of required knowledge? To add to that, it is important to understand that the activity in the sphere of the supercomputing technologies will vary for various specialists. They can be creating new efficient parallel algorithms, or developing parallel programs, or solving complex computation-intensive applied problems, etc. As a result, it becomes clear that we should identify various directions in the supercomputing technologies education.
Among them there could be the following:

1. Designing, developing and using supercomputing systems,
2. Administration of supercomputing systems,
3. System programming,
4. Applied programming,
5. Advanced training for university teachers giving the training in various aspects of parallel programming (system programming, programming in various subject areas – physics, mechanics, etc.),
6. Advanced training for specialists in various applied areas where supercomputing systems can be used for problem solving.

It should also be clear that the training in a certain selected area should significantly depend on the target category of the trainees. Thus, the programs for teaching students should undoubtedly differ from the programs of the advanced training for IT specialists or university teachers.

Forming a certain “complete set” of supercomputing knowledge is still ahead and the significance of this task will apparently attract the attention of many scientists, teachers and specialists. A possible approach may involve creative usage of the development methodology of Computing Curricula 2005 of the international associations ACM and IEEE. Based on this approach the required “complete set” of the supercomputing knowledge can be named as Supercomputing Curricula and its initial version can be as follows.

Mathematical foundations of the supercomputing technologies

1. Mathematical foundations of parallel computing

2. Algorithms and analysis of the parallel computing complexity
3. Computational mathematics and methods of parallel computing

Information technologies and high performance computing

4. Architecture and organization of the high performance computing systems
5. Operating systems and management systems for high performance computing systems
6. Distributed computing and Grid technologies
7. Organization of the human-computer interaction in conducting numerical supercomputing experiments
8. Computer graphics and visualization of the high performance computing results
9. Information management in conducting high performance computing experiments

Methods and technologies of supercomputing

10. The basics of parallel programming
11. Languages and parallel programming technologies
12. Supercomputing software engineering
13. Software systems for supercomputing experiments
14. Social and professional issues of the supercomputing technologies

4. Supercomputing Education – Where?

The magnitude of the supercomputing education problem requires the consolidation of efforts of all the university community alongside with close collaboration with the world scientific centers and the leading IT companies. The creation of the Supercomputing Consortium of Russian Universities (http://www.hpc-russia.ru) in 2008 became one of the most significant happenings in this area.

The purpose of the Supercomputing Consortium is to develop and to ensure holding a set of events and activities aimed at the efficient usage of the higher education system potential for developing and implementing the supercomputing technologies in the Russian education, science and industry.

Within the framework of the purpose in view the Consortium is oriented at solving the following major issues:

- Coordination and organization of cooperation among the higher educational institutions in Russia in terms of using and developing the modern supercomputing technologies in education, science and industry,
- Development of the educational programs for training the highly qualified specialists, advanced training and continuing education in the sphere of the supercomputing technologies and high performance computing,
- Contribution to the development of the network of highly efficient supercomputing centers in Russia oriented at implementing high performance computing technologies in all the spheres of the national economics,
- Formation of the advanced experience exchange system, distribution of the latest achievements in the sphere of high performance computing,
- Development of the scientific educational and technical target programs in the sphere of the supercomputing technologies and high performance computing,
- Contribution (in collaboration with the Russian Academy of Science organizations) to enhancing the fundamental and applied research requiring high performance computing resources in the Russian universities,
- Development and support of the system of the Russian scientific conferences and youth scientific schools on the supercomputing technologies and their applications,
- Enhancement of the international educational and scientific collaboration of the Russian universities in the sphere of the supercomputing technologies and high performance computing.

Another important direction in the activities of the Consortium is the active promotion of the works on practical usage of the supercomputing technologies for solving today's problems of science, technology, industry and business, that could not have been analyzed earlier due to the limitations in the existing computing tools and technologies.

At the beginning of 2010 more than 25 leading universities of Russia were the members of the Consortium. They are followed by the associated members, - Russian Academy of Science institutions and IT companies.

5. Supercomputing education – the national program in Russia

The significance of the development and practical use of the supercomputing technologies sets the problem of the supercomputing education as a task of the state level importance. The National Project of the Supercomputing Education has been created in Russia to solve this problem. It was initiated by the Supercomputing Consortium and supported by the president and the government.

Formation of the integral system of training highly qualified specialists in the sphere of the supercomputing technologies (SCT) must become the major result of the Project. The basis of this scheme will be the scientific educational centers (SEC) in SCT. The main task for these centers will be the efficient organization of the universities' activities aimed at teaching, continuing and advanced training in the sphere of the SCT. The first result of their activity will be the development of the professional competencies (Supercomputing Curricula) in the sphere of the SCT according to the activity types and target groups. Based on this recommendations there will be implemented the modernization (update) of the federal state educational standards of the third generation. The modernization will be aimed both at the in-depth training in the SCT in mastership and at the implementation of the SCT in the basic education (bachelorship) starting from the first years of the university studies. The following educational standards will be updated first and foremost: Mathematics, Mathematics and Computer Science, Fundamental Informatics and Information Technologies, Applied Mathematics and Informatics. It is planned to train no less than 500 highly qualified specialists in the sphere of the SCT and their applications and to provide advanced and continuing training to 150 representatives of the teaching staff on the basis of the updated educational standards.

To realize the new educational programs the advanced and continuing training for teachers in the sphere of the SCT will be organized on the base of the SEC SCT. As a result of the Project realization no less than 25 universities will be involved in the training of specialists in the SCT.

Another important activity for the SEC SCT will be forming special groups of the trainees for various categories of participants (students, masters, postgraduate students, teachers, specialists) for intensive target-oriented training in the sphere of the SCT. Such kind of training takes into account the cross-disciplinary nature of the SCT and will make it possible to train highly qualified specialists that will be able to develop and to efficiently use the SCT in conducting fundamental and applied research, to implement them in industry and economics.

The national system of SCT education quality monitoring will be established on the base of the SEC SCT. Under the SEC SCT control there will be developed new educational courses and programs in the SCT and their applications. In the course of the project execution there will be prepared a series of textbooks and manuals in the SCT.

Another way to teach the basics of the SCT will be the Internet University of the Supercomputing Technologies. On the base of this university the trainees will be able to study distantly.

The SEC SCT will have another important function – to maintain close relationship among the higher educational institutions, the Russian Academy of Science, industry and business. Within the realization of the Supercomputing Education Project they will cooperate to train specialists and to conduct scientific research in the sphere of the SCT.

The Project realization will be aimed at the in-depth integration of the Russian universities into the world education process in the sphere of the SCT. In the framework of this direction it will be possible to cooperate with the foreign scientific educational partner organizations, to develop cooperative educational programs, to attract foreign scientists to training Russian highly qualified specialists in the sphere of the SCT.

A series of popular scientific articles, printed and electronic publications, other presentations on the project results will help to inform the society about the achievements in the development of the SCT and to attract talented young people to it. There will be arranged a periodical edition "Supercomputing Education in the World" that is an approved and effective way to promote the SCT. The presence at various significant events will also be arranged to inform the society and to present the achievements in the sphere of the SCT.

The major objectives of the Supercomputing Education Project are:

Objective 1. Creating the network of scientific educational centers of the supercomputing technologies (SEC SCT). The objective includes creating the infrastructure, developing the regulatory framework and methodological support, purchasing hardware and software, maintaining and staffing the SEC. The objective realization is aimed at the concentration of the scientific educational potential on the base of the universities having the resources and significant experience in the development of the supercomputing technologies (SCT).

Objective 2. Developing the methodical provision of the system of teaching, advanced and continuing training for the personnel in the sphere of the supercomputing technologies. The following activities will be realized in the framework of this objective: developing and/or updating the educational standards of the new generation, preparing the new set of knowledge and skills containing the requirements to the professional competencies of the specialists in the sphere of the SCT, developing the curriculum and educational programs, producing textbooks, manuals, problem books and other literature on the SCT, preparing and publishing the consolidated list of the main scientific conferences, seminars, contests, schools on the SCT, developing the strategy of the supercomputing education quality monitoring. This objective is aimed at the creation of the scientific methodic basis built upon the educational standards of the new generation for training specialists in the sphere of the SCT.

Objective 3. Realizing the educational programs for teaching, advanced and continuing training for the personnel in the sphere of the supercomputing technologies. The objective includes the following:

- Training 500 specialists in the sphere of the SCT;
- Advanced and continuing training for the teaching staff of the universities;
- Advanced and continuing training for specialists having professional higher education with the use of the distance learning technologies;
- Creating Internet centers of the system of educational resources in the sphere of the SCT.

This objective is aimed at creating the scientific educational potential that ensures the high quality of the training provided to specialists in the sphere of the SCT.

Objective 4. Developing the integration of fundamental and applied research and education in the sphere of the supercomputing technologies on basis of the cooperation with the Russian Academy of Science, industry and business. The objective is aimed at the development of the integration of fundamental and applied research and education in the sphere of the supercomputing technologies, the cooperation with the Russian Academy of Science, industry and business. Reaching this objective presupposes establishing the partner relationships among the SEC SCT, the Russian Academy of Science, industry and business. There will be published a series of reviews, articles, analytical reviews, works in the sphere of the SCT and their application in science and business. In the framework of this objective there will be held meetings with invited representatives of industry, business and science that will make it possible to integrate the leading hi-tech areas of economy and to staff these areas with the specialists in the sphere of the SCT.

Objective 5. Enhancing the international collaboration in creating the system of the supercomputing education. The objective includes attracting the leading foreign specialists in the sphere of the SCT to participate in the training of specialists. The objective presupposes establishing the partner relationships with foreign scientific educational partner organizations, arranging traineeship for Russian teachers, participation of foreign professors in the development and realization of collaborative educational programs. There will be periodic release of the edition "Supercomputer education in the World". The objective is aimed at increasing the academic mobility and enhancing the quality of the education in the sphere of the SCT.

Objective 6. Developing and realizing the system of information support of the society on the achievements in the sphere of the SCT, which presupposes publishing popular science articles, preparing and broadcasting TV programs, publishing materials in electronic media and other presentations of the results of the SCT education system development. The objective is aimed at the popularization of the scientific knowledge in the sphere of the supercomputing technologies, at attracting talented young people to science and to the development of the economics strategic direction.

The Supercomputing Education Project realization is planned for the period of 2010-2012. The following events can be considered as the first prominent results of the Project realization:

All-Russian Youth School. The Supercomputing Consortium of Russian Universities conducted a unique All-Russia Youth School "Supercomputing Technologies and High Performance Computing in Education, Science and Industry" (http://www.software.unn.ru/ccam/?dir=278).

All-Russia Youth School made it possible to move to a new quality level of conducting scientific educational schools. The peculiarities of this school are as follows:

- Active support of the school from the university community: the educational program was realized by the leading scientists and well-known specialists of the country;
- Comprehensive coverage of the supercomputing technologies and high performance computing issues: there were 7 parallel sections covering a wide range of topics;
- The variety of forms and methods of teaching: the school program included intensive classes with an enlarged laboratory courses, review lectures from the leading scientists and specialists, trainings of various complexity level;
- A significant number of the participants: 75 participants from 25 cities of Russia, young specialists, postgraduate students, students of various qualification levels underwent the school training.

The participation in the school provided a unique opportunity of the additional professional training in the sphere of the supercomputing technologies. The feedback shows that this school was well-organized and excellently conducted; it proved to be exceptionally useful. It was decided to include this school in the plan of annual events of the Supercomputing Education Project.

Target educational programs. The training for the first special group consisting of 40 senior students of Computational Mathematics and Cybernetics, Mechanics and Mathematics and Physics Faculties of Moscow State University was conducted according to the specially developed educational program “Supercomputing Technologies”. Leading specialists of MSU and institutions of Russian Academy of Sciences were engaged for lectures and seminars. A computational laboratory course for these special groups was conducted on the basis of the high performance systems IBM BlueGene/P and SKIF MSU “Chebyshev” of the MSU supercomputing center.

The Supercomputing Consortium Meeting. The annual meeting of the Supercomputing Consortium of Russian Universities was held in Moscow University and covered the topic “The Realization of the State Decisions on the Training in the Sphere of the Supercomputing Technologies and High Performance Computing”. Apart from the rectors of the Universities among the participants there were the leading scientists of the Russian Academy of Science and the representatives of a number of the industrial companies. The meeting covered the tasks of developing the training and the issues of using the supercomputing technologies in the sphere of engine-construction, ship-construction, oil and gas industry, biotechnology and pharmacology.

The Internet University of the Supercomputing Technologies (http://www.hpcu.ru) is oriented at teaching a wide range of trainees (students, specialists, teachers) and presupposes the availability of various areas of the training to meet various professional requirements in the sphere of the supercomputing technologies (users, programmers, engineers). The project was supported and became an integral part of the Supercomputing Education Project.

The basics of the learning process organization is as follows:

- The teaching process in the University is formed on the basis of the distance education technologies (presenting the learning materials in the Internet, modular presentation of the learning material, automated testing);
- In presentation of the learning materials the traditional hypertext format is accompanied by the widely used video materials;
- Holding classes in the form of video conferences is an important part of the training; to a great extent it makes it possible to realize the opportunities of the classical full-time education

(presenting the learning material by the teacher, testing the trainees, organizing the work under the teacher's supervision) on a new technological basis;

- Independent work of the University trainees and the possibility to conduct the computational laboratory work in the course of studies is ensured by providing the access to the supercomputing centers of the universities participating in the project;
- In the course of the project development there might be an opportunity to regularly hold the full-time seminars-schools (for instance, in the framework of the scientific technical supercomputing conferences).

During two academic years 2009-2010 more than 200 trainees had the training in the University, which can be considered as the first results of the Internet University of the Supercomputing Technologies. The results of the training show that the chosen direction has great perspectives. The ultimate success will depend a lot on several factors: how the project will be perceived by the supercomputing community of the country, if it will be possible to involve the best teachers and specialists into giving classes and how interested the Russian universities will be in this form of education.

Conclusion

The selected number of the key issues of the national system of supercomputing education shows the magnitude of the efforts taken in this direction in Russia. This experience can be useful in solving the problems of the supercomputing education in other countries and can also be taken into account in organizing the large-scale educational programs in other crucial areas of information technologies.

It should be noted that the supercomputing education system which is being formed in Russia now is oriented at the deep integration of the Russian universities into the world educational process in the sphere of the SCT. With this objective in view the widest possible collaboration with foreign scientific educational partner organizations is planned. This collaboration can also contain the development of the joint educational programs, the participation of foreign scientists in training of highly qualified Russian specialists in the sphere of the SCT, the student and teacher exchange etc. Attracting the attention of all the interested parties to such kind of collaboration is one of the major objectives of this report.

New Master Specialization in ”Knowledge Engineering”

Pavel Kordík

Dept. of Computer Science, Faculty of Information Technology, Czech Technical University in Prague

kordikp@fit.cvut.cz

Abstract. *Business intelligence is helping companies to survive, gain advantage, increase profits. Many BI positions are available even in the crisis, but there is lack of people with required expertise on the market. Recently, the Faculty of Information Technologies of the Czech Technical University in Prague opened new specialization on Knowledge Engineering to fill this gap. This contribution explains the concept of our specialization, list prerequisites and core modules with a detailed description of their objectives and topics.*

Keywords

Knowledge engineering, master specialization, data mining, business intelligence.

1 Introduction

Due to the strong demand from industry, we are opening new master specialization on Knowledge Engineering. A successful absolvent of our specialization will predominate with strong programming background and deep theoretical insight when compared to graduates enroling related specializations abroad.

Members of our team have a long record in teaching Knowledge engineering and Business intelligence related modules. We pursue research and develop open source applications in these fields. Excellent students contribute to our research projects and are co-financed from our research grants.

We plan to use commercial business intelligence software in application oriented modules and open source software in theory oriented modules. In application oriented modules, part of lectures will be given by top industrial experts, who will formulate topics and supervise part of student theses. Our students can be both theoretically and practically oriented and continue in our doctoral programme with the same specialization or become industrial experts in the field.

2 Prerequisites

In this section, we summarize the knowledge each applicant should have in order start studying in our master specialization.

Enroling students should be familiar with the *basic statistic* which is obligatory for majority of bachelor programmes worldwide. We prefer students with excellent *algoritmization* skills and programming background in Java or C++. Basic knowledge of *databases* is also important.

Our faculty offers a bachelor programme that train students in all above mentioned skills.

2.1 Bachelor programme modules

In the bachelor programme, we offer several modules that form solid ground for our master specialization. Beside the essential modules *Database systems* and *Probability and statistics* with standard contents, we offer three specialized modules that are worth to be described in detail.

Searching Web and Multimedia Databases This module is recommended for students that are interested in deeper understanding of web search engines. In particular, text, hypertext, and multimedia retrieval techniques are explored. The retrieval techniques are described in three layers: theoretical (model), algorithmical, and application. Then, in experimental projects, the students can implement the methods and employ them in various web applications. Students gain basic knowledge concerning retrieval techniques on the web, where the web environment is viewed as a large distributed and heterogenous data repository. In particular, the students will understand the techniques for retrieving text and hypertext documents (the web pages). Moreover, they will be aware of similarity retrieval methods focused on heterogenous multimedia databases (unstructured data collections, respectively).
Lectures Topics: Introduction. Web infrastructure. Data distribution on the web. Web document types. Web retrieval modalities. Text retrieval - Vector model. Implementation. Hypertext retrieval. Hypertext ranking. Semantic web. Implementation. Recommender systems. Collaborative filtering. Multimedia databases. Text- based multimedia retrieval. Similarity as a model for content-based retrieval. Feature extraction from multimedia documents. Similarity queries and their integration into SQL. Indexing metric similarity for efficient multimedia retrieval. Approximate similarity retrieval. Advanced models - nonmetric similarities, dynamic similarities & queries, etc. Text retrieval - Boolean model. Implementation. Search engine optimization.

Information Systems Design The aim of the module is to inform students about the types of information systems (IS) and their practical implementation aspects. After completing the module, students will be able to match the needs of different market segments (customers) with applications of existing technologies (databases, programming languages, GUI etc.). Knowledge obtained in this module can serve as the fundamentals for IS project management, as well as for selecting an optimal solution (including technology), choosing a supplier, and supervising the delivery of an information system by a third party. Students will learn various ways and methods of design and implementation of information systems. They will gain overview in different kinds of ISs, appropriate technologies, and practical areas of their deployment. Students will also be able to consider customers' requirements on the IS and choose proper technologies for their implementation.
Lectures Topics: Introduction to Information Systems, glossary and definitions, subject motivation. Project way of realization, methodology. Postimplementation stage of project realization. Legal questions of information systems implementation, specification of information systems delivery to different segments. Market verticals, information systems types. Information systems for small- and medium-size companies. Information systems for large-size companies. Information systems for manufacturing companies. Information systems for Customer Relationship Management (CRM). Information systems for the public administration. Geographical information systems. Information systems for content management. Business intelligence. Methods of Information Systems implementation, process control.

Data Mining The module aims to introduce students to a rapidly developing field - knowledge discovery in data. Students are introduced to the basic methods of discovering knowledge in data. In particular, they learn the basic techniques of data preprocessing, multidimensional data visualization, statistical techniques of data transformation, and fundamental principles of knowledge discovery methods. Students will be aware of the relationships between model bias and variance and will know the fundamentals of assessing model quality. Data mining software is extensively used in the module. Students will be able to apply basic data mining tools to common problems (classification, regression, clustering).
Lectures Topics: Introduction to data mining, data preparation, data visualization. Cluster analysis. Statistical analysis of data. Data model, nearest neighbour classifier. Training, validation and testing, model's quality evaluation. Artificial neural networks in data mining. Unsupervised neural networks - competitive learning. Probability and Bayesian classification. Decision trees and rules. Neural networks with supervised learning. Combining neural networks and models in general. Data mining in the SAP and SPSS products. Text mining, Web mining, selected applications, new trends.

3 Core modules

There are seven obligatory modules in the knowledge engineering master specialization. The stress is given to the data extraction and preprocessing from various sources, advanced methods for data mining and to the deployment of solutions in a company or institution. Bellow is a detailed description of four selected modules that can be find in the knowledge engineering specialization only.

Data Preprocessing is crucial for successful data processing and takes a lot of time - usually more than the data processing itself. Knowledge of algorithms for extraction of parameters from various data sources is a fundamental part of knowledge engineering, Students learn to prepare raw data for further processing and analysis. They learn what algorithms can be used to extract parameters from various data sources, such as images, texts, time series, etc., and learn the skills to apply these theoretical concepts to solve a specific problem in individual projects - e.g., parameter extraction from image data or from Internet. *Lectures Topics:* Data exploration, exploratory analysis techniques, visualization of raw data. Descriptive statistics. Methods to determine the relevance of features. Problems with data dimensionality, noise, outliers, inconsistency, missing values, non-numeric data. Data cleaning, transformation, imputing, discretization, binning. Reduction of data dimension. Reduction of data volume, class balancing. Feature extraction from text. Feature extraction from documents, web. Preprocessing of structured data. Feature extraction from time series. Feature extraction from images. Data preparation case studies. Automation of data preprocessing.

Pattern Recognition is the prerequisite for modern approaches to artificial intelligence, machine perception, computer graphics, and many other related disciplines, such as date mining, hypermedia, etc. Students will learn elements of pattern recognition, Bayesian decision theory, learning theory, parametric and nonparametric classifiers, support vector machines, classification quality estimations, feature selection, and cluster analysis. The aim of the module is to give a systematic account of the major topics in pattern recognition with emphasis on problems and applications of the statistical approach to pattern recognition. Students will learn the fundamental concepts and methods of pattern recognition, including probability models, parameter estimation, and their numerical aspects. *Lectures Topics:* Elements of pattern recognition. Basic pattern recognition concepts. Bayesian decision theory. Learning theory. Parametric classifiers. Non-parametric classifiers. Support vector machines. Hierarchical classifiers. Pattern recognition using neural networks. Classification quality estimation. Dimensionality reduction. Feature selection. Cluster analysis.

Computational Intelligence Methods The module gives an overview of basic methods and techniques of computational intelligence that stem from the classical artificial intelligence. Computational intelligence methods are mostly nature-inspired, parallel by nature, and applicable to many problems in knowledge engineering. Students will understand methods and techniques of computational intelligence that are mostly nature-inspired, parallel by nature, and applicable to many problems. They will learn how these methods work and how to apply them to problems related to data mining, control, intelligent games, optimizations, etc. *Lectures Topics:* Introduction to computational intelligence, its uses. Algorithms of machine learning. Neural networks. Evolutionary algorithms, evolution of neural networks. Computational intelligence methods: for clustering, for classification, for modeling and prediction. Fuzzy logic. Swarms (PSO, ACO). Model grouping and combining. Inductive modeling. Quantum and DNA computing. Case studies, new trends.

Knowledge discovery from databases The aim of this module is to bring together and to systematize the knowledge and skills received in specialization-specific knowledge engineering modules, as well as to demonstrate its use on real-world problems. Students will learn to solve the problems of data mining in a complex way, starting from preprocessing of raw data to presentation of mined knowledge and deployment of data mining solutions in a company. They will get skills to apply their knowledge of data mining to solve various problems. Furthermore, they will acquaint with the GUHA method and observational calculi. *Lectures Topics:* KDD (Knowledge Discovery from Databases), CRISP-DM. Self organizing data mining. GUHA Method. The LISp-Miner system. Observational calculi. Applications of selected methods of knowledge engineering in KDD. Presentations of companies and institutions describing their use and handling of knowledge.

The other three obligatory modules are *Functional and Logical Programming*, *Advanced Information Systems* and *Advanced Database Systems*.

Besides these seven modules, several useful modules such as *Statistics for Informatics* are also part of programme curriculum.

4 Optional and supplementary modules

We list selected modules from other specializations at our faculty, allowing students to fine-tune their knowledge engineering specialization.

- Web Data Mining
- Retrieval of Multimedia Content on the Web
- Web 2.0
- Semantic Web
- Integration in Information Systems
- Management of Business Informatics
- User Interface Design
- Advanced Algorithms
- Modern Internet Technologies
- Web Services and Middleware
- Modelling of Business Processes
- Information Security
- Security and Secure Programming

Our programme also offers several supplementary modules. These modules are often supervised by industrial experts. Individual and team project modules allows students to work on larger projects during their study. Such experiences are necessary for their success at job markets.

5 Conclusion

The flexibility of the public education is increasingly important as job markets change rapidly. Our faculty offers new specializations designed for future informatics positions.

Teaching Parallel Programming for IT Students
Why, When, and How?

Ahmed Wagueeh (ahmedx.wagueeh@intel.com), Intel Corporation, Intel Academic Community

Abstract

With the increasing demands for performance and availability of super-computers capabilities in almost every processor-based device around us, like PCs, notebooks, and even hand-held devices; parallel programming is now a must-to-have tool that all IT students should master to be able to explore and utilize existing and future platforms. These platforms that they have to understand and program through their academic and/or practical career

This paper is a result of Intel initiative to help academic community to integrate parallel programming concepts into various curriculums. Part of this paper is based on cooperation with Saint Petersburg University in Russia to develop educational labs and exercises for junior students, and to make them available for other universities across Russia and Europe

Through this paper, we will focus on answering three main questions of:

Why it is important to teach parallel programming?

At which we will cover why industry moved to multi-core systems, relation between new hardware and software development model, and the increasing importance of parallelism

When to start teaching parallel programming?

Here we will discuss importance of early introduction of parallel programming concepts to IT students, which will lead to form their thinking and design skills to adequate parallelism as a second nature. Also, we will discuss which courses are affected by wide availability of multi-core based systems

How to integrate parallel programming into standard IT courses?

Finally, we will present several ideas of how to integrate parallel programming concepts into existing standard IT courses, with a case study of Saint Petersburg University project

Although this paper is based on Intel funded project and under full supervision of Intel Corporation, the presented results and concepts are generic and can be easily applied to almost every available technology

Reporting on a Successful Experience in Educating Informatics Students in Communication Skills: Beyond a single subject towards the full curriculum

Ma. Jesús Marco-Galindo[1], Joan A. Pastor-Collado[1], Rafael Macau-Nadal[1]

mmarcog@uoc.edu, jpastorc@uoc.edu, rmacau@uoc.edu

[1] Informatics, Multimedia and Telecomunicación Studies, Universitat Oberta de Catalunya (UOC**),**
Rambla Poblenou, 156, 08018, Barcelona, Spain.

Abstract: This paper presents an on-going experience in curricular innovation regarding learning written communicative skills for the informatics engineering degrees provided by the Open University of Catalonia (UOC). After reporting on the design and deployment of an optional subject established for such a purpose, we analyze its results in the period ranging from 2004 to 2010. Given the overall positive results, we now regard this initiative as a first phase to be extended beyond its current state. Thus, we have decided to evolve that subject into a mandatory one for all of our ICT engineering students, and to design a new more ambitious transversal learning system for them to follow during the rest of their curricula. This system is starting to be deployed as the new ICT degrees aligned with the European Higher Education Space are deployed at UOC. In this paper we point out the rationale for such an approach and the basic ideas and components of such a new learning system.

Keywords: on-line and virtual learning; transversal and soft skills; communication skills and competence; evidence-based evaluation; e-portfolio, information and communication technologies (ICT).

1. Introduction

This paper describes our experience of an initiative by the Informatics, Multimedia and Telecommunication Studies department at the Open University of Catalonia (UOC), which began in early 2002, when we decided to address the written communication difficulties that many of our engineering students shared. In our daily work, we are quite often faced with essays, reports or projects that are difficult to read and understand. This is usually not due to the complexity of their content but to the way in which they are written, which shows that our students, future computing engineers, often lack the ability to write sufficiently clearly and precisely.

In 2002 we decided that it was appropriate to attempt to improve this situation in our own particular context by designing a new subject specifically to improve our students' communication skills and introducing it into the computing engineering curricula. At that time, several international engineering studies projects such as the Tuning project [13] and proposed curricula [2] had already recognized that teaching communication skills to computing students was necessary. Currently, this is also being called for by reports that have studied the professional field, such as the recent 2009 Job Outlook from the National Association of Colleges and Employers (NACE) [9].

We then approached the improvement of the situation by taking into special consideration our particular context, being the first virtual or distance learning university in Spain having introduced university-level computer science education. At that time our computer science engineering degrees were all the official ones in Spain, namely Technical Engineering in Management Informatics, Technical Engineering in Systems Informatics, and Higher Engineering in Informatics. More recently, we have extended our portfolio with a fourth engineering degree, Technical Engineering in Telecommunication, for which we have also adapted the subject. In total, the four ICT related engineering careers at UOC involve more than 5000 students.

Regarding the subject under consideration in this paper, in [7] we presented its basic lines to the Spanish informatics community. Our first step was to design and deploy a new optional subject, initially named “Communication competence for informatics professionals”, which was taught from the second semester of the course 2004-2005. This subject was later on slightly adapted to incorporate telecommunication students and renamed as “Communication competence for ICT professionals.”

We have recently presented a first analysis of this experience in [8], partly included in this paper, which globally represents a very successful case, both in quantitative and qualitative terms. From such an analysis, plus the acquired practical knowledge, we are now advancing a further step in proposing a more ambitious learning system, in a transversal way, in order to extend the learning of the communication competence beyond the mentioned subject, so that students can continue to improve their communication competence in a progressive and incremental manner during their remaining curricula. This new system has just started this second semester of course 2009-2010, in the new ICT degrees that have been designed for the European Higher Education Space (EHES).

The new approach pretends to facilitate practical, continuous and progressive learning of the communication competence along the whole career, starting with the dedicated subject, now mandatory for all students, and then continuing with an evidence-based evaluation scheme, supervised by a dedicated tutor and supported by an ad-hoc information system named ‘electronic transversal learning portfolio’, *etransfolio* for short. While the student progresses, he uses the *etransfolio* to provide evidences of his improvements regarding of his communication competence level, with the supervision of his tutor.

The rest of this paper is organized as follows. After a more detailed description of our experience of teaching communication skills in our initial optional subject, which now represents the first phase of our overall experience, we present and analyze the results of such first phase. We then present the basic rationale and details of our new more transversal learning system, which starts with the specific subject, now mandatory for all of our new ICT students; given its novelty, we may not yet provide any analysis or results on this second phase, which has just started. Finally, we shortly present our conclusions and the work that we want to do in the future along this topic.

2. First phase: the initial subject

2.1. Designing the subject

As we have said, deciding and designing a new subject for all of our informatics curricula (later on also our telecommunication career) was the first step to address the improvement of the communication competence for our students, so that they would be more competent in writing academic and professional documents.

Back in 2002, prior to deciding on the nature and contents of our new subject, we reviewed the main references for informatics curricula in Spain, in Europe and internationally. As mentioned above, these highlighted the importance of learning several generic skills, particularly communication skills. Significantly, the need for education in communication skills was explicitly recommended in the latest version of the ACM/IEEE Computing Curricula 2001 [2], which in point 12.4 insists on the importance of knowing how to communicate ideas effectively. However, none of the references provided any indication of how such an education in communication skills could be addressed. We also reviewed literature on innovative education in informatics, from which we drew some interesting case studies, before going on to analyze some relevant university initiatives, especially in English-speaking countries, such as those shown in [5,6,10,11,14]. These generally tried to improve the communication competences of their students through explicit evaluation of different aspects of writing in some or all of the subjects of their computing curricula. See [8] for a detailed and more extended account of this state of the art.

All this preliminary work was inspiring and each of the published experiences appeared to contain very good options for its respective context. However, given the limitations of our own knowledge and our particular context, briefly described above, we decided to complement the lessons drawn from those experiences with an initiative of our own: developing a new subject exclusively devoted to improving the communication skills of our type of students in a practical yet formal way. As we have explained, we then decided that before contemplating other more ambitious strategies, we would introduce an optional subject, which would help us to test the whole issue and the contents with those of our students that decided to enroll for it.

2.2. Subject approach and contents

From the beginning we wanted to design a subject that was theoretically sound in the sense of having a solid, relevant linguistic grounding. This theoretical basis would enable us, in a rationally structured way, to provide all the basic knowledge that underpins good written communication and that would become the reference point for students in their future writing tasks, inside and outside the subject, in other subjects and in their daily work. We thought that it was appropriate to provide our students with serious, formal help in improving their writing skills, particularly bearing in mind their specific profile. For addressing such an endeavour, we organized an interdisciplinary team formed by two computer science engineers (M.J. Marco and R. Macau) with both professional and academic experience, together with two Philology academic specialists (M.J. Cuenca and F. Nicolau) that had long experience in teaching communication competence but in other areas very different from engineering.

2.3. Learning environment

UOC is a distance university that bases all its programs upon virtual learning. In this type of studies, the majority of students are adult people that may not attend physical classes or that find more convenient to study on-line. They study for themselves at home, by interacting on-line with other students, and they organize their learning efforts with the help of subject guides and the support of tutors and consultants from the university. In contrast with other virtual learning models, UOC's model is centred upon interaction with a pedagogical system based in asynchronous space and time encounters that rely in three elements: the existence of sound *learning materials*; *teaching action* and *continuous evaluation*. Teaching and learning happens within a virtual classroom organized around four independent sections, namely: planning, communication, resources and evaluation.

Within the subject, learning action is guided by a professor that we name *consultant*, specialist in the area, while outside the subject, for more transversal or general issues, guidance is provided by a *tutor*. The consultant is responsible for the learning plan, answers questions and doubts regarding the subject and is also responsible for the continuous and final evaluation of students. Learning is driven by the study plan that includes learning goals, contents, didactic resources, learning methodology and evaluation model.

At UOC the profile of our students is significantly different from that of students in most other universities requiring physical presence from their students. In our ICT careers, most of our students are adults, in their 35-40 years of age, most married with families, already working as ICT professionals, and some already with a university career outside ICT.

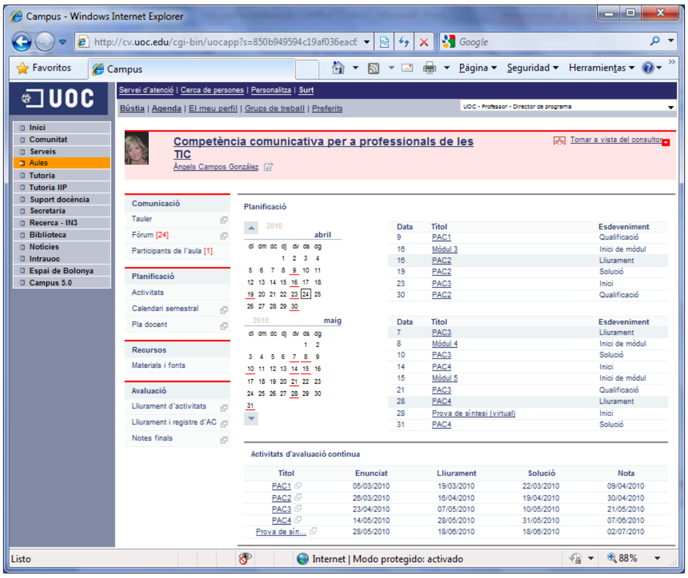

Fig. 1. Virtual classroom of our subject ‘Communication competente for ICT professionals’

2.4. Subject approach and contents

Taking into account our learning environment and the characteristic profile of our students, the established team designed the subject, its goals and contents as well as the learning and evaluation systems.

Firstly, attending to the virtual character of the subject and of our studies at UOC, the learning materials were planned and written in a way that they would be sufficiently clear for the student, avoiding the need for additional explanations or for physical presence of a teacher. Along the same line, the materials should be self-contained, so that students would not need other complementary sources for understanding the main concepts in written communication, unless they wanted to deepen their knowledge beyond the subject learning goals.

Secondly, the subject goals were set up in close alignment of the specific needs of our students regarding their written communication typical weaknesses. Thus, subject contents were determined from the identification of the main problems observed in texts written by the students. In addition, good and bad communication examples were picked from real texts drawn from actual professional and academic situations that the students could encounter. In this way the practical and applied character of the subject was also stressed.

The communication weaknesses of the students typically range a varied set of problems, from syntactic ones to problems regarding orthography, sequencing, registering, etc. While some of those problems, specially orthographical issues, have other means to solve them, we soon saw that there was not enough space in a single subject to cover appropriately the wide set of issues raised. Thus, the most important weaknesses were identified in [4] in order to determine the kernel contents of the learning materials. Table 1 gives a short account of those most common problems, classified along the properties of good written communication. From this prior analysis, the team prepared a proposal for a selection of skills based on Text Linguistics [4], giving shape to a syllabus organized around three core themes:

- ✓ Basic concepts (communication competences, text, text properties, types of text).
- ✓ General characterization of some typical texts drawn from various computing disciplines, both from academic and professional situations.
- ✓ Production techniques for specialized texts (production process, context awareness, coherence, cohesion).

Table 1 – Written communication problems initially detected.

Property	**Problems**
Adequacy	✓ Informal or slang written forms used in formal texts. ✓ Lack of adequacy to specialized language, by using words with several meanings without clarification, and by using terms with a presumed meaning different from their actual one. ✓ Lack of homogeneous treatment in the use of the grammatical person and in other linguistic solutions adopted. ✓ Errors related to the typical characteristics of formal registration, specially those related to the knowledge of grammatical rules, specially in the use of relatives.
Coherence	✓ Lack or organization in the information provided as the consequence of lacking a previous script or plan. ✓ Incorrect writing of sections for introduction and conclusions. ✓ Generic ideas, often repeated or irrelevant with regard to the main theme of the text. ✓ Lack of symmetry in the structuring of the text and in the form of expression.
Cohesion	✓ Incorrect use of connectors. ✓ Use of referential elements without a clear antecedent. ✓ Lack of lexical precision, such as abuse of generic terms. ✓ Not maintaining the correlation of verbal times. ✓ Altering the basic order in phrasing, and faulty construction of sentences, often too long and with problems in their grammatical concordance. ✓ Incorrect punctuation, specially using an excess of, or too few periods, misuse of colons and errors in the citation marks. ✓ Irregular use of capital letters and in graphical marks.

In addition to the learning materials, the subject design was completed with an ad-hoc learning methodology, a specific continuous evaluation system for the subject, and a specialized team of consultants, all of them academics with teaching and research experience in Applied Linguistics from other universities different from UOC. Since then, and in order to ensure personalization of the subject to informatics, this team as well as the subject deployment and occasional updates have been led and coordinated by an informatics engineer (M.J. Marco, first author of this paper). Thus, since the inception of our subject, we keep this multidisciplinary team with some additions when new student groups need to be addressed. We consider this multidisciplinarity as a key factor to the uccess of the subject, given its formal linguistic contents but landed with practical and applied examples drawn from real computing situations, very similar to the texts that our students are asked to produce both in the professional life and in their academic learning progress.

The mentioned learning methodology first requests the students to address the formal contents of each point of the materials, including the examples given, and then demands from them a set of writing activities relevant to the points studied. Given its incremental and cumulative nature, learning of the writing concepts and techniques is only gained by following the guidelines and by doing the writing exercises.

Continuous evaluation in the case of this subject is based in four tests (*continuous evaluation tests* or PACs) and of a final practical report. Each PAC includes structured written phrasing exercises of incremental difficulty and cumulative value, both within a single PAC and between consecutive ones. The final practical report consists in the writing of a basic specialization report where the student must show his proper use of those written communication skills and specialized text production techniques learned in the subject.

The prior work related to the overall design of our subject and of the learning materials created for it took a couple of years and the subject could be launched at the second semester of course 2004-2005. More recently, as a consequence of the positive diffusion made by our students to their working mates, some of these ICT professionals (not studying at UOC) approached our Studies with the demand for more reachable materials for them to use by themselves. From such a request we decided to create a public book [3], by adapting our previous materials, for any ICT professional to improve their written communication skills and that of their teams. The book, first edited in Catalan and now being translated and adapted to Spanish, has also been distributed to the rest of university informatics schools for them to consider in their new curricula, since all of them are also starting to address in a more formal way the education of their ICT students, younger and with less experience than UOC's, in communication skills.

2.5. Results from the experience with the subject

Over the six years we have been teaching *Communicative Competence for ICT Professionals* it has been taken by 1302 students, a very high number for an optional subject, at least in our particular context.

2.5.1. Quantitative analysis

The quantitative data that allow us to analyze the evolution of the subject over the 2004/10 period basically refer to three indicators: registration, academic performance and student satisfaction.

1) Registration

Registration numbers (Table 2) confirm that in the curricula where it is offered, this subject has a very high enrolment. In fact, it is our optional subject with the highest enrolment. When we introduced the subject we thought that it would be taken by a minority, with a maximum of 60 students per year. From the first semester, these expectations were far surpassed. The qualitative data on student interests explains the reasons for these figures, as will be discussed further on.

Table 2 – Student registration

Semester	Number of students
2004-05/2	99
2005-06/1	148
2005-06/2	158
2006-07/1	131
2006-07/2	99
2007-08/1	54
2007-08/2	166
2008-09/1	147
2008-09/2	177
2009-10/1	121

2) Academic performance and completion of continuous assessment tasks

In terms of the percentage of students that pass the subject, academic performance was very high throughout the period, as it has always been around 80%. Also, the percentage of students completing the continuous assessment (CA) tasks shows a steady, stable trend at around 90%.

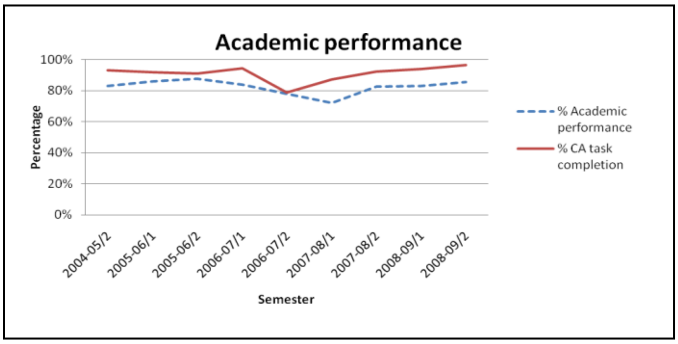

Fig. 2. Academic performance

3) Student satisfaction

At the end of each semester, students are asked to complete online a standard university survey of satisfaction. The average response in recent years has been 23.4%, which is a normal rate for this type of online survey.

The satisfaction survey is structured in four blocks:

- ✓ General satisfaction with the subject (content, work load and satisfaction of expectations)
- ✓ Satisfaction with the consultancy (study plan, facilitation of the learning process, coherent customized assessment)
- ✓ Satisfaction with the teaching-learning resources (updating of materials, suitability of tasks, usefulness of the classroom content)
- ✓ Satisfaction with the form of assessment (coherence of the continuous assessment tests).

The degree of general satisfaction with the subject remained steady at around 70%. This is high, bearing in mind that the subject is not at all common in computing curricula and is substantially different from the other (technical) subjects. Besides, it is well above the 63% average for subjects in these degrees.

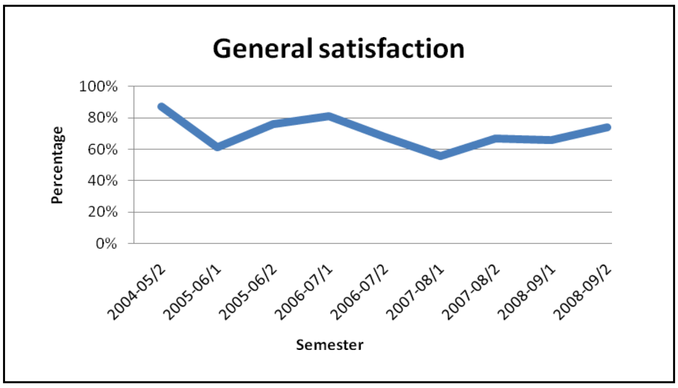

Fig. 3. General satisfaction

Satisfaction with the *learning resources* shows a downward trend. On analyzing the data by item, the information sources space indicator has been found to obtain the worst results and, indeed, shows a clear downward trend, reaching 40.67% in the last semester. This can be explained by the fact that the initial writing skills level of the students is increasingly low [12]. Therefore, each semester they ask for more complementary resources to improve their weaknesses in spelling and basic grammar, both of which are aspects that, while important for good text production, are not the purpose of this subject.

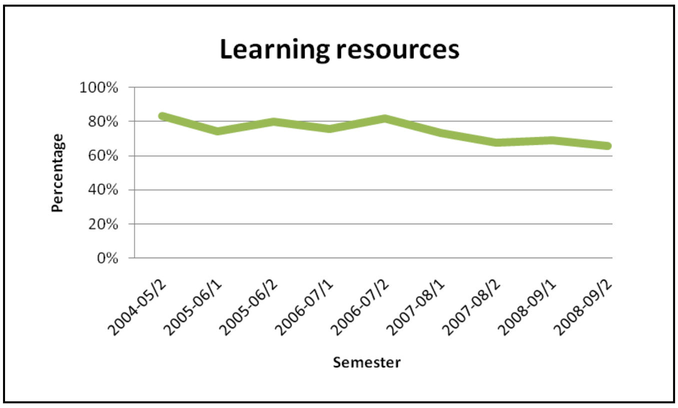

Fig. 4. Learning resources

The general satisfaction with the *consultancy* is very high and always near to 80%. In fact, this is the most highly rated aspect of the subject, fundamentally because of the individual feedback that the consultants give when correcting the tasks.

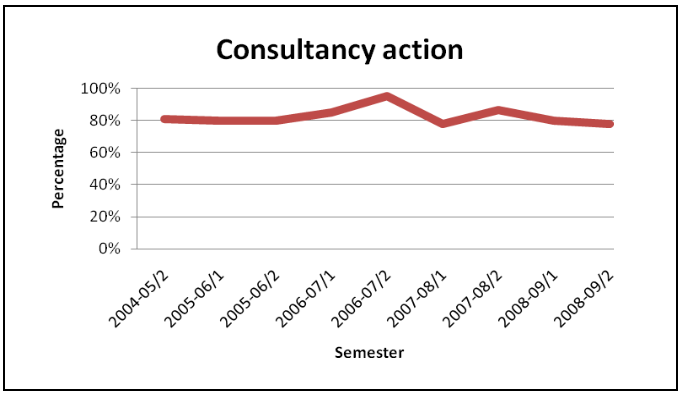

Fig. 5. Consultancy action

Satisfaction with learning outcomes *assessment* also follows a high trend around 80%.

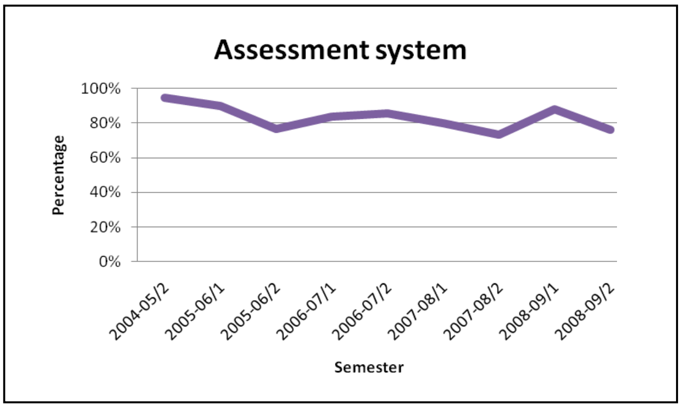

Fig. 6. Assessment system

2.5.2. Qualitative analysis

In addition to the data collected by the university, at the end of each semester the students make a final assessment of the subject at the request of the consultants. In this way we receive their comments, complaints, needs, suggestions, thanks, etc. by means of messages sent to the classroom forums. These provide us with very valuable information on their initial expectations of the subject, their interest in the subject and their opinion on any aspects of the process: what has been missed, how they have been guided by the consultant, difficulties with the content, etc.

Unlike the institutional satisfaction survey, here the students give their opinion freely and openly in terms of both format and content. We can therefore grasp specific, concrete aspects of the subject and can also bring to light any concerns, opinions or suggestions that the students may have. However, being qualitative data, this information is more difficult to analyze. For this purpose we use a qualitative analysis tool called Atlas.ti (http://www. atlasti.com).

Initial analysis of the comments received from 131 students allows us to classify the main items that the students mention. For space limitations, we refer the reader to [8] for a full account with testimonials representing qualitative feedback provided by our students. In short, the issue that students mention most is the excellent attention that they receive from the consultants. They also perceive that what they are learning will be (or already is) useful in their work.

3. Second phase: learning communication skills along the curricula

With the above results in mind, so far we may regard the experience with our initial optional subject as an innovative and globally satisfactory one, which has already consolidated after several courses, various adaptations and a big number of students.

In fact, given our good feeling with such a subject addressing such an important skill, we have decided to build further from this subject, which will now become a solid starting point for additional learning along the rest of our students' curricula. In other words, we believe that written communication skills are too important professionally and academically to leave their learning to a single optional subject, and in our new EHES ICT engineering careers we have included this same subject as a mandatory one for all of our students to take, and then we are designing a further learning system for them to continue improving their written communication skills along the rest of their curricula and up until the finish their career with their final career project.

But it is important to make clear that it is only from the very satisfactory results and positive feedbacks from our initial optional subject that we may now go further with a reasonable level of confidence. Certainly, the new situation will be different with a mandatory subject, both in student amounts and motivations, which also implies a bigger supporting team. This will even be more important when first students pass the subject and require continuous follow-up during the rest of their studies.

The fundamental idea behind our new extended learning system is that we want the student to apply his learnt knowledge regarding written communication in any opportunity that he encounters in the rest of subjects that he takes during the rest of his studies. Obviously, he should not do that alone and neither can we rely solely on the consultants of other subjects, who are specialists in the respective topics but not necessarily in written communication. Thus, we are now designing an appropriate learning system with specialized consulting support as well as the technological support in the form of an information system for registering and following the learning evidences that are provided by the student and evaluated and approved by his tutor. In sort, we need a system that, building up from the already existing subject, rightly sized to address a much higher number of students, can provide for an extended pedagogical model for improving our students' communication skills along their full curricula. For such a system to be successful, not only pedagogical but also organizational and technological issues must be considered.

Regarding the pedagogical perspective, the learning and evaluation of transversal skills such as written communication imply an important change in approach. On one side, it requires that proposed learning activities are very much practice oriented, provided in a context as similar as possible to professional or academic practice. On the other side, in our case it implies that we must change the role of our existing subject from being the unique learning framework for written communication skills into being a key but initial starting point for a wider system for learning and improving such skills. A system with a transversal and more interdisciplinary dimension designed to ease the progressive improvement of the skill further outside the initial dedicated subject and to link and relate a selective set of written communication activities scattered along different subjects, while leaving evidence of the student's skill competence level from start to career completion.

The system that we are currently considering is based upon an evidence-based evaluation methodology where the student presents his eventual learning evidences, which are discussed and evaluated by him and a specialized consultant before they may become part of the set of improvement evidences. Thus, it is the student who becomes responsible for thinking explicitly about his communication competence and related improvement opportunities, and for selecting and providing along his curricula those evidences that he considers a proof of his improvements. He must do this only after having past the subject on "Communicative competence for ICT professionals", from which he knows the concepts and techniques of written communication, as well as its competence levels (three in our case: minimum, desirable and excellent) and the indicators describing each of those levels. In our implementation those indicators are directly related to the three textual properties that we have chosen to concentrate on, namely adequacy, coherence and cohesion (see Table 1). All the information regarding evidences, proven properties with levels and indicators, along the curriculum starting with the initial subject, becomes what is technically named the *rubrics* for the written communication skill.

As we have indicated, all along this learning process the student is accompanied by a professor-tutor, who assists the student in the selection of evidences from those texts that the students must produce for different subjects or from professional assignments, and who is responsible for their evaluation and acceptance, or else in providing relevant feedback for the student to learn before he can propose new evidences. Thus, from the organizational perspective, it is necessary to adequately define this new tutoring role as well as the processes and procedures related to evidence production, selection, evaluation and competence level consolidation, among other possible ones.

Finally, in other to manage all the information considered in students' rubrics, a great amount of varied data, quantitative and qualitative, and documents regarding the transversal learning and evaluation in our virtual environment, the institution has developed a new dedicated information system in the form of a so-called *electronic portfolio* (Fig. 7), that we have named *etransfolio*. It is a tool purposefully designed for evidence-based learning and evaluation, where the student and tutor can enter and maintain the learning evidences in various formats, together with their comments and personal reflections. Thus, the set of evidences and information conforming the rubrics for a student's written communication competence constitute a formal witness of his work, evolution and accomplishments in this skill, both within one subject and along the curriculum. Additionally, the *etransfolio* acts as a learning tool by itself, since it favours the student's own reflection, with the assistance of the tutor, on his communication competence levels shown by his evidences, and on the way ahead to their improvement.

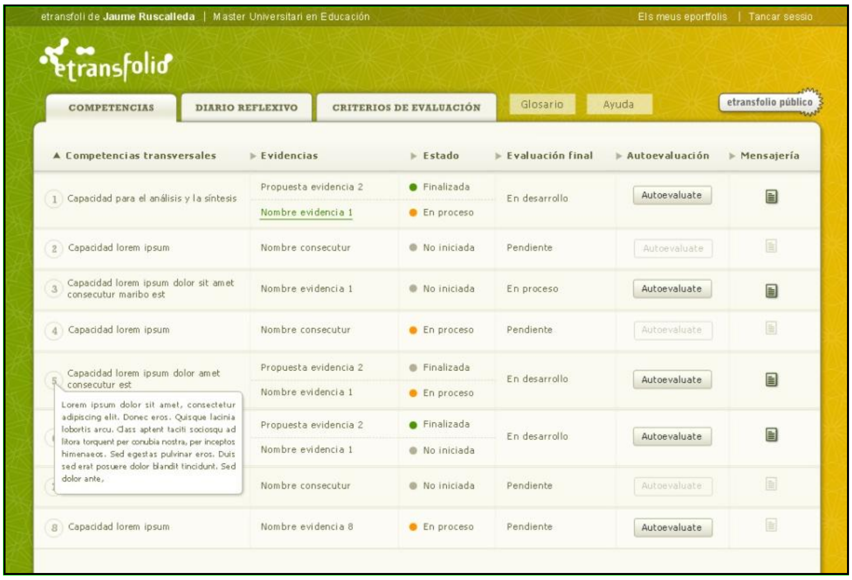

Fig. 7. *etransfolio* main screen

The deployment of this new transversal system has just started this past second semester of course 2009-2010, together with the deployment of UOC's new ICT engineering degrees adapted to the EHES.

4. Conclusions and further work

In this paper we have presented a mature optional subject for improving the communicative competence of ICT engineering students of our university, for which we have also presented an analysis of its quantitative and qualitative results. In addition, we have presented our more ambitious goals for extending such a learning experience for all of our engineering students, by making the subject mandatory and by designing a follow-up learning system for the students to continue improving their communication skills along their remaining curricula.

Given its goals and approach, our initial subject has become a pioneering experience, surely internally at UOC and also in the context of Spanish ICT university degrees. The high number of engineering students that have voluntarily taken it, as well as the very positive results obtained, have converted it in a reference point for the design of newer subjects at UOC dedicated to other transversal soft skills, which are currently being designed for the new EHES degrees.

With the advancement of the EHES, and the importance given to the effective learning of soft skills along the specific knowledge skills of the career, experiences such as the one represented by our initial subject on communicative competence for ICT professionals can be a good reference and starting point for other experiences in different types of higher education institutions, and not just for distance or virtual learning universities.

However, the analysis presented in this paper of our initial subject shows that, even if students give a very good feedback and consider it very useful for their professional life and academic studies, it can still be improved with additional means and resources, dedicated specially for students coming in with very low communication levels. The key success factor of personalized attention by consultants must be kept and taken care of, particularly in the new situation where the number of students will grow significantly.

In our case, in addition to trying to improve the subject presented, we have explained in this paper how we have decided to build upon it, to convert it to mandatory for all of our ICT engineering students, and to design from it a more comprehensive learning system covering the rest of our students curricula. We believe that, once a student has practised and learnt the contents of the subject, he will be in a more knowledgeable position to continue improving his communication skills while studying other subjects from his career, by practicing them in other writing activities and proposing evidences with the assistance of a dedicated tutor, and finally by showing his communicative competence in his final career project.

In this paper we have shortly presented the main ideas for such a transversal learning system, as well as some of the pedagogical, organizational and technological issues that we have already faced in its initial implementation. However, it will be the actual use of this system that will show along the way its strong and weak points, from which analysis we will hopefully be able to improve it. Thus, in the future we plan to follow closely this new scenario and to establish ways and indicators to understand its consequences and to flexibly adapt it in order to improve it.

5. References

[1] Anewalt, K. (2002). Experiences Teaching Writing in a Computer Science Course for the First Time. *Journal of Computing Sciences in Colleges*, 18(2), 346-355.

[2] Computing Curricula 2001 Task Force. *Computing Curricula 2001: FinalDraft* http://www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/cc2001/cc2001.pdf

[3] Cuenca, M.J. & Nicolau, F. (2008). Pensar, Organitzar, Escriure. *Editorial UOC* [supervised and revised by M.J. Marco]. Barcelona.

[4] Cuenca, M.J., Marco-Galindo, M.J. & Nicolau, F. (2004). Destreses comunicatives per a professionals de la informàtica; anàlisi i ensenyament-aprenentatge d'aspectes discursius i lingüístics. Proceedings of the GLAT, Barcelona, 395.

[5] Gersting,J.L. & Young,F. (2001). Shall We Write? *SIGCSE Bulletin*, 33(2),18-19.

[6] Hartman, J.D. (1989). Writing to Learn and Communicate in a Data Structures Course. *SIGCSE Bulletin*, Vol. 21, 32-36.

[7] Marco-Galindo, M.J. *(2008)* ¿Se puede mejorar la competencia comunicativa a través del currículo de ingeniería en Informática?. Una experiencia práctica. *Actas de las XIV JENUI 2008, 677-678.*

[8] Marco-Galindo, M.J., Macau-Nadal, R. & Pastor-Collado, J.A. *(2010) Learning written communicative skills in UOC engineering curricula*. In Proc. of the 1st IEEE Transforming Engineering Education Conference, Dublin, May 2010.

[9] National Association of Colleges and Employers (NACE). *Job Outlook 2009.* http://www.naceweb.org/products/info_pages/joboutlookreport.htm

[10]Pollock, L. (2001). Integrating an Intensive Experience with Communication Skills Development into a Computer Science Course. *SIGCSE Bulletin,* 33, 287-291.

[11] Taylor, H. G. & Paine, K.M. (1993). An Inter-Disciplinary Approach to the Development of Writing Skills in Computer Science Students. *SIGCSE Bulletin*, 25, 274-278.

[12]Tovar, E., Inarejos, E., Cerdà, B., Soto, O., Mas, J. & Marco, M.J. (2009). Estudio comparativo sobre el nivel de desarrollo de competencias transversales en alumnos de nuevo ingreso en enseñanzas de Ingeniería en Informática. *Proyecto EA2008-0043 del Departamento de Educación.*

[13]Tuning Project (2008). Tuning Education Structures in Europe. 2nd. Edition, Publicaciones de la Universidad de Deusto, 2008.

[14]Walker, H.M. (1998). Writing Within the Computer Science Curriculum. *SIGCSE Bulletin*, 30 (2), 24-25.

SOFTWARE ENGINEERING DISCIPLINES IN COMPUTING CURRICULA - AN EXPERIENCE OF TWO ROMANIAN UNIVERSITIES

BAZIL PÂRV
Babeş-Bolyai University Cluj-Napoca, Romania

VIOREL NEGRU
West University Timişoara, Romania

ABSTRACT. This paper describes the experience of two Romanian universities, Babes-Bolyai University of Cluj-Napoca, and West University of Timisoara, in adopting SE courses in their computing curricula.

1. INTRODUCTION

There is universally understood that software engineering (SE) is an important knowledge area of a Computing Curriculum, as CS 2001 [CS2001] and CS 2008 [CS2008] agree. According to [CS2008],

> Software engineering is the discipline concerned with the application of theory, knowledge, and practice for effectively and efficiently building software systems that satisfy the requirements of users and customers. Software engineering is applicable to small, medium, and large-scale systems. It encompasses all phases of the life cycle of a software system. The life cycle includes requirements analysis and specification, design, construction, testing, deployment, and operation and maintenance.
>
> Software engineering employs engineering methods, processes, techniques, and measurement. It benefits from the use of tools for managing software development; analyzing and modeling software artifacts; assessing and controlling quality; and for ensuring a disciplined, controlled approach to software evolution and reuse. Software development, which can involve an individual developer or a team of developers, requires choosing the tools, methods, and approaches that are most applicable for a given development environment. The SE toolbox has evolved over the years; for instance, the use of contracts (such as a 'requires' clause, an 'ensures' clause, class invariants, etc.) is now regarded as good practice.

The IEEE official document [SWEBOK] contains a more detailed definition of SE. Over the years, the number of SE topics/courses and their importance in CS curriculum were growing continuously.

The structure of paper is as follows. After this introductory section, the second presents some historical data regarding informatics programs in Romanian higher education institutions. Based on the SE topics of CS 2008, the third section contains a comparative study of SE courses taught at two universities. Last sections describe some future plans and conclusions.

2. INFORMATICS EDUCATION IN ROMANIAN UNIVERSITIES

Romanian undergraduate programs in *Informatics* started in 1971, being organized by Faculties (Departments) of Mathematics in four main universities (Bucharest, Cluj-Napoca, Iasi and Timisoara). In more than 20 years, computer science curriculum, inspired originally by mathematics curriculum, evolved into a self-contained one. In the last decade of the 20th century, separate programs in *Mathematics-Informatics* were organized, with a roughly 60%-40% ratio of mathematics-computer science disciplines. Before 2005, these programs were 8 semesters long, and today they have 6 semesters, according to Bologna document. Recently, new undergraduate programs emerged, like *Applied Informatics* (6 semesters) and *Information Engineering* (8 semesters). Technical Universities have their own *Computer Science* programs with many variations and names, evolved mainly from older *Computer Engineering* ones. Also, there are academic programs in *Economic Informatics and Cybernetics*, taught at Economic Universities and Departments.

One-year graduate programs in *Informatics* started in 1976, continued in the '90s in the form of advanced studies programs, and followed in the new millennium by one-year master programs. Since 2005, all master programs have 4 semesters.

Informatics education in Romanian high-schools includes IT-related disciplines (how to use computers and computer applications) and Informatics disciplines. The last ones are taught in Mathematics-Informatics classes, and refer to programming languages, algorithms and data structures, graph theory, databases, and web design. Consequently, most of the first-year students in Informatics have a decent knowledge of programming languages.

West University Timisoara (WUT) and Babes-Bolyai University Cluj-Napoca (BBU) are two of the four "classical" Romanian Universities involved since 1971 in Informatics education. The current academic offer of WUT and BBU in Informatics (see [WUT2010] and [BBU2010]) include undergraduate programs in *Informatics* (I, taught in Romanian, English), *Mathematics-Informatics* (MI, Romanian), and *Applied Informatics* (AI, Romanian) at WUT, respectively in *Informatics* (Romanian, English, Hungarian) and *Mathematics-Informatics* (Romanian, Hungarian, German), and *Information Engineering* (IE, Romanian, English, Hungarian) at BBU. Informatics master programs organized at WUT are *Artificial Intelligence and Distributed Calculation* (AIDC, Romanian, English), *Software Engineering (SE, Romanian)* and *Computer Science Applied to Sciences, Technology, and Economics* (CSA, Romanian) while BBU offers four programs taught in English (*Component-Based Programming* CBP, *Intelligent Systems* IS, *Formal Methods* FM, *Modeling and Simulation* MS), two in Romanian (*Databases DB* and *Distributed Systems* DS), and one in Hungarian - *Optimisation of Information Models* OIM.

3. LEARNING FROM EU EXPERIENCE

As Steve McConnell points out [MCC2004], the world's first master SE program was introduced by Seattle University in 1982, while the first undergraduate SE program was launched by the Sheffield University in the United Kingdom in 1988. Regarding SE programs in Europe, [DJM2004] contains a very detailed analysis of their status at the beginning of third millenium.

Curricula for undergraduate Romanian academic programs in informatics (computer science) is largely inspired from similar programs offered by well-known European universities from UK, France, Germany, Netherlands, and Switzerland. Many of the recent developments were performed in the frame of different forms of cooperation between Romanian universities and their EU partners.

4. SOFTWARE ENGINEERING DISCIPLINES vs ACADEMIC PROGRAMS

Table 1 contains main SE courses in the current curricula of academic programs (AY 2010/2011) offered by WUT and BBU, containing CS2008 SE topics. Many of these courses were adopted in the last ten years.

As a remark, SE topics as *Software evolution*, *Software reliability*, *Risk assessment* and *Robust and security-enhanced programming* are not covered by the disciplines in the current CS curricula in our universities.

Table 1: SE topics in WUT and BBU computing curricula

SE topic	**WUT courses/programs**	**BBU courses/programs**
undergraduate programs		
SE fundamentals	Software engineering (I, AI)	Software engineering (I, MI, IE)
Software design	Object-oriented programming (I, IA)	Object-oriented programming (I, MI, IE) Design patterns (I, MI, IE) Advanced programming methods (I, IE)
Using APIs	Environments for design and programming (I, IA)	Environments for design and programming (I, IE)
Tools and environments	Software project management (I, IA)	CASE tools (I, MI, IE)
Software processes	Software engineering (I, AI)	Software engineering (I, MI, IE)
Requirements specifications	Software engineering (I, AI)	Software engineering (I, MI, IE)
Software verification and validation	Software testing (IA)	Software verification and validation (I, IE)
Software project management	Software project management (I, IA) Team software project (I, IA)	Team software project (I, IE)
Specialized systems		Virtual Instrumentation (IE) Integrated Information Systems (IE)
master programs		
Software design	Software design (AIDC, SE) Software engineering architectures (SE) HCI design (SE)	Software architecture (CBP,IS,FM,MS) Framework design (CBP,IS,FM,MS,OIM)
Requirements specifications		Software modeling (FM) Behavioral modeling of software systems (CBP, IS, FM)
Software project management	Processes and management in SE (AIDC, SE)	Management of software projects (DB, DS)
Component-based computing		Component-based programming (CBP)
Formal methods	Models and formal methods in SE (SE)	Formal methods in programming (CBP, FM) Formal methods in programming languages (FM) Formal methods in parallel programming (FM)
Software reliability	Software engineering quality assurance (SE)	
Specialized systems	Workflow technologies (AIDC, SE, CSA) Distributed systems (AIDC, SE)	Decision support systems (CBP) Workflow systems (CBP)

In the last decade, main directions of change were: improving the content of existing programming courses with SE topics, the appearance and diversification of modeling disciplines, and the inclusion of practical SE disciplines.

5. FUTURE PLANS

Our common experience shows that the adoption of SE disciplines in our academic programs in Informatics increased their attractivity and offered our graduates competitive advantages in their search for jobs. In this respect, the improved links with industry is beneficial, allowing our students to have a direct contact with real-world, in terms of internships or practical works. In the same time, the feedback provided by employers helps us to improve continuously our academic offer. As a matter of fact, both universities intend to include new SE disciplines in their programs, as detailed below.

The Department of Computer Science of BBU is planning a new master programme in *Software Engineering*, starting with AY 2011/2012, covering all major topics in the table above, including *Software Evolution*, *Software Reliability* and *Risk Assessment*. Also, a new course on *Model Checking* will be included in the curricula of CBP master program.

The development of *Software Engineering* master from the Computer Science department of WUT is incremental. The master started 2 years ago based mainly on AICD master courses. and each year about three new courses were proposed. We intend to extend this courses in the topics of *Formal verification*, *Software quality* and *Testing*. Also we will increase the mobility of faculty members inside the “Universitaria” Consortium (BBU and WUT are members together with University of Bucharest and “Al. I. Cuza” University of Iasi).

6. CONCLUSIONS

Since 80s, software engineering has moved significantly toward recognition as a professional discipline. This fact was acknowledged by the rise of graduate and undergraduate SE programs in American and European universities.

The rise of separate SE undergraduate academic programs in Romania is at the beginning. The common situation is to have master SE programs following classical CS programs. This is the case of master SE programs at WUT, University of Bucharest, and “Al. I. Cuza” University of Iasi.

As [MV2004] points out, a separate undergraduate SE program can be carried out with minimal variations of an existing CS program. This is the case with four-year *Information Engineering* programme at BBU, which is very similar to the *Informatics* program in its first three years.

As a first conclusion, we can state that the inclusion of SE-related disciplines in our computing curricula was beneficial, allowing our graduates a better insertion on the job market and providing a better attractivity to employers.

Second, the development of separate SE programs in Romanian universities, either undergraduate or graduate, proved to be beneficial. This encourages us to look forward in considering further improvements in their curricula.

REFERENCES

[BBU2010] Academic Curriculum 2010-11, Department of Computer Science, Babes-Bolyai University http://www.cs.ubbcluj.ro/www/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=228

[CS2001] Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, http://www.acm.org/education/education/education/curric_vols/cc2001.pdf

[CS2008] Computer Science Curriculum Update, http://www.acm.org//education/curricula/ComputerScience2008.pdf

[DJM2004] O. Dieste, N. Juristo and A.M. Moreno, How Higher-Education Systems Influence Software Engineering Degree Programs, *IEEE Software*, July/August 2004, 78-85.

[MCC2004] S. McConnell, *Professional Software Development: Shorter Schedules, Better Projects, Superior Products*, *Enhanced Careers*, Addison-Wesley, 2004.

[MV2004] F. Meziane and S. Vadera, A Comparison of Computer Science and Software Engineering Programmes in English Universities, Proc. of the *17th Conf of Software Engineering Education and Training CSEET04*, 65-70.

[SWEBOK] IEEE. *Software Engineering Body of Knowledge*, 2004. http://www.swebok.org/ironman/pdf/SWEBOK_Guide_2004.pdf

[WUT2010] Academic Curriculum 2010-11, Department of Computer Science, West University of Timisoara, http://www.math.uvt.ro

Bazil PÂRV
Department of Computer Science
Faculty of Mathematics and Computer Science
Babes-Bolyai University
1, M. Kogălniceanu
Cluj-Napoca 400084, Romania
email:bparv@cs.ubbcluj.ro

Viorel NEGRU
Department of Computer Science
Faculty of Mathematics and Computer Science
West University of Timisoara
4, V. Pârvan
Timisoara 400084, Romania
email:vnegru@info.uvt.ro

The Bologna Process and its Impact on MS Programs in Turkey: A Case Study at Atilim University

Ali Yazıcı

Atilim University, Software Engineering Department, 06836, Ankara, Turkey
aliyazici@atilim.edu.tr

1. Introduction

In line with the Lizbon Recognition Convention[1] (2004), Council of Higher Education in Turkey adopted the regulations on May, 2007. Since then, Turkish Universities started to work on credit conversion from the standard credit system to ECTS and produced diploma supplements accordingly to comply with the rules and regulations in the Bologna Process.

Several workshops had been organized by the Council of Higher Education, and the universities themselves with the growing support from the academic and administrative staff. In the Turkish university education system, there are four compulsory subjects (Turkish Literature, I, II, and History of Turkish Revolution, I, II) to be studied by all the undergraduates as dictated by the council. These courses are sometimes non-credit courses or different universities assign varying credit units to these subjects. In addition, engineering students have compulsory Summer Practices (at least one summer session). Summer Practice courses are usually non-credit courses with a Pass/Fail grade. Most of the Turkish Universities adopted the same ECTS credits to these compulsory courses and similarly, non-credit courses, now, have the relevant ECTS credits.

However, until now, most of the work has been restricted to the undergraduate (first cycle) programs, and only a few Turkish universities had considered the MS and PhD programs and their position in the Bologna Process.

This paper, tries to identify the problems associated with the conversion of the standard credit units to ECTS units in the Turkish MS programs in general and as a case study, the current implementation at the Software Engineering MS Program is given.

2. MS Programs ECTS Credits in Europe

Majority of the European Universities is offering MS programs in compliance with the rules of the Bologna Process. The duration of the MS programs is ranging from two semesters (one year) to four semesters (two years) with 60 and 120 ECTS units respectively depending on the program's duration. Some examples from European Universities MS programs are given in the table below:

Table 1. Some MS Programs in Europe with the ECTS Units

#	University	Program	ECTS
1	Ecole Polytechnique Federale de Lausanne, Switzerland	Electrical Engineering	90 or 120
2	Hauto Ecole de Gestion Fribourg, Switzerland	Business Administration	90

[1] http://www.coe.int/t/dg4/highereducation/recognitionof /lrc_EN.asp

3	Universite Catholique de Louvain, Belgium	Computer Science	120
4	University of Lugano, Switzerland	Applied Informatics	90
5	University of Skovde, Sweden	Bioinformatics	60
6	Gotland University, Sweden	International Management	60
7	Vestfold University College, Norway	Microsystems Technology	120
8	TU Berlin, Germany	Computational Neuroscience	120
9	The UPC. Barcelona Tech., Spain	Computing	120
10	Grenoble University, France	Informatics	120

3. Barriers in Turkish MS Programs

According to the framework set by the Council of Higher Education of Turkey, MS programs (with/without thesis) have certain restrictions to be followed:

- The duration of the MS programs is four semesters (two years).
- Depending on the progress of the student, in certain cases, the program duration can be extended to three years.
- The number of courses is at least 7 (21 standard credits) and 10 (30 standard credits) for the thesis and non-thesis options respectively.
- For the thesis option, student should attend a seminar course, and deliver a speech related to his/her master's research topic.
- In the case of the non-thesis option, there is no seminar activity, however, a one-semester graduation project work needs to be completed.

In many universities, 7 and 10 courses per program for the thesis and non-thesis MS programs respectively are very common.

On one hand, the framework set by the council makes the ECTS conversion easy, on the other hand, the 30 ECTS units/semester may create a barrier in the future when standard credit units are removed from the system. This is because of the flexibility of the MS programs in Turkey, letting students register to at least two courses per semester.

The post-graduate programs in Turkey, are managed by the Graduate Schools (Institutes) of Social/Natural and Applied Sciences. Especially, in Enginering, students take courses from different disciplines. Henceforth, the courses should carry the same ECTS unit. This requires a close collaboration between the graduate schools in a university. This becomes a more severe restriction when the student would like to transfer his/her credits across the universities within Turkey. On the other hand, the number of courses to be taken in a program (7 or 10) is only the minimum requirement. Once, these number changes, the ECTS unit requirement is automatically effected. In addition, in the current credit system, Thesis work, Seminar and Graduation Project (for non-thesis) is non-credit courses with a Progress, Unsatisfactory, or Pass/Fail grades respectively.

In Turkey, only a few universities have set forward their ECTS units in the MS programs. Some of these are given in the table below.

Table 2. Some MS Programs in Turkey with the ECTS Units

#	University	Program	ECTS	Thesis ECTS	Course ECTS
1	Mehmet Akif Ersoy University, Burdur	Science Education in Elementary Education	120	40	7.5
2	Uşak Üniversity, Uşak	Geography	120	60	6 or 9
3	Middle East Tech. Univ. - Eindhoven University of Technology	Industrial Engineering	127	60	5 or 8
4	Middle East Tech. Univ. - TU Delft	Computational Design	90	10	2-12

4. A Case Study: Atilim University Software Engineering MS Program with ECTS Credits

Atilim University, Graduate School of Natural and Applied Sciences have been working on the Bologna Process for the MS and PhD programs at the Engineering College since 2009. According to the action plan set by the Graduate School, graduate programs and their ECTS units in Turkey and Europe are investigated and similarities and discrepancies in the programs are determined from different perspectives. Accordingly, the graduate school came up with a uniform program across the university which allows student exchange from other universities without too many complications.

As a case study the MS in Software Engineering at Atilim University is given in Table 3 below. Thesis study is completed in three modules starting from semester 2 and onwards with separate ECTS units (7.5 + 22.5 + 30 = 60 ECTS)). For the non-thesis option, the Graduation Project is completed in two modules (15 + 30 = 45 ECTS).

Table 3. MS in Software Engineering at Atilim University, Turkey

Graduate Program with Thesis (4 semesters - 120 ECTS)
• 7 courses (7 x 7.5 ECTS = 52.5 ECTS) • Graduation Seminar (7,5 ECTS) • Thesis Subject (60 ECTS)
Graduate Program without Thesis (4 semesters – 120 ECTS)
• 10 courses (10 x 7,5 – 75 ECTS) • Graduation Project (45 ECTS)

A more detailed curriculum is provided in Table 4 where the standard credit units as well as the ECTS units are displayed in the last two colums for the thesis option. By the rues and regulations of the University, the thesis work is offered in three modules starting from the second semester. Obviously, 30 ECTS limitation per semester causes some inefficiencies in running thew program.

Table 4. MS in Software Engineering (with Thesis option) at Atilim University, Turkey

Course	Course Type	Crd	ECTS
	Semester 1		
SE	Software Group Course	(3-0)3	7.5
SE	Software Group Course	(3-0)3	7.5
COMPE	Computer Science Group Course	(3-0)3	7.5
	ELECTIVE	(3-0)3	7.5
	Semester 2		
	ELECTIVE	(3-0)3	7.5
	ELECTIVE	(3-0)3	7.5
	ELECTIVE	(3-0)3	7.5
SE 599-1	Thesis	(3-0)3	7.5
	Semester 3		
SE 589	Graduation Seminar	NC[2]	7.5
SE 598	Special Studies on Thesis Subject	NC	10.0
SE 599-2	Thesis	NC	12.5
	Semester 4		
SE 599-3	Thesis	NC	30

5. Remarks

In this article, the transition of the standard credit units in the national MS Programs to the ECTS unit system according to the Bologna Process is discussed. Problems pertaining to the ECTS unit system for the Turkish Universities were summarized and as a case study, the Software Engineering MS Program at Atilim University is investigated. It is assumed that, this program will be a good example for the other universities in Turkey under the current restrictions set by the Council of Higher Education. A more realistic credit distribution is only possible if these restrictions are released from the system to allow better mobility within the European countries and Turkish universities.

[2] NC = Non Credit

Students as Catalyst of the Knowledge Economy: Evolution of Synergies between Companies and Universities

Rainer Groh, Dietrich Kammer, Ingmar S. Franke

Technische Universität Dresden
Faculty of Computer Science
Institute of Software and Multimedia Technology
01062 Dresden, Germany

{rainer.groh, dietrich.kammer, ingmar.franke}@tu-dresden.de

1 Introduction

Universities can play an important role in economic and technological development [Steenhuis et al. 2006]. In this contribution, we demonstrate how companies and universities can immediately cooperate in order to satisfy their respective business and research interests (cp. Figure 1). A major trend in informatics is the starting point of our cooperation.

Multi-touch interaction is both a research topic and a part of business models. Consumer devices like Apple's iPhone and iPad [Apple 2010] demonstrate the capabilities of this direct interaction on a small scale. In contrast, devices like Microsoft® Surface [Microsoft 2010] are larger, offering the possibility to use both hands and allowing users to work in a cooperative, intuitive, and approachable manner. These devices are currently rather expensive for state universities and customers alike. In contrast, private universities have private funds, tuition fees, and other benefactors available. To decrease the price of novel technologies, suitable concepts need to be devised in order to implement applications that exploit their potentials.

To address this challenge, we first describe funding issues, which are relevant to state universities. In the following, we discuss two education models that allow students to catalyze multi-touch topics. We address student enrollment, the necessary infrastructure, and our evaluation of the conducted educational methods.

2 Funding Issues

Demand by companies for specific skills is not part of the schedule of state universities. They need to be answered by innovative, novel education models. Such models consist of teaching methods and appropriate evaluation techniques. Funding is necessary to implement and test these in the form of a pilot project. Educators must be paid and necessary technology has to be provided.

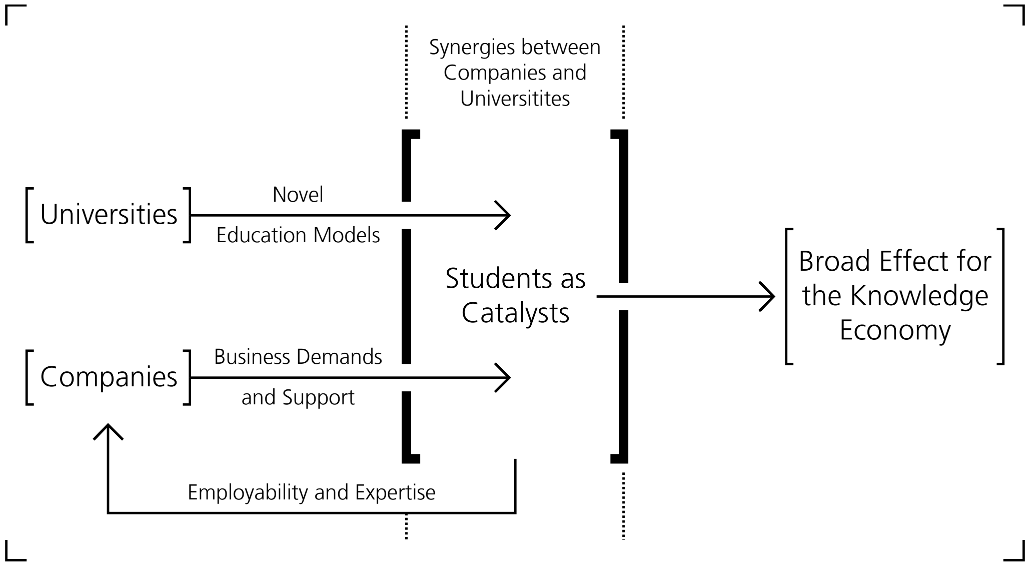

Figure 1: Interplay of companies and universities in the knowledge economy

In our contribution, equipment support of local companies, T-Systems Multimedia Solutions GmbH and queo GmbH, both located in the city of Dresden and members of Silicon Saxony e. V. [Silicon Saxony 2010], was available. Moreover, human competence and resources for teaching are necessary. At this point, the European Social Fund (ESF) can be exploited as a program, which is focused on additional education. The ESF includes all EU regions with a Gross Domestic Product (GDP) per head below 75% of the Community average. The countries and regions eligible under the convergence objective will receive more than 80% of the EU funding [ESF 2010]. Various models are available, which could be used for similar education models as described in this contribution. The ESF is especially relevant for eastern European countries, because funding is spread across the member states and regions, in particular those where economic development is less advanced. As a result, people from these parts of the European Union can setup projects and benefit from them, for example by receiving training in new skills or guidance.

3 Students as Catalyst

In this section, effective methods for student enrollment are discussed. Next, two education models and corresponding evaluation methods are presented.

3.1 Student Enrollment

Screening tests are a common tool at some universities to evaluate applicants. Especially with disciplines that are highly demanded, but offer few places at university. Another method of selecting applicants prior to enrollment at universities is to judge them according to their last degrees. In contrast to student enrollment at the university, later participation in specific activities is often decided on a “first come, first serve” basis. In some cases, electronic submission tools are used [jExam 2010] and luck decides, whether a student can participate in a laboratory or seminar.

To identify suitable students to act as better catalyst, we argue that a different casting process is advisable. Students were attracted to the workshop and laboratory by flyers, posters, and short announcements in lectures. Their application included their student’s curriculum vitae, skills they possess, and an overview of private or business projects. These applications could be overlooked both by business experts from the involved companies and by instructors of the university. Furthermore, technical and conceptual skills could be balanced across the selected participants. As a result, universities effectively receive consulting services from businesses regarding the participants that would be exceedingly expensive without the cooperation described in this contribution (cp. Figure 1).

3.2 Education Models

The first education model was a business oriented *workshop* which lasted two weeks [Franke et al. 2010]. The second constituted a *laboratory* which spanned four months (cp. Figure 2). The workshop was much more focused and had students work full-time each day. In some cases, students used weekends as well, showing their extended motivation and dedication. The workshop's advantages were fast results and clearer guidance by counselors. The net teaching time of 80 hours was identical for both approaches.

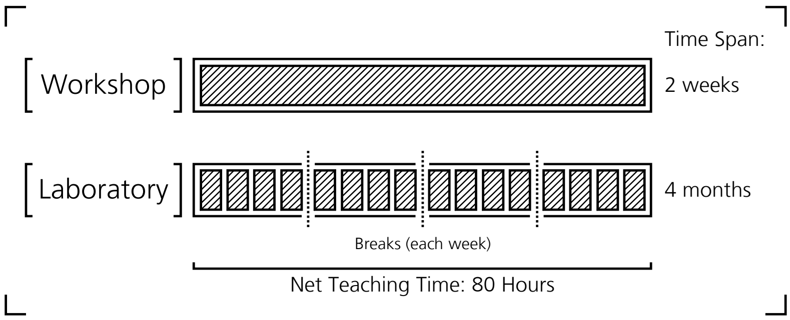

Figure 2: Comparison of education models

The laboratory required students to spend approximately 5 hours of work each week of the four months. This offered more freedom and intermediate time to reflect upon the solutions being realized. The workshop imposed more boundaries on students, but also required more educational efforts to teach students how to use necessary tools. During the laboratory, students had the freedom to choose their development environment and were requested to study necessary tools on their own.

In each case, the necessary infrastructure was established at the university (computer labs), with addition of the necessary multi-touch devices provided by companies. The multi-touch table xdesk [impressx 2010] was provided by queo GmbH and a Microsoft® Surface [Microsoft 2010] by T-Systems Multimedia Solutions GmbH. The Surface was the only multi-touch device used in the workshop. In the laboratory, students were free to choose between either – the majority used the open-source compatible multi-touch table xdesk. Concrete tasks in the workshop evolved from a traditional brainstorming focused on challenges, solutions, and concepts offered by the technology [Osborn 1979] during the opening session. The laboratory acquainted students with open research questions, which were refined to concrete concepts during the first two weeks.

3.3 Evaluation

Evaluations of the presented education models were different. The workshop allowed daily reports over the two weeks to evaluate the progress of each student group (five groups of two students each). The educational results of the laboratory were evaluated by a questionnaire at the end of the four months, showing very varying times needed to adjust to the development of multi-touch applications. As a consequence, times of productive work varied greatly as well.

In each case, a final presentation served to evaluate the implemented prototypes (cp. Figure 3). These final events were also attended by interested persons of the cooperating companies. At this moment, excellent students were able to be recommended as possible new employees. Altogether, students were given a central role in cooperation between companies and university – they became a catalyst of the knowledge economy.

4 Conclusions and Future Work

In our experience, the workshop met the requirements of the involved business partners best. In contrast, laboratories have been proven to be better suited for research oriented problems. However, both education models enable students to serve as catalyst to achieve creative and innovative solutions, which are necessary in the knowledge economy (cp. results in Figure 3). In addition to acquiring new expertise in the field of multi-touch technology, students from both – workshop and laboratory – were hired as student trainees or interns by the involved companies. Their additional employability resulted from evolved synergies between companies and university by exploiting education models specifically set up according to the demands of business, research, and science (cp. Figure 1).

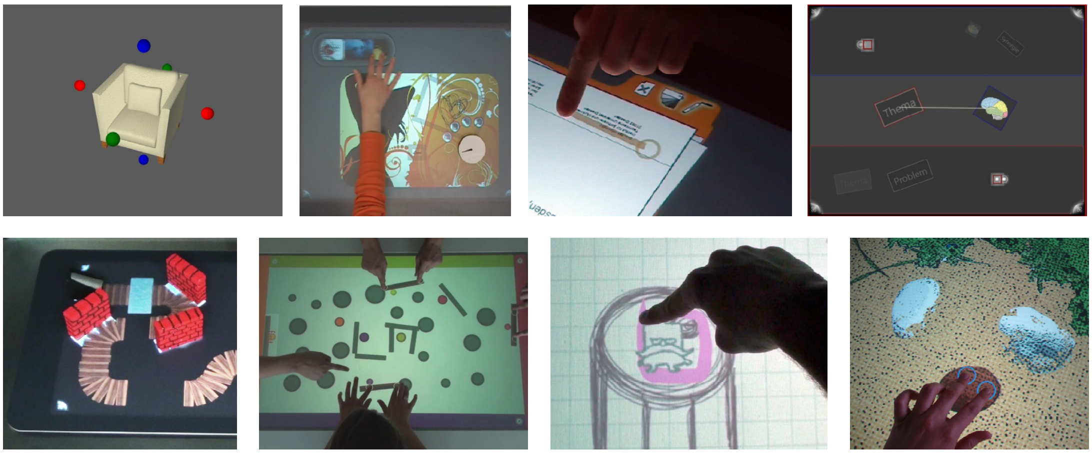

Figure 3: Banner of images from prototypical implementations from the workshop (top row) and laboratory (bottom row)

In the future, continuation of both education models in close cooperation with partners from industry is projected. In addition, benefits from both approaches are to be joined in so-called *boot camps*. Excellent business professionals are invited to pass their experiences on to students. Activities are distributed over three concentrated events, lasting three days each, similar to the workshop. The benefit of this approach is twofold. Invited teachers are accommodated because they can fit boot camps into their busy schedules and students are allowed more intermediate time to reflect on the contents of the lectures and practice on their own, similar to the laboratory.

Acknowledgements

The authors would like to thank the companies involved in the first trials of the workshop (T-Systems Multimedia Solutions GmbH [TMMS 2010]), Dresden) and laboratory (queo GmbH [queo 2010], Dresden). Thanks are due to the numerous students, who acted as catalyst and the Free State Saxony for providing the fundamental infrastructure at Technische Universität Dresden.

References

[Apple 2010] Apple Inc. Apple - iPhone - Mobile phone, iPod, and Internet device. Website. Available at: http://www.apple.com/iphone/. Retrieved: 9/19/2010.

[ESF 2010] European Commission European Social Fund Website. Website. Availabe at: http://ec.europa.eu/employment_social/esf/discover/participate_en.htm. Retrieved: 9/19/2010.

[Franke et al. 2010] Ingmar S. Franke, Dietrich Kammer, Rainer Groh, Simone Happ, Juliane Steinhauf, Frank Schönefeld: Developing Multi-touch Software through Creative Destruction. Entertainment Interfaces Track. Mensch und Computer 2010. Duisburg.

[impressx 2010] xdesk surface computing. Website. Impressx GmbH. Available at: http://impressx.com/ Retrieved: 9/19/2010.

[jExam 2010] Online Submission Tool for Students. Technische Universität Dresden. Available at: http://jexam.inf.tu-dresden.de/. Retrieved: 9/19/2010.

[Microsoft 2010] Microsoft®. Microsoft® Surface. Website. Available at: http://www.microsoft.com/surface/. Retrieved: 9/19/2010.

[Osborn 1979] Alex F. Osborn: Applied Imagination. Scribner, June 1979.

[queo 2010] queo GmbH. Available at: http://www.queo-media.com/. Retrieved: 9/19/2010.

[Silicon Saxony 2010] Silicon Saxony e.V. Website. Available at: http://www.silicon-saxony.com. Retrieved: 7/30/2010

[Steenhuis et al. 2006] Harm-Jan Steenhuis, Denis O. Gray: The university as the engine of growth: an analysis of how universities can contribute to the economy. International Journal of Technology Transfer and Commercialisation, Issue: Volume 5, Number 4 / 2006, Pages: 421 – 432

[TMMS 2010] T-Systems Multimedia Solutions GmbH. Available at: http://www.t-systems-mms.com/. Retrieved: 9/19/2010.

How to Involve Bulgarian Students in Academic Activities: The Experience of LIM

Maria M. Nisheva
Faculty of Mathematics and Informatics, Sofia University “St. Kliment Ohridski”
marian@fmi.uni-sofia.bg

Abstract. The paper presents in brief the aims and the main activities of the Laboratory for Interactive Multimedia at the Faculty of Mathematics and Informatics, Sofia University, and its role in motivating and encouraging undergraduate students for intensive involvement in academic activities.

Keywords: Academic Activities, University Research Fund, Interactive Multimedia, Rich Internet Applications, Rich Digital Content

The aim of this paper is to share the experience gained at the Faculty of Mathematics and Informatics (FMI), Sofia University “St. Kliment Ohridski” (SU), in the motivation and encouragement of undergraduate students for intensive involvement in a variety of academic activities.

During the last 1-2 decades, along with the positive influence of the democratic changes in the country, an alarming occurrence is speeding up in Bulgarian Universities – undergraduate students orientate early to business and draw back from most forms of academic activities, mainly from research activities. The good opportunities for professional growth and payment offered by the software industry strengthen this phenomenon in FMI and motivate its teaching and research staff to look for new forms for attracting the best students in our Bachelor and Master programs with RTDI activities.

An appropriate form in this sense was found to be the new Laboratory for Interactive Multimedia (LIM) at the Department of Computer Informatics of FMI. The Laboratory is an auxiliary section of FMI and includes some permanent staff (7 professors and assistant professors) as well as a number of temporary collaborators (approximately 15-20 FMI students).

The areas of interest for LIM include:

- development of rich Internet applications;
- innovative concepts for design of user interfaces;
- application of interactive technologies in education.

The Laboratory was established with the following main objectives:

- to promote the accumulation and improvement of human potential in the above areas by design and implementation of innovative software and research projects;
- to search out and encourage collaborative activities between University lecturers and students as well as collaboration with other academic and non-academic organizations;
- to carry out practical training of undergraduate students by participation in real projects, technology and problem-solving oriented workshops, etc.

The partners of LIM offer noteworthy payment that enables the Laboratory staff to focus their efforts on the specific academic activities, without the necessity of seeking additional earnings from side-line.

Among the most significant projects developed by (or with the active participation of) staff members of LIM, one may indicate:

- **INTERVISO** – directed to the application of multimedia technologies in teaching and learning Geometry (in partnership with Zuse Institute Berlin and Berlin Free University);
- **Sophie (the Future of the Book)** – an open source software for development and management of multimedia content (in partnership with The University of Southern California);
- **TENCOMPETENCE** – part of the European network for lifelong competence development.

The main activities of LIM during the last year were directed to the implementation of Sophie 2.0 project. Sophie 2.0 is software for writing and reading rich media documents (interactive e-books) in a networked environment. The program emerged from the desire to create an easy-to-use application that would allow users to combine text, images, video, and sound not only quickly and simply but with precision and sophistication. Sophie's goal is to open up the world of multimedia authoring to a wide range of people, institutions, and publishers. In so doing, Sophie redefines the notion of a "book" or academic paper to include both rich media and mechanisms for reader feedback and real time conversation. Its users are interested in creating robust, elegant, networked texts without requiring programming knowledge or training in the use of more complex software tools. Within their work on the project, the participating students gain professional experience in the used new technologies (Java, Flex etc.).

At the same time, the staff of LIM carries out a series of complementary activities within FMI:

- preparation of elective courses at Bachelor's level (Design and Analysis of Computer Algorithms; Advanced Java Technologies; Development of Rich Internet Applications with Adobe Flex);
- installation and maintenance of the E-learning platform Moodle.

The Laboratory offers an annual scholarship for five outstanding FMI students selected by competition. The grants are given by Astea Solutions AD and The Haemimont Foundation under the steerage of LIM.

After the official end of Sophie 2.0 project, the next mission of LIM will be to make an effort to integrate the new software in the activities of several faculties of SU (FMI, Faculty of Economics and Business Administration, Faculty of Philosophy, Faculty of Journalism and Mass Communications etc.).

Along with that, some new partnerships with foreign Universities like the University of Toronto, Cornell University and University of Illinois at Urbana-Champagne are in a process of preparation. These partnerships are aimed at the progress of the ideas, worked out during the implementation of Sophie 2.0.

Recently a part of the staff of LIM is working on a new project named "Methods and Tools Supporting the Lifecycle of Rich Digital Content". This project is supported by the Sofia University SRF. It is aimed at the implementation of research directed to development of a methodology and software tools for creation and publication of rich digital content. Some former results of the research team concerning (1) the development of software tools for writing and reading rich media documents in a networked environment and (2) the development of methods and software tools for creation of academic digital libraries, are expected be integrated and completed.

More precisely, the project has the following objectives:

- to analyze the components of the so-called rich digital content and the requirements to a modern format which would give the best possibilities for presentation of multiform digital content;
- to examine the requirements to an innovative system integrating the entire set of necessary instruments for the support of all phases of the lifecycle of heterogeneous academic digital content (creation, management, browsing/reading, reviewing/annotating, distribution);
- to work out and to experiment an appropriate methodology for application of semantic technologies and tools in management, annotating and searching of rich digital content.

As a result of the preliminary analysis carried out by the research team, four main tasks were formulated in order to achieve the project objectives:

- identification of the main types of authors and users of academic digital libraries with rich digital content and precise definition of their specific roles;
- formulation of requirements to a format for presentation of multiform digital content;
- elaboration of a model of the complete lifecycle of rich digital content;
- advancement and extension of the approach to application of semantic technologies in building academic digital libraries, developed by members of the research team.

The main tasks defined within the project presume the accomplishment of research activities in several directions: analysis of the requirements of authors and users of rich digital content; examination of the formats and standards for presentation of rich digital content; study of the lifecycle of rich digital content; studies in the area of academic digital libraries with rich digital content. More than 10 FMI students (undergraduate students, postgraduate and Ph.D. students) are included in the implementation of all of these tasks. Their efforts have been concentrated in the following activities:

- study of the architecture and the principles of implementation of an open source software tool supporting the lifecycle of rich digital content;
- analysis of the requirements of the main types of users of academic digital libraries and application of the corresponding results in the design and implementation of appropriate authoring tools for a particular academic digital library being under development at FMI. These tools are oriented to the teaching and research staff members and the library administrators who may create and edit catalogue descriptions and upload library resources in the proper repositories;
- implementation of a specific methodology for application of semantic technologies and tools in the creation, maintenance, annotation and search of rich digital content.

As a result of the successful collaboration between University professors, assistant professors and FMI students, seven teamwork research papers were presented at three international conferences (hold in Helsinki, Belgrade and Sofia respectively) during the first six months of 2010. Three of these presentations were successfully realized by our students. In this way they gained useful experience, established new professional contacts and found an opportunity to match their achievements with the best European practices.

The further plans of LIM include some activities directed to the improvement of Sophie 2.0 and its integration with other special-purpose software tools used in FMI and other Faculties of SU. The work within the “Methods and Tools Supporting the Lifecycle of Rich Digital Content” project will continue with a stress on the formats for presentation of rich digital content and the semantic technologies issues. The specific learning and teaching activities will be concentrated on the preparation of a new elective course on Development of Mobile Applications for OS Android and the elaboration of the “old” course on Development of Rich Internet Applications with Adobe Flex.

Some ideas for collaboration with the newly established Centre for simulation, business processes and 3D visualization at FMI have been discussed as well.

International Dimensions in Informatics Education at CTU

Josef Kolář, Faculty of Information Technology
Czech Technical University in Prague

Introduction

Specific strategic goals in the domain of internationalisation of the Czech Technical University in Prague (CTU) Long-Term Intention for Years 2006 – 2010 include the ambition of raising the number of international students in our study programs from approx. 1200 (in 2005) to some 2500 or 3000 in 2010 thus reaching the proportion 12% – 14% of international students in the whole CTU student population. Although the end of year 2010 is fast approaching, achieving this and other related goals is still more a dream than reality.

We briefly discuss the issue of internationalisation from the Informatics Education perspective since this domain has always been most popular among incoming mobility and full-degree students and we believe the newly established Faculty of Information Technology (FIT) will confirm this tradition.

Reasons for Going International

Nowadays, universities that aim at being recognized as important research and educational institutions cannot limit their activities by regional or national borders. International cooperation of researchers, exchange stays of academic staff and mobility of students have become a necessity, so that it is not existence but quality and intensity of these activities that differentiate the best ones from the average. Internationally renowned universities more likely succeed in competition for research grants, excellent staff and quality seeking students. In short, internationalisation is one of the most important conditions to achieve excellence.

The process of internationalisation concerns all university activities and has so many forms and aspects that its comprehensive analysis cannot be done in this text. Instead, as indicated in the introduction, we concentrate on how international aspects have been perceived while preparing and carrying out educational programs in Informatics within the CTU context. One year of existence of the Faculty of Information Technology at CTU could hardly offer enough experience to draw any conclusions if we could not count with more than 45 years of presence in ICT education market while forming the team of the Department of Computer Science and Engineering (Faculty of Electrical Engineering, CTU).

As compared to the rest of the faculty, the department has been exposed to a relatively high presence of international students in our programs (both as mobility and full-degree students) for most of its existence. Of course, conditions for international cooperation changed in a decisive way after the political changes in 1989 that removed former political barriers. Yet there was still much to do before CTU could make use of the new conditions, especially regarding its attractiveness to international students.

Former fixed study plans were replaced with a credit system that gave students more freedom and responsibility to select and define the contents of their study plans. In early 1990's, CTU English study programmes did not exist yet, but there were Ph.D. students coming for a complete degree programme being tutored on an individual basis by English-speaking staff. A typical Ph.D. student was a holder of governmental scholarship coming from some Arabic or other developing country. At that time, the average level of fluency in English of CTU's academic staff was hardly reaching that of arriving students.

Czech Dimension

Historically, former Czechoslovakia used to have three German universities, but they were dissolved by a presidential decree just after the World War II in 1945. Since then, Czech and Slovak were used as the only teaching languages at Czechoslovak universities. At that time, international students coming to enrol full-degree programs at some local university were supposed to take a preparatory one-year course organised by a specialized institution. Apart from learning the language (either Czech or Slovak), the students were taking some supplementary courses according to their future university study programme.

After the split of the country into Czechia and Slovakia in 1993, only Czech remained as an exclusive teaching language although Slovak is commonly not considered a "foreign" language and Slovak citizens have got some legal advantages as compared to the rest. As a result of long-time isolation from western countries and reluctance to follow the official strong orientation to Soviet Union and its satellite countries, generations of Czech youngsters have grown up without need and wish to master any foreign language and to travel abroad.

The situation among university teachers and scientists was not remarkably better, even though passive knowledge of some foreign language (especially English in the domain of Informatics) was common and needed for their work. Very often, their attempts to cooperate with colleagues in the West or get a study or working stay at a university abroad were considered undesirable and could even have a negative impact on their professional career. Now it seems to be a story from other world, but a scientist who was happy enough to participate in a conference abroad was obliged to report in written any "contact with a foreigner" that he had during his trip to special office that existed at every faculty.

Information resources as books and journals concerning Informatics were accessible since early 1960's to some economically feasible extent, so that we could follow the initiatives in computer science curricula (such as [1], [2], etc.) being developed by the Association for Computing Machinery (ACM) and other professional bodies. We were trying to adhere to or at least to learn from these recommendations in spite of the fact that a strictly centralized governmental control of that time dictated the same structure and contents of university study programmes in Informatics at all technical universities across the country.

The fall down of the communist regime finally opened the door, so that to travel abroad and participate in international projects has fast become a commonplace. With academic freedom and access to information resources we thought that to define our research domain and develop Informatics study programme according to most recent principles (as presented e.g. in [3], [4], [5], etc.) is just at hand. Although we were able to offer an up-to-date study programme, it did not match the frame set up by the Faculty of Electrical Engineering that was still considering Informatics (Computer Science) a part of electrical engineering.

This was in brief our situation at the time when CTU as well as other Czech universities started to participate in exchange student mobility programs in mid 1990's what has lead to opening of English courses or even English full-degree programs. Our offer of Informatics courses was attractive for mobility students since they could take them without the electrical engineering and physics courses that were compulsory in the full-degree programme.

As compared to universities located in anglophone countries, opening English courses at CTU has always been painful from a number of reasons. At the beginning the pain was mostly due to the lack of language abilities of academic staff, nowadays, the economic factors prevail. Teaching relatively small groups in English obviously does not pay off even though a relatively high tuition fee is applied to full-degree students, since there are only a few of them. There are two possible solutions to this problem – either to have large enough groups of international students or to deliver – at least some – courses of the Czech program in English. The best option, perhaps, is to do it both ways.

Western Dimension

Students coming from USA or West-European countries are mostly bilateral exchange or Erasmus mobility students. Their numbers have been steadily growing what led to improvements in accessibility of necessary information and in effectiveness of the related administrative processes that are now supported by easy-to-access web applications. Even though, there is still very much to do in both the completeness and structuring of the web pages that are supposed to give info and necessary guidance either to prospective full-degree program students or to incoming mobility students. In my role of a contact person for a number of mobility agreements, I frequently spend long minutes trying to dig out answers to students' queries while browsing the respective CTU pages, which means the needed info is in fact inaccessible for a foreigner.

In spite of some persisting deficiencies, the actual care given to international students at CTU is highly appreciated by the students themselves. It has become a part of standardized process in which all parts are gradually improving. CTU's International Student Club (ISC) has developed into a very efficient counterpart of the Department of International Relations of the CTU Rectorate. Incoming students get their ISC "buddies" that help them at arrival and in social adaptation. Orientation week organized just before the winter semester helps international students get fast all necessary info regarding administrative and study issues as well social and cultural life at CTU and Prague in general.

ISC is run by CTU students themselves, which is considered to give a best preparation for possible mobility abroad. The interest and ability of local CTU students to spend some part of their study at some foreign university has been growing remarkably during the last 20 years, but has not become a commonplace yet for a number of reasons. It still concerns the best (or most active) 10% while the rest are reluctant to invest extra efforts in language preparation and social adaptation. On the other hand, although not supported by any official statistics, there are visible signs that seeking a complete university education abroad is gaining a growing popularity among Czech young people leaving high school.

Eastern Dimension

Due to the absence of language barrier, Slovak students can easily enrol the Czech study program, so that they represent the largest group of international students both at FIT and CTU in general. In the admission procedure, candidates from Slovakia are not asked to submit legal translation or nostrification of their respective documents, which is the case of candidates from other countries. Considering the student population of FIT, we find as rather surprising a high number of enrolled students from Russia, Ukraine, Belorussia and even from some Asian republics of the former Soviet Union. Again, they enrol our Czech program since it implies no tuition fees, which is for them a sufficient motivation to start learning Czech "on the fly". Of course, some of them underestimate the problems they could have with the language and drop out fast.

For candidates from these countries, the study at CTU means "going west" while preserving reasonable cost of living and possibly earning a recognised diploma. When considering this dimension from CTU perspective, the conclusion is to concentrate on excellence. This group of international students will probably include a higher proportion of active and gifted students who are interested in full-degree programme in informatics, take seriously their duties and wish to cooperate in research. Another important outcome of this dimension is that higher enrolment of these students has an important publicity effect and thus it can raise the number of students that can afford studying our English program (and paying tuition fee).

The author is convinced that there is a lot talent in mathematics and informatics in this group of countries – the history results of the ACM International Collegiate Programming Contest (see [6]) are more than convincing. Therefore, CTU should concentrate on bringing this talent to its classrooms, not important whether at the Faculty of Information Technology or other.

Southern and Far Eastern Dimensions

Africa, Latin America, India and countries of East Asia and Pacific Region represent another extremely vast and heterogeneous yet promising target for quality students recruiting. We were surprised that more than half of applications of self-paying students of the Bc and MSc Informatics programs have come from Africa. We have even got some offers of local African and Indian agencies willing to help us with recruiting.

We realize the potential of the continent but at the same time we feel a serious lack of experience and information, specifically concerning the education system in different countries, level of confidence of potential partners and prospective students. We are well aware of the risk that instead of quality study programs, applicants can just be seeking a way to get visa that would enable them to move to Europe.

With respect to reliability and information, as compared to Africa and Far East, Latin American countries in general do not represent such a *terra incognita* to us. CTU has a long tradition in cooperation with Latin America and FIT has already established a number of useful personal links in this region. We actively participate in projects of the Magalhães Network that brings together 16 European and 15 Latin American universities that are evenly spread across both regions.

The network has been very successful with its SMILE program that helped establishing bilateral student and staff exchange agreements between participating institutions. Another example of recent activities of this network is the EU Brazil StartUp project supported by a grant from the Erasmus Mundus External Cooperation Window program of the European Commission. This project will support more than 170 mobilities from Brazil to Europe and vice versa covering transport costs and scholarships.

Although Australia has become the main player in the educational market of India, East Asia and Pacific Region, the demand in this region for quality university education is still growing fast enough to make CTU interested. It is worth mentioning that Australian universities have turned international students' education into one of its most important economic resources. The growing demand for university education abroad in countries like India, China, Vietnam and others comes along with the economic development of this region that has seemingly not been that influenced by the ongoing economic crisis.

Conclusion

Even though the goal of reaching some 12% of international students by the end of 2010 is definitely not realistic for CTU as a whole, the Faculty of Information Technology has a pragmatic yet ambitious goal to get 10% in 10 years. We see a big potential in the countries of Far East and Latin America as far as full-degree Informatics students are concerned and we believe to get many new European partner universities for short-time students exchange.

References:

[1] Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science. Communications of the ACM, Volume 11, Issue 3 (March 1968), pp. 151 – 197, ISSN:0001-0782.

[2] Curriculum '78: recommendations for the undergraduate program in computer science - a report of the ACM curriculum committee on computer science. Communications of the ACM, Volume 22, Issue 3 (March 1979), pp. 147 – 166, ISSN:0001-0782.

[3] Denning, P. J. - Comer, D. E. - Gries, D. - Mulder, M. C. - Tucker, A. - Turner, A. J. - Young, P. R.: Computing as a Discipline. Computer, Volume 22, No. 2 (February 1989), pp. 63-70, doi:10.1109/2.19833.

[4] Computing Curricula 1991. Communications of the ACM, Volume 34, Issue 6 (June 1991), pp. 68 – 84, ISSN:0001-0782.

[5] Computing Curricula 2001 (draft). The Joint Task Force on Computing Curricula, IEEE CS and ACM, March 2000.

[6] The ACM-ICPC International Collegiate Programming Contest Web Site sponsored by IBM (http://cm.baylor.edu/welcome.icpc)

ResEval: A Mashup Platform for Research Evaluation

Muhammad Imran, Florian Daniel, Fabio Casati, Maurizio Marchese

Dipartimento di Ingegneria e Scienza dell'Informazione
University of Trento
{imran, daniel, casati, marchese}@disi.unitn.it

Abstract. Bibliometrics has changed out the way the research evaluation conducted, and it is widely used to evaluate research groups, individual research's, department and many more. However establishing fair criteria to evaluate the scientific community, as well as individual publications and researcher, is a tough task and constitutes a challenge that has not been achieved yet. This paper addresses the problem of research evaluation and introduces ResEval, a mashup platform that enables the creation of customize metrics and their computation in order to make the scientific evaluation easier. This platform addresses various problems with current approaches such as data incompleteness, flexibility in defining new metrics, fixed UI restrictions for the customization of metrics and to apply filters.

Keywords: mashups, research evaluation, bibliometrics

1 Introduction

The evaluation of researchers and research artifacts (e.g., a scientific paper) is important for a variety of reasons, ranging from hiring and promotion (for people)[1] to selection of contributions (for journals or conferences) and to searching for interesting content within an ocean of scientific knowledge. However, there is little consensus today on how research evaluation should be done, and it is commonly acknowledged that the quantitative metrics available today are largely unsatisfactory. Indeed, today people evaluate research contributions mainly through publication in venues of interest and through citation-based metrics (such as the h-index), which attempt to measure research impact. While these metrics are typically considered to be better than nothing, currently significant compilations of research indicators heavily rely on contributions and citations and some other sophisticated techniques. These techniques mainly rely on citation analysis which is used to evaluate research performance of individual researcher as well as research groups. However there are different opinions on how citation statistics would be used, and they have well-known flaws. For instance[3] pointed out shortcomings, biases, and limitations of citation analysis.

In ECSS last year and in other papers [2] we showed that peer review (todays gatekeeper for publications) is essentially applicable only to filter clearly bad papers, and many studies show that (i) citations are to a very large extent affected by the effort made by the authors to promote the paper, and (ii) the cited papers are rarely read by the citing authors [4, 5]. Furthermore, current metrics are limited to papers as the unit of scientific knowledge, while today there are many other artifacts that do contribute to science, such as blogs, datasets, experiments, or even reviews, but that are not considered for research evaluation. In this paper we describe our ideas, preliminary results, and ongoing work on a mashup platform for research evaluation.

Besides the flaws of current metrics, the fact remains that people have - and we believe will always have - different opinions on which criteria are more effective than others, also depending on the task at hand (that is, the reason why they are conducting the evaluation). For example, in our department, the evaluation criteria for researchers are defined in a detailed document of 10 pages full of formulas and are mostly based on publications in venues considered important in the specific community and normalized following some shared and agreed criteria. Other institutions use citation counts normalized by the community to which the authors belong and then grouped by research programs to assess each research group, not individuals. Examples are countless and, much like in the soccer world cup, everybody has an opinion on how it should be done. Not only people may select different metrics, but also different sources (e.g., Google Scholar[1] vs. Scopus[2]), different normalization criteria (e.g., normalizing the value of metrics with respect to averages in a given community), different ways to measure individual contributions (e.g., dividing metrics by number of authors), or different ways to compare (e.g., compare a candidate with the group that wants to hire them to assess the autonomy and diversity of the candidate from the group), with different aggregation functions (e.g., aggregated h-index of a scientists co-authors, aggregated citation count, etc.).

For example, in order to assess the independence of young researchers, we typically look at which contributions the researcher has published without the cooperation of his/her PhD supervisor, and whether he/she was able to publish on new topics independently. As a more detailed analysis, we might, for instance, want to measure the h-index of the researcher without considering papers co-authored by the supervisor, and then compare the cleaned h-index with that of other young researchers in the same field. In order to evaluate the supervisor, instead, we might want to compute his/her impact on the h-index of all his/her PhD students, that is, the aggregated h-index values of all papers co-authored with the supervised PhD students. Of course, the sum of h-index values is not an h-index again, but we can still use the result as a measure of the supervisors capability to (i) attract PhD students (in terms of quantity) and (ii) produce high-quality PhDs.

These observations and examples suggested us the following approach to address the problem of research evaluation and provide a platform for it:

[1] http://scholar.google.com

[2] http://www.scopus.com/home.url

1. We need to provide a platform that allows the computation of a variety of scientific metrics from a variety of sources.
2. We believe that people will want to define their own metrics and these metrics can be very complex. Therefore, we need to enable each person to define the metric computation logic, including the specification of the logic, the sources, the normalization, the aggregation criteria, the comparisons desired and the like.
3. In addition to the above, while the platform above should allow one to include publication and citations as basic metrics and operators, we need a way to provide the means to have alternative metrics that better reflect the opinion and reputation of scientist and contributions with respect to what peer review and citation count do.

In the following paper we focus on the first two items. For the third aspect (which is orthogonal to the issue of defining a platform for research evaluation as it refers to having another base metric in addition to the traditional ones), we refer the reader to our work on liquid journals described at liquidjoural.org.

2 Reseval Concepts and Model

To support research evaluation, we propose a domain-specific information mashup language and platform through which users (possibly with no programming skills) can define, execute, and visualize the metric combination logic and its result.

We provide a mashup platform, because we expect users to need to combine both UI components (widgets that can show charts and trends) and information from a variety of (web) sources. We aim for a domain-specific language because we believe the only way in which we can have a mashup platform that can be used by non-programmers is to constrain the concepts and possibly even the functionality of the platform for the sake of ease of use. Finally we talk about information mashup as we see the platform not so much as a generic composition tool (a la BPEL) but more like a pipe that channels, filters, cleans, aggregates, and transforms information flowing from sources (e.g., repositories of publications such as DBLP[3] or Springer[4], or Liquid Journals[5]) to obtain indicators corresponding to the desired scientific metrics.

We now describe the main elements of the language and of the platform. ResEval provides to its users a mashup model that consists of the following entities:

– Data sources (and an associated model, which is a subset of the ResEval model) that provide the appropriate input to the specific metric computation (i.e., publication metadata, like authors, affiliations, title, citation, etc.);

[3] http://www.informatik.uni-trier.de/∼ley/db/
[4] http://www.springer.com/
[5] http://liquidjournal.org/

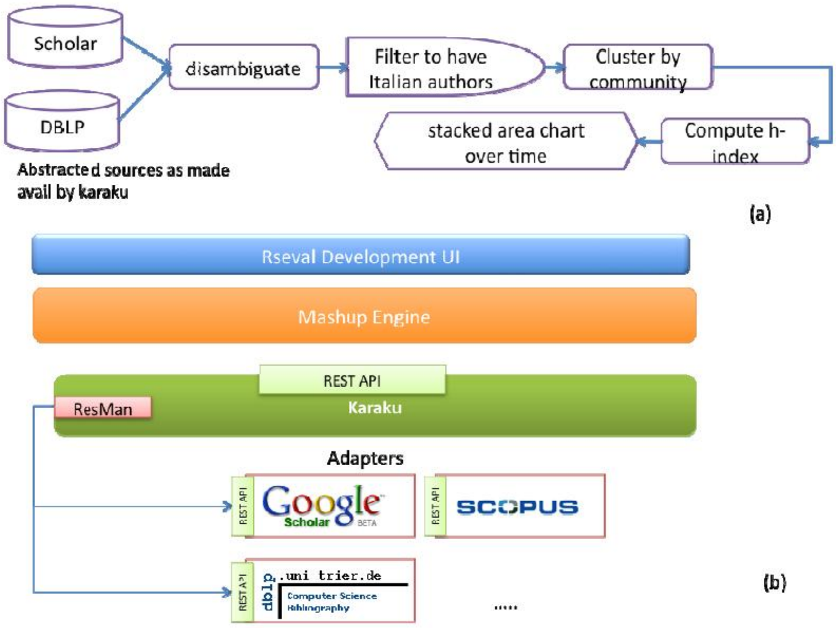

Fig. 1: ResEval (a) mashup modeling example and (b) architecture

- Operators that allows the extraction, elaboration and aggregation of the information as well as the definition of specific and personalized metrics;
- A set of filters that support the transformations of the information; UI components that support the interaction with the final users and the visualization of the computed metric;
- Flow pipes that allow the user to interactively define the flow and transformation of the selected information from the data sources;
- A conceptual model that fixes the main concepts and operators used in the platform. This is part of the essence of the mashup being domain-specific, as discussed later. The model is materialized via an ER diagram (processed by ResEval components) and XML schemas that define the structure of the information flowing through the pipes.

Figure 1 shows a simple example where sources, operators, filters, and UI components are used to compute a metric, Figure 1(b) a first draft of the ResEval architecture.

The data source components constitute the bridge between the ResEval domain-specific world and the Web at large. Because all information on scientific dissemination and the reputation is available on the Web (but not always as a service with an API), we need a way for ResEval to access this information. This implies having a layer that exposes Web sources as services, and in particular as data sources available to ResEval users. This is a complex endeavor because Web sources keep changing, they are intrinsically heterogeneous and not all of them expose standard software interfaces on the Web (such as SOAP or RESTful services). To bridge this gap we developed Karaku[6] and its engine Resman[7], a tool that facilitates exposing web applications as services with a stable API and that also provides caching and crawling of scientific entities (e.g., papers as contributions, journals as collections) to relieve the users of the sources from these tedious tasks. Furthermore, Karaku allows exposing these sources in accordance with the conceptual model of ResEval.

Incidentally this implies that the effort of adding new sources is dependent on programmers to write a wrapper for each source in order to provide services to upper layers. We also assume that expert developers will provide a library of components (such as filters, operators) and make them available to the final users. Final users will instead develop the mashup flows (the metric computation logic).

3 Status

The current public version of ResEval (http://reseval.org) does not yet make available the complete mashup platform. For the time being it allows to compute a predefined set of metrics on top of Karaku, and it allows for creation and comparison of groups of researchers with these metrics. It provides above mentioned exiting metrics and it is continuously increasing its performance and coverage in the sense of data sources. ResEval is a tool for evaluating research contributions and researcher by using citation based indicators. These indicators include popular citation based metrics such as h-index, g-index, noise ratio, citation count. More importantly, this tool allows one to separate citations from self-citation, that is citations coming from the author itself. Moreover, ResEval provides other functionalities, as the possibility to find top co-authors and top citers of a researcher. In our current work we will progressively begin to release on reseval.org both the initial mashup platform as well as functionality to combine information from other sources to define new metrics.

References

1. Bertrand Meyer, Christine Choppy, Jrgen Staunstrup, Jan van Leeuwen. Research evaluation for computer science, 2008.
2. F. Casati, M. Marchese, K. Mirylenka, and A. Ragone. Reviewing peer review: a quantitative analysis of peer review. http://eprints.biblio.unitn.it/archive/00001813/, Feb. 2010.
3. A. Chapman. Assessing research: citation count shortcomings. *The Psychologist*, 2:336–44, 1989.
4. A. Haque and P. Ginsparg. Positional effects on citation and readership in arXiv. *Journal of the American Society for Information Science and Technology*, 60(11):2203–2218, 2009.
5. M. Simkin and V. Roychowdhury. Stochastic modeling of citation slips. *Scientometrics*, 62(3):367–384, 2005.

[6] http://project.liquidpub.org/karaku/

[7] http://project.liquidpub.org/resman/

Why is wrong to evaluate researchers in CS through journals and what can we do about it ?

Ricardo Jiménez-Peris and Marta Patiño-Martínez

Universidad Politécnica de Madrid

Abstract

There is a trend in many European countries to evaluate research in Computer Science (CS) and its impact only using journals. This also implies to use the Thomson Reuters Journal Citation Report (JCR) as ranking mechanism to weight and compare journal publications. In CS the venue for publishing results in most areas are mainly conferences. This is translated in the most relevant results and with most impact being published at top conferences. JCR has several serious drawbacks for evaluating research in CS. First, it only indexes a fraction of the research in CS and a small fraction of the most relevant research. Second, the impact factor shown is very small compared to the real one due to the miss of all citations made by conference publications. Third, the impact factors of journals are unrealistic due to they stem from a low fraction of citations what results in ranking very differently the journals from their real impact. Good journals attain good rankings in general, but medium and bad journals may be ranked better than top journals. Fourth, the JCR publishes the top 50% percentile of indexed journals. With such a high percentile it reports JCR journals with no impact at all (e.g. 0.0000). By using the wrong tool to evaluate research we can simply ruin the next generation of CS researchers that will end up pursuing to publish in journals that nobody will read and cite, simply because they are indexed in JCR, instead of doing relevant and competitive research that can be published in top venues. The main issue with conferences is the lack of a ranking tool like the JCR. Previous efforts such as Citeseer have failed. The panorama has drastically changed with the evolution of Microsoft Academic Search that is providing a tool with a high level of quality and currently indexing over 2.5 million CS papers.

In this paper, we discuss a set of recent tools such as Microsoft Academic Search and a number of metrics that can be used to evaluate CS research and its impact. The provided metrics enable not only to evaluate research and its impact quantitatively through citation analysis, but also qualitatively.

1. Introduction

The evaluation of research and its impact is an instrument to select and promote personnel, give research grants and evaluate the results of research projects. This means that the quality of the process is of crucial importance. There is a trend in many European countries to evaluate research in Computer Science (CS) using journals. This trend is promoted by older sciences such as Physics, Biology, Medicine, etc. For these sciences, journals are the main means of publication and conferences are mainly networking events so publishing in them has little value. For these sciences it makes a lot of sense to rely on the Thomson Reuters Journal

Citation Report (JCR) as ranking mechanism to weight and compare publications, and measure the impact of journals, researchers and organizations.

However, applying the same methodology to CS does not make sense anymore. The reason is that for most research areas in CS the main venue for publishing results are top conferences. When using the JCR for evaluating CS research many problems arise. The first one is that since it only indexes journals this only represents a small fraction of CS publications. Just to quantify this fact, Microsoft Academic Search indexes (as of April 23rd, 2010) 2596 conferences and 706 journals. In terms of papers, indexed conferences have published 50% more papers than indexed journals, or in other words journal papers represent only 1/3 of the overall number of papers published in both journals and conferences. This results in having unrealistic impact factors and giving the appearance that other sciences have venues with higher impacts than CS. What is worse, this results in that journal impacts computed by the JCR and based solely on journal citations can be totally wrong since they are missing 2/3 of all published papers and their citations.

In terms of impact, just looking at the 500 top venues (corresponding roughly to percentile 15%) of the combined ranking of journal and conferences ordered by impact taken from Microsoft Academic Search, it turns out that there are 300% more conferences than journals, or more precisely, in the top 500 venues, there are 350 conferences but only 150 journals. That is, journals account only for 30% of the 500 top venues. This signifies that looking only at journals the evaluation of the publications of a researcher can be totally biased. What is more looking only at journals one is missing 70% of the most relevant research.

Another important problem with JCR is that it only tracks citations from journals and does so only in windows of 2 years (recently also in 5-year windows). Tracking only citations from journals misses the citations from a large fraction of the most relevant research, 70% according the above figures, what might result in a ranking very far from reality. In fact, one easy exercise is to look for the ranking of top ranked journals such as traditional ACM transactions: TOCS, TODS, TOIS, TOPLAS, TOSEM, JACM, TOCHI, and TOG. In JCR they are ranked in 2009 in the positions over all the CS categories: 51st, 198th, 132nd, 281st, 92nd, 42nd, 216th, 12th, and. This is certainly very far from the real impact of these journals. Each of them is considered the top journal without discussion in their research areas. When considering both citations from conferences and journals, it turns out that they are ranked 1st, 6th, 7th, 8th, 13th, 14th, 18th, and 23rd. That is, they are ranked within the 25 top journals what corresponds to the notion the researchers in their areas have. However, in JCR only one appears among the top 25. It should be noted that the data from which the ranks are obtained in Microsoft Academic Search are really massive. For instance, for TOCS, 463 articles are indexed (TOCS publishes around 12-20 papers per year over roughly 30 years) and 29,713 citations have been tracked. However, JCR simply looks at 20 articles over the 2-year period 2008-2009.

Given all the above considerations, adopting journal publications and the JCR to evaluate research and its impact in CS does not make much sense. Unfortunately, this trend is even being adopted by growing sets of researchers due to it is being employed by several funding agencies to evaluate CVs and project results. Certainly, this is not a problem for senior researchers for which to publish a few papers per year with relevance in JCR but no real impact is not an issue (publishing papers in journals with high real impact like ACM transactions is not that easy), although it can be a big waste of time. The main problem comes for the new generation of scientists that instead of publishing in top conferences and journals that have a lot of difficulty due to the high competition with the best researchers might start to pursue publications in journals well ranked in JCR but where it is much easier to publish due to the real reputation and impact of the journals is much lower. This can result in ruining a large fraction of the next generation of researchers in many countries that is the real danger.

The question raised is therefore which tools can be used to evaluate the publications of researchers in both conferences and journals, that is, the impact of the venues and of the publications. Citeseer was a very welcomed effort, but unfortunately it encountered serious problems that have not been successfully solved despite its evolution into Citeseer-X. The main issue is that the number of indexed papers has not grown significantly since 2003 that remains at a modest number of 1.6 million papers (Citeseer reached 1.3 million indexed papers). Fortunately, the panorama has drastically changed with the upcoming of Microsoft Academic Search [AcademicSearch] launched some years ago that has reached a good level of maturity and it is currently indexing 2.5 CS conference and journal papers and providing high quality citation information.

In this paper, we discuss a number of tools available to measure the impact of venues, publications and researchers, as well as a set of metrics to perform the measurements with some advices on how to use and combine the tools. The metrics focus not only in measuring quantity, but also made an effort in measuring quality. We hope that the provided information can be useful to improve the evaluation of research and its impact by means of quantitative and qualitative citation analysis. In the discussion and conclusions Section we also bring some thoughts on other directions that the CS community can take to solve the issue of research evaluation in CS that can be taken at political level.

2. Tools to evaluate research impact

The main tool that has been inherited from other research domains such as Physics, Biology, Medicine, etc. to rank venues and publications is the Thomson Reuters Journal Citation Report (JCR) [WoS]. In such domains in which the main publication means are journals, this makes a lot of sense. Conferences in these domains are mainly networking events and the publications in them have no value. However, in computer science in most of its research areas the main publication venues are top conferences. We will provide later quantitative data about this claim. The main issue is whether there is an available tool with a quality similar to JCR to rank conferences. Another important matter is whether the JCR provides a good ranking for journals or it is too biased due to it only tracks citations from journals.

Some well known tools are Citeseer [Citeseer] and its evolution Citeseer-X [Citeseer]. They analyze citations from documents available from Internet and rank the impact of papers, authors and venues (both conferences and journals). Google Scholar is also well-known. They analyze citations of papers, but allow performing searches of papers from a particular author. Publish-or-Perish is an interface to Google Scholar that enables to count all citations to papers from a particular author. The interface enables to uncheck incorrect citations, and perform relatively complex queries to obtain an accurate gross citation number and the resulting citation indexes such as g-index and h-index.

Less known are other tools. Scopus [Scopus] is a tool from Elsevier that provides a citation analysis of papers from both journals and conferences. It aims at competing with JCR. As JCR it is an expensive tool that only allows institutional subscriptions. The ACM Digital Library [ACMDL] tracks citations of ACM publications in journal and conferences and from some other venues. It provides a full citation tracking of authors although it only tracks ACM publications plus a few others.

A relatively recent tool, still not widely-known is Microsoft Academic Search. This tool indexes papers from journals and conferences, and it also provides rankings for authors, papers, venues and organizations. Its quality is quite high and the number of indexed CS papers is 2.5 million. According to our analysis it is currently the best and most complete free tool for ranking both conferences and journals. Ranking of venues is pretty good and stable. Some data cleaning is needed before using them, but this can be performed very easily. There are a few venues for which a very low number of papers appear as indexed but with a too high number of citations. This data can be easily cleaned up by removing venues with a small number of publications (e.g. less than 200 publications). The only other issue, which is shared with JCR, is that some series like LNCS and ACM SIG (special interest groups) newsletters are indexed that mix different kinds of publications. The issue with LNCS is that is just a mix of the proceedings of many conferences, compiled books and books. In LNCS one can find proceedings of top conferences like ECOOP and proceedings of national meetings without even peer review. Since conferences are already indexed by themselves LNCS can be simply removed from the ranking. Similarly happens with ACM SIG letters, since they contain the proceedings organized by the ACM SIG group they may have very high impacts. However, the letters are a mix of the proceedings and the letters. The former has a peer review process and typically have good to excellent impacts, whilst the latter do not have a peer review process and have low impacts. Since, again, these conferences are already ranked in the conference ranking, one can simply remove the letters from the journal ranking.

Another recent tool developed by Lisbon University is CIDS [CIDS] that it is an interface to Google Scholar that removes self-citations from all co-authors. DBLP is a well-known tool that provides the best digital library of CS publications although it does not perform any citation analysis. There are some human-made rankings that have become relatively known such as CORE [CORE] performed by Australian researchers.

In the following table we summarize the comparison and analysis of tools:

Tool\Features	**Cost**	**Indexed CS venues**	**CS coverage**	**Rankings Provided**	**Recommended for**
JCR [JCR]	Very high	Journals	Yes	journals, researchers & organizations	
Scopus [Scopus]	Very high	Journals and Conferences	Yes	journals, conferences, researchers	

				& organizations	
ACM Digital Library [ACMDL]	Affordable for Individual Researchers	Journal and Conferences	Yes	No	
Google Scholar [Scholar] + Publish or Perish [PoP] + CIDS [CIDS]	Free	Journal and conferences	Yes	No	Researcher Citation Analysis
Microsoft Academic Search [AcademicSearch]	Free	Journal and conferences	Yes (2.5 million papers)	journals, conferences, researchers & organizations	Journal, conference, top-1000 researcher and organization rankings
Citeseer-X [Citeseer-X]	Free	Journal and conferences	Yes (1.6 million papers)	journal, conferences and researchers	
CORE [CORE]	Free	Journal and conferences	Yes	journal and conferences	

3. Metrics

There are two related families of metrics to evaluate research: publications and the impact of the publications. Publications can be measured by quantitative and qualitative metrics. Quantitative metrics measure the gross number of publications, whilst qualitative metrics provide a measure of the quality of the venue and its impact.

The metrics related to the publication venue are the acceptance rate and the reputation/impact of the venue. The acceptance rate can give an idea of the difficulty to publish, however, by itself is meaningless. The acceptance rate can provide complementary information to the venue given a ranking of the venue. This enables to compare the relative competitiveness to publish in different areas. For ranking the venue there are two methods: human ranking and automatic ranking. The first one can be made by organizations, as it is the case for CORE. However, it can always be biased by the membership. Additionally, the categories are too broad. The current top category in CORE is A and corresponds to the top-30% venues. This strip includes from top conferences to 3-4th tier conferences with very low impacts. Automatic ranking can be attained by the impact of the venues through citation analysis. The absolute impact has little meaning since it depends on many factors, such as the total amount of indexed papers. A good means to make the impact relative is to use percentiles for citation ranking.

The impact of the publications can be measured by their citations. Again, the metrics can be quantitative or qualitative. The quantitative metrics is the gross number of citations and the number of citations without citations from any of the co-authors. Evidently, it is the latter the only one that makes sense, however, there are many tools only offer the former and requires to remove the self-citations manually. However, as with publications quantitative measures are not good enough and require to be complemented with qualitative measures.

There are a number of qualitative citation metrics that can be used. The first one is in which venues happen the citations. Being cited in a top conference or journal is certainly quite different and has a different value from being cited in an unknown conference. One way to provide a measure in this direction is to provide the list of the highest ranked conferences (given their percentile in the venue ranking) in which an author is cited. This can be done categorized per area what provides the additional information of whether an author is only known in its area or has been able to transcend her area and be known in other areas. The second qualitative measure is the periodicity of citations in top venues. In which venue or set of related venues an author is cited (giving the number of citations per venue and year).

The third qualitative measure is by whom an author has been cited. This measure can be provided by giving the list of authors out of the top-1000 most cited authors that cite the author being evaluated (providing the position in the ranking of the authors). This list can be either constrained to the top-1000 or extended to the top-5000.

4. Using the Tools to Obtain the Metrics

In this section we suggest how to obtain the different metrics to evaluate the research of a scientist in CS. It shows how freely accessible existing tools with high quality levels can be used to obtain the metrics.

Publication Evaluation based on Conference and journal ranking. In order to obtain the conference and journal ranking our proposed method is to use Microsoft Academic Search. One can obtain the ranking for CS conferences and journals. These ranking are HTML pages that can be introduced in an excel file. The ranking includes the total number of publications of each venue and the total number of citations received by the venue. From this data it is easy to compute the impact (citations/publications) and the percentile in the ranking (position/number of venues).

The proposed percentiles to categorize conferences are: 5%, 10%, and 15%. This can be cut depending on the competitiveness of the process where the research is being developed. In high competitive processes the percentile 5% is a good cut since it includes the top venues of most areas. In processes with low competition one can go till percentile 15%, although this percentile includes already a good fraction of poor venues. When considering more than one percentile, one counts the number of publications on each percentile. Then comparison is performed as with the Olympic medals, in lexicographic order from the top percentile to the bottom percentile. For instance, considering the percentiles 5%, 10% and 15%, a researcher with [1, 0, 0] would be ranked better than a researcher with [0, 0, 1000]. That is, the comparison is like in tennis, a player that has won 1 grand slam would always be better than one than won 1000 times a local tournament.

Another possibility is simple to use the percentile and use an exponential weighting to achieve the same effect as lexicographic comparison. This second possibility has the additional advantage that two venues just at the end of one percentile and at the beginning of the next percentile would be ranked very similar unlike the previous method.

The venue ranking based on percentiles is used not only to evaluate the research based on the publications but also to evaluate the impact of this research as we will see later.

Total number of citations without self-citations and derived indexes. The most complete tool up-to-date for obtaining the citations of an author is Google Scholar complemented with the Publish-or-Perish interface and the CIDS tool. The Publish-or-Perish enables to query Google Scholar and check/uncheck each of the papers recovered by the query to be counted/discounted from the publication and citation count, and citation indexes. This allows obtaining the correct set of papers of a particular author and their citations. CIDS can then be used to obtain the citations without self-citations. Based on this information it can be obtained the different citation indexes such as g-index and h-index.

Quality of Citations – Top Venues. This index aims at making a qualitative analysis of the citations. There are two flavors of the index, a simple one and a more complete one. The simple one provides the set of venues in percentile 5% in which a researcher has been cited. As before, this can be extended to other percentiles. This enables to compare the relevance of the impact obtained. When an author is cited in higher ranked venues the impact is more relevant. The more complex one provides the number of citations per venue and year in the percentile 5%. This provides the periodicity with which a researcher is cited in top venues.

Quality of Citations – Top Researchers. This index makes a qualitative analysis of who cites a researcher work. The citations from a researcher always come from a variety of researchers. However, some researchers are only cited by researchers with modest impacts, whilst others are cited by researchers with high impacts. Being cited by top-researchers is a distinguishing factor in the quality of citations. This index consists of the list of top-cited authors (and their rank position) that have cited the publications of the researcher under evaluation. This list can be extracted from the ranking made by Microsoft Academic Search that provides a good approximation for the top-1000 highly cited researchers. This can be extended to the top-5000 highly cited researchers for evaluations with low competition.

5. Discussion and Conclusions

The tools and metrics used for evaluation of research at funding agencies and hiring and promotion bodies have a crucial importance. They affect the target of new generation of researchers, and when this target is not the most relevant venues, many of the researchers will just be ruined and their time, and also the investment in their training, will be just wasted. This is what is happening with the extended practice of using exclusively journals and their JCR impact to evaluate research. As shown in previous sections, journals just capture a small fraction of the most relevant research and JCR performs a really bad job in evaluating their impact since it misses the citations coming from conferences, and also by missing the evaluation of conferences themselves. Also the use of 2-year (or more recently 5-year) windows is really inappropriate, especially taking into account than in CS journals the publication process typically takes 2 years and in some cases 3 years what makes that most citations made from the journals are just not counted. This contrasts with other sciences in which journals are the main means of publication and the review process is much agile and additionally, there is a lot of prestige associated to be reviewer of prestigious journals, similar to be in the programme committee of reputed conferences in CS.

Our proposal is that the CS community tries to promote the right set of tools and metrics to evaluate CS research at all levels, regional, national and international. Microsoft Academic search for evaluating venues and top-1000 researchers is promising, combined with Google Scholar, Publish-or-Perish and CIDS, to obtain the citations without self-citations of researchers and papers. This paper makes an effort in this direction. However, this will always require a continuous effort in arguing with policy makers and funding research agencies of making a case for a different way of evaluating research in CS. Possibly, the most plausible way to solve this situation forever is to lobby our research organizations such as ACM, IEEE Computer Society, USENIX, IFIP and others, in order to convince them to index all ACM and IEEE conference proceedings in JCR. Since they all have an associated ISSN, it is just a question of having them indexed. With this the situation of research evaluation in CS would be highly solved.

References

[AcademicSearch] Microsoft Academic Search. http://academic.research.microsoft.com/

[ACMDL] ACM Digital Library. ACM. http://www.acm.org/dl

[CIDS] CIDS: Web-based citation analysis discerning self-citations. University of Lisbon. http://cids.di.fc.ul.pt/

[Citeseer] Citeseer. http://citeseer.ist.psu.edu/

[Citeseer-X] Citeseer-X. http://citeseerx.ist.psu.edu/

[CORE] CORE. Computing Research and Education Association of Australasia. http://www.core.edu.au/

[Google] Scholar. Google http://scholar.google.com/

[PoP] Publish or Perish. V3.0. Harzing, A.W. http://www.harzing.com/pop.htm

[Scopus] Scopus. Elsevier. http://info.scopus.com/

[WoS] Web of Science. Thomson Reuters. http://thomsonreuters.com/products_services/science/science_products/a-z/web_of_science

The maintenance is the implementation

OR

Why people misunderstand IT systems

Enrico Nardelli
Dept. of Mathematics, Univ. Roma Tor Vergata
nardelli@mat.uniroma2.it - http://www.mat.uniroma2.it/~nardelli/

This short paper has been written more as a position paper intended to stimulate discussion than as a research or experience description paper.

1. The traditional scenario of mechanization

Since the 18th century humankind has reassigned to machines jobs previously performed by human beings and animals, and this process is well understood.

Assume you have to harvest a huge field or to assemble thousands of pieces or to execute billion of calculations.

In the pre-industrial era you needed a lot of people and possibly the help of several animals. In the industrial era you buy a machine and use just a few specialized persons.

The machine does the physical work, while the people drive it and supply the required cognitive processing. If there is a big stone in the field, a broken piece, a meaningless number, the person sees it, recognizes it, and avoids it.

This applies to almost all industry mechanization. Machines are a substitute for physical effort by humans or animals. They need no capability to adapt to a changing environment, since persons driving them have this capability. They need no intelligence (in the human sense), since persons guiding them can supply theirs.

Once you have mechanized and automated a task, more or less 80% of the job is done. It just remains a 20% maintenance job. This is the current view.

2. Computer-based mechanization and flexible/adaptive mechanization

Then computers arrived, and humankind started to automatically solve a different kind of tasks by means of this new kind of machines.

A computer is a machine automatically executing a "problem solving" task working on a suitable abstract representation of problem data and their relations.

However, this time the problems to be solved, i.e. the tasks to be automated, are the ones where mainly human intelligence is required, and not so much strength or manual ability. But, on the other side, a computer just “shuffles symbols around”: each step in the execution of its problem solving activities only changes strings of bits or symbols, attaching no meaning whatsoever to these representations and operations. E.g., there is no difference between a string representing a picture and one used to reserve a flight seat. It is only the intelligence of the human beings that reads them differently. It is indeed this blind procedure that enable us to apply this new kind of machines and solve so many different kinds of tasks, including running the machines themselves.

This is a completely different scenario from the ones humankind has experienced in the past centuries: now we are automating not just those tasks where there is a physical component and a cognitive component, using machines to execute the former and people driving them to take care of the latter.

We are now mechanizing almost purely cognitive tasks ("services", in modern parlance). Service automation has completely changed the usual mechanization scenario. We are replacing some intellectual capabilities of human beings with machines.

We have been able to successfully automate some of these cognitive tasks. For example those "control" tasks which mainly require a quick decision to react to many sensor-read values to execute a pre-defined action with a low cognitive content, e.g. controlling the strength and speed of car braking. Or those low complexity clerical tasks that can be fully described by means of a small number of fool-proof rules, e.g. money withdrawal from one’s own bank account. Both of them are usually industrialized in IT systems with a high success rate. Most of the industrial productivity increase in the last 30 years is due to the introduction of this type of mechanization.

But when more specialized knowledge is required we are truly substituting the human intelligence with a machine. And we do not witness so much success here[1].

One of the essential features of the human intelligence is the adaptability to a changing environment, the flexibility in coping with modified or new requirements. Human beings have a “built-in” capability of adapting to changes and learning from errors. Machines don't have it.

And we know the world around any organization is constantly changing, if not the day after, certainly the month after.

3. A change of paradigm

Apparently, we witness just a slight shift in the meaning of word “mechanization”. Instead, we have to cope with a dramatic change of paradigm. Our society has not yet understood it, and that's why people usually think: "now we have an IT system and we are done".

Wrong!

Once an IT system is built, the real job starts. "Panta rei": the environment changes, and the IT system has to be adapted, and requirements change, and the IT system has to be modified. And this goes on and on.

[1] Also, those "common sense" tasks where there is a low amount of specialized knowledge but a huge amount of common sense knowledge coupled to a sophisticate movement control, e.g. cutting the nails to an elderly person, are difficult to mechanize (may be even more than the previous ones), but they usually do not have a high cognitive complexity.

The IT professional community knows very well that software maintenance absorbs the largest share of the resources spent in an IT system lifetime. But our society at large is not aware of this, since most people - especially decision makers - are used to the old view of mechanization.

Hence they do not understand that building an IT system is only 20% of the job. The real - and expensive - job is managing its adaption to an ever changing environment. As an aside, we conjecture that this is one of the economic motivations for open source software: because it gives you for free just the first 20%, and then service companies are happy to be paid for the remaining 80%.

This misunderstanding is the main reason why IT system development efforts are late and over budget in a proportion embarassingly higher than for any other industrialized field. Whether they are a success or a failure is an orthogonal question not discussed here.

4. We can and should do something

Lawyers try to fix in contracts what an IT system should do. But legal descriptions in most cases do not work, since even if you are able to write a perfect contract specifying everything, the moment after you have finished it, the reality around the system has changed. Instead, contracts for hiring people usually work, because the human intelligence fills the unavoidable gaps existing in any contract.

One possibly effective approach to tackle this scenario is the use of a development "philosophy" inspired to the so called "agile methods". We deliberately avoid the use of the term "methodology" here, since the point is not about following more or less faithfully a sequence of steps but taking a completely different viewpoint on the development of IT systems.

With such a philosophy you build a system a piece at a time, constantly analyzing the solutions found, and adjusting to errors, like a person learns to do her job a bit every day. And you do this having defined a general plan and an overall design, but without detailing too much too early. The focus is on the responsibilities of those developing the system and of those driving, on the client side, this development, rather than on a detailed development plan often inflexible and difficult to adjust.

Another possible approach is to develop application development environments specialized for narrow application classes, where the end-user directly develops what she needs. This is the so-called “*citizen development*” that according to Gartner will have citizens build at least 25% of new business application by 2014 [1]. And the recent release by Google of “*App Inventor*”, the development environment making it possible to users of its Android-based mobile phones to develop their own applications, is a clear step in this direction [2].

Since many years informatics scientists working in the software engineering field are trying to understand better how the software development process can be shaped to better match these needs for adaptability and flexibility of IT systems. And to understand how we can produce software able to evolve in a changing environment by self-adapting their behaviour to new and modified situations ("situational computing" [3] and “self-managing situated computing” [4]), possibly using probabilistic models and reasoning tools to deal with uncertainty, and using the web as source of knowledge. We have an absolute need for this kind of results.

Still, the informatics community should probably act more proactively towards society in educating people that for the development of IT systems:

"***the maintenance IS the implementation***".

5. Conclusions

Twenty five years have passed since the resignation of David Parnas from the panel on "Computing in Support of Battle Management" convened by the Strategic Defense Initiative Organization [5] and his statement "*I believe that I have as sound and broad an understanding of the problems of software as anyone that I know. If you give me the job of building the <SDI> system, and all the resources that I wanted, I could not do it. I do not expect the next 20 years of research to change that fact.*" And I think the basic issues on which this statement is based hold still true, even abstracting away from the very strict requirements that this specific military system had.

In the almost contemporary and equally well-known paper [6] Frederick Brooks cited as the two most important issues to improve software production "*incremental development*" and "*great designers*". The first is a clear anticipation of the viewpoint taken by the Agile Manifesto in 2001, and the second is an anticipation of the conclusions contained in the report produced in 2006 by ACM's Job Migration Task Force led by Moshe Varde and Frank Mayadas, that only standardized IT tasks are offshored and the mission critical ones are kept in-house [7, 8]. Once again, the old paper provided a clear picture of the current situation.

Indeed, at least part of the Informatics community is fully aware of the issue I have addressed in this paper. As it was pointed out by the referee, this year International Conference on Advanced Information Systems Engineering conference (CAISE'10 – http://www.caise2010.rnu.tn/) has as its focus "Evolving Information Systems", stating that "*The evolution of an information system should be a continuous process rather than a single step, and it should be inherently supported by the system itself and the design of the information system should consider evolution as an inherent property of the system*". A perfect match with my views described in this paper.

But if, after more than 20 years, notwithstanding the warnings by two of the most reknown scientists in the software development field, society still considers IT systems just a sophisticate descendant of the plow and not a radically different new species, the problem of the role played by the informatics community-at-large in educating the society remains.

Acknowledgements: The author would like to thank the many colleagues he has shared his reflections with and whose comments greatly helped in focusing them: Silvana Castano, Paolo Ciancarini, Pierpaolo Degano, Paola Inverardi, Letizia Jaccheri, Maurizio Morisio, Daniele Nardi, Giancarlo Succi, Letizia Tanca, Franco Turini. Insightful comments from the reviewer have been greatly appreciated.

References

[1] "*Gartner Says Citizen Developers Will Build at Least 25 Percent of New Business Applications by 2014*", October 22, 2009, http://www.gartner.com/it/page.jsp?id=1212813.

[2] http://appinventor.googlelabs.com/about/.

[3] C. B. Anagnostopoulos, Y. Ntarladimas, S. Hadjiefthymiades: *Situational computing: An innovative architecture with imprecise reasoning*, J Syst. Soft. 80(12):1993-2014, Dec. 2007.

[4] Carlo Ghezzi, *SMSCom*, ERC Advanced Investigator Grant N. 227977 [2008-2013], http://www.erc-smscom.org.

[5] David Lorge Parnas: *Software Aspects of Strategic Defense Initiative*, Comm.ACM 28(12):1326-1335, Dec. 1985.

[6] Frederick P. Brooks: *No Silver Bullets: Essence and Accidents of Software Engineering*, IEEE Computer 20 (4): 10–19, Apr. 1987.

[7] William Aspray, Frank Mayadas, and Moshe Y. Vardi, Eds, *Globalization and Offshoring of Software*, Report of the ACM Job Migration Task Force, 2006.

[8] David Patterson: *Offshoring: Finally Facts vs. Folklore*, Comm.ACM 49(2):41-42, Feb. 2006.

Research Evaluation for Computer Science

An Informatics Europe report

Draft, not for general circulation. Prepared by the Research Evaluation Committee of Informatics Europe. Version 6.0, 20 May 2008

Executive summary

1. Computer science is an original discipline combining characteristics of science and engineering. Researcher evaluation must be adapted to the specifics of the discipline.

2. A distinctive feature of computer science publication is the importance of conferences, some of which are extremely selective, and books. Journal publication, while important for in-depth treatments of some topics, does not carry more prestige than top-quality conferences and books.

3. An important part of computer science research produces artifacts other than publications, in particular software systems. In measuring impact, these artifacts can be as important as publications.

4. In the computer science publication culture, the order in which a publication lists authors is generally not significant. In the absence of specific indications, it should not be used as a factor in evaluation.

5. Publication counts, weighted or not, must not be used as indicators of research value. They measure a form of productivity, but neither impact nor research quality.

6. Numerical impact measurements, such as citation counts, have their place but must never be used as the sole source of evaluation. Any use of these techniques must be subjected to the filter of human interpretation, in particular to avoid the many possible sources of errors. It must be complemented by peer review, and by attempts to measure impact of contributions other than publication.

7. Any evaluation criterion, especially if it yields a quantitative result, must be based on clear and published criteria.

8. Numerical indicators must not be used to compare research or researchers across different disciplines.

9. In assessing publications and citations, the ISI Web of Science is inadequate for most areas of computer science and must not be used. Alternatives, imperfect but preferable, include Google Scholar, CiteSeer and (potentially) the ACM Digital Library.

10. Evaluation criteria must themselves be subject to assessment and revision.

1. Research evaluation and its role

Research is a competitive endeavor. Researchers are accustomed to constant evaluation: any work submitted to a workshop, conference or journal — even, in some cases, an invited contribution — will be peer-reviewed; rejection is frequent, and even for a senior scientist remains a possibility for every new submission. Once a researcher's work has been accepted and published, it will be regularly assessed at all career stages against the work of other researchers. In addition to being evaluated, all researchers except for the most junior evaluate others: they act as referees, as participants in editorial board and program committees, as members of promotion and tenure committees, by responding to project evaluation requests from research-funding agencies, by writing letters of evaluation of colleagues as asked, often out of the blue, by various institutions.

The whole research management edifice relies on assessment of researchers by researchers. The criteria must be fair (at least as fair overall as can be expected of an activity circumscribed by human judgment); they must be clearly and publicly specified; and they must be globally accepted by the corresponding scientific community. This means in particular acceptance by the specific discipline involved: while other disciplines are often represented in evaluation processes, in particular for recruitment, it is not acceptable to impose criteria from one discipline on another, for example from an older, well-established science on a newer one that has developed its own distinctive principles.

In the case of computer science, a consensus has largely emerged in the US on both the peculiarities of the discipline and the properties it shares with others. This is in particular the result of the work of the Computing Research Association (CRA), which over the past three decades has represented the voice of academic computer science and established a fruitful relationship with other fields of research. An influential CRA report from 1999[1] defines, clearly and concisely, a set of "best practices" for the evaluation of computer scientists and engineers. The situation in Europe is less developed, as computer scientists have not so far made a concerted effort to explain the issues and principles to their colleagues from other disciplines.

It is one of the primary tasks of Informatics Europe, the association of Information Technology research and educational institutions in Europe, created in 2005[2], to make the requirements and specificities of this field of research and evaluation widely known.

The present Informatics report ,builds on the CRA's work; while it highlights only a few European specificities such as language diversity — for the simple reason that there are hardly any others to pinpoint, the criteria for research quality being the same anywhere in the world — it expands on some of the CRA report's points, and takes into consideration a number of developments that have happened since 1999.

---

[1] Computing Research Association: *Best Practices Memo — Evaluating Computer Scientists and Engineers for Promotion and Tenure*, prepared by a David Patterson, Lawrence Snyder and Jeffery Ullman; in *Computing Research News*, September 1999, available at www.cra.org/reports/tenure_review.html.

[2] www.informatics-europe.org

Computer science, the focus of this report, is a central part of Information Technology but not all of it. While many of its analyses and conclusions extend to IT as a whole, they may need some adaptations for fields such as digital media that overlap the humanities with somewhat different publishing traditions.

2. Computer science and its varieties

Computer science concerns itself not with computers but with*computing*: processing information using algorithmic techniques. The term *informatics*, popular in Europe, highlights the need for a broad perspective including human aspects of information technology. The present report applies this broad view and does not distinguish between the two terms.

Computer science research divides itself into three broad categories: Theory,Systems and Applications. The division is not absolute, as much research work on any side involves elements from the others, but is convenient as a broad characterization.

Theory research concerns itself with conceptual frameworks for understanding computations, algorithms, data structures and other aspects of computing. It can itself be divided into three rough subcategories:

- *Algorithms, complexity and combinatorics* (mathematical models for understanding machines and computations).
- *Semantics, specification and proofs* (mathematical models of programming and programming languages, in particular to ensure correct functioning).
- *Computational science* (mathematical models for high-performance computations).

All three variants make extensive use of mathematics, although the mathematics relevant for the first two cases is from domains not central to traditional scientific education: logic, formal languages, automata theory.

Systems research is devoted to producing artifacts and assessing their properties. The artifacts may be programs, but also systems that involve software along with other elements, as in “embedded systems” (cell phones, trains, air traffic control…) which include both software and hardware, and in “management information systems” which include both software and organizational processes. The main subdivision here is between:

- *Building* systems — research prototypes, but also software that is stable enough to be actually used for production.

- *Measuring* the properties of existing systems and processes. This is known as *experimental computer science* and draws some of its techniques from the natural sciences and statistics (for "performance analysis", which studies for example the throughput of networks).

These two variants are often intertwined, since researchers who build a system will also analyze their properties. Classified along its areas of application, Systems research includes such specialties as: software engineering, which studies the best ways of building and maintaining high-quality software systems, in particular large and complex ones; programming languages and their implementation (compilers, interpreters); human-computer interface and graphics; database research, which tackles the issue of managing rich repositories of information; networking and operating systems; security, which addresses the issue of maintaining integrity and privacy of information; and others.

The value of Theory and Systems research cextends beyond any particular application area of computing.**Applications** research is devoted to the specifics of computing for a particular discipline, for example computational chemistry or computational finance. This area is sometimes called "*Computational X*". Evaluating such research requires combining criteria specific to computer science and criteria from the application area — the *X*. For this reason the present report limits its scope to the other two variants, Theory and Research (sometimes called "core computer science"), leaving it to the reader to determine how to perform this combination for a particular *X*.

Theory research brings computer science close to mathematics and to *sciences* such as physics. Systems research in its experimental variant shares many properties with *engineering* research in areas such as electrical and mechanical engineering. This duality is part of the attractiveness of computer science, but also makes research evaluation more delicate, at it requires the proper mix between criteria generally applied to pure science and those appropriate to engineering research.

Whether Theory- or Systems-oriented, computer science research exhibits a strong set of distinctive characteristics, which have evolved over the half-century of its existence. (While it is possible to find precursor work all the way back to the ancient Greeks, Arabs and Indians, not to forget mathematical pioneers such as Pascal, the discipline as such took off with the appearance of computers after World War II, and the first CS departments in universities were created in the nineteen-sixties.) Embodying the discipline's spirit are a number of seminal concepts and paradigms, such as the notion of algorithm, computability, invariants, the distinction between specification and implementation (information hiding), recursion and fixpoints, the issues of scaling up, the role of notation, translation between languages, the duality between function and data, the static-dynamic duality, the role of concurrency, the notion of algorithmic complexity, the concept of protocol, fixpoints, refinement and a number of others. This common conceptual corpus is essential for computer scientists; experts from other disciplines sometimes do not realize it even exists, viewing computers as essentially a tool and computing as a supporting task rather than a distinct scientific domain. This view is all the less justified that almost all scientists today rely for their daily practice of science on developments — from computational methods to advanced text processing to the Internet to massive search mechanisms on the Web — have only been made possible by of the application of such core computer science concepts. It is the responsibility of computer scientists to educate their colleagues from other fields about both the scientific basis that computer science shares with these fields and the specifics of computer science research.

This overview leads to this report's first recommendation:

> 1. Computer science is an original discipline combining characteristics of science and engineering. Researcher evaluation must be adapted to the specifics of the discipline.

3. The computer science research culture

Whether from the nature of the discipline or from circumstances, computer science research has developed some distinctive properties.

3.1 Publication channels

Computer Science has developed a tradition where prestigious and highly selective conferences play a primary role for presenting new and original research. This is in marked contrast with disciplines where the prestigious publication venues are journals, conferences being the opportunity to present raw initial results. In computer science, some

In computer science, some of the highest publication value goes to conference proceedings such as POPL, PLDI, OOPSLA, ECOOP, ICSE (in the software engineering and programming areas), SIGGRAPH and Eurographics (in graphics and HCI) and others. These conferences ahave acceptance rates hovering between 10 and 20%, for example[3]:

- ICSE (International Conference on Software Engineering): 13%.
- OOPSLA (Object-Oriented Programming): 19%.
- POPL (Principles Of Programming Languages): 18%.

Archival journals in the corresponding fields have their role, but often as a way to publish more in-depth versions of papers already presented at conferences, with details that could not be included because of the more stringent page limitations. While many researchers take the time to publish such longer versions, there are also examples of excellent researchers with mostly conference papers. This can cause problems in multi-disciplinary research evaluations, where colleagues from other disciplines consider journal publication as the basic yardstick of recognition.

[3] Averaged in each case over last three conferences, 2005-2007; for full details and many more conferences see ase.csc.ncsu.edu/semap/

A related issue is the value of *books*. In many disciplines books are not considered important scientific contributions. In computer science (as in some fields of humanities), books can be the primary form of impact. Many computer scientist, when asked about the most influential publication ever, will cite Knuth's *The Art of Computer Programming*, a compendium of books. In the evolution of software engineering, seminal concepts (such as "Design Patterns") became known through books before other modes of publication.

In increasing order of prestige, the typical process for publishing a new idea involves some of the following steps. The author may first publish an internal *report* of his or her institution, to establish precedence. Next he might submit a paper to a *workshop* to test the waters; workshops typically have the advantage of a fast review process, and are often affixed to a conference on a more general topic, allowing joint attendance. The next step is a *conference* submission; here the reputation of the conference, well understood by members of the discipline, will determine the prestige of the publication. In many cases the process will stop here, but the author may produce a longer *journal* version. Or he may decide to write a *book* giving a full exposition of the ideas.

Any evaluation process for computer scientists must be compatible with the discipline's publication culture:

> 2. A distinctive feature of computer science publication is the importance of conferences, some of which are extremely selective, and of books. Journal publication, while important for in-depth treatments of some topics, does not carry more prestige than top-quality conferences and books.

Even if correctly assessed, however, publications are not necessarily the sole form of scientific contribution (as they may be in some other disciplines). For some researchers, especially when their work involves a Systems component, the best way to demonstrate value is often to produce a program or other artifact that attracts the attention of their peers and of the rest of the world. This may be a more irrefutable demonstration of impact than a dozen papers. As an example, the Google success story is, at its heart, based on a fixpoint algorithm (building on one of the seminal concepts listed above): "Page Rank", which determines the popularity of a Web page based on the number of links to it, globally computed through an iterative algorithm. Before Google was commercial it was a research success story. One of the outcomes of the research was a paper on Page Rank[4]; another was the Google site and software. The paper described an ingenious algorithm, one of many that get published all the time; but the site had — beyond its future commercial value — a *research* value that the paper could not convey: the demonstration, for millions of users, of the scalability (another one of the above concepts) of the approach. Had Messrs. Brin and Page continued as researchers and come up for evaluation, the software would have been just as important as the paper.

[4] Sergey Brin and Lawrence Page: *The anatomy of a large-scale hypertextual Web search engine*, in *Computer Networks and ISDN Systems*, 33: 107-17, 1998. Also at infolab.stanford.edu/pub/papers/google.pdf.

It should be noted that assessing the value of such contributions can be delicate, since one must also judge scientific merit: the mere report that a program has been downloaded a million times over the Internet is not by itself proof of its conceptual contribution. Conference and journal publication, with its well-defined peer review process, provides more easily decodable and hence more reassuring evaluation grids. In assessing computer science and especially systems research, however, it is improper to focus on publications alone and ignore contributions in kind:

> 3. An important part of computer science research produces artifacts other than publications, in particular software systems. In measuring impact, these artifacts can be as important as publications.

3.2 Coauthoring

Another component of researcher evaluation is the often difficult issue of determining an individual's contribution to a collective work. Various disciplines have different practices; mathematical articles often have a small number of authors, whereas work in experimental disciplines commonly includes many participants who all want to share some of the credit. The practice in computer science is quite different. Not surprisingly, the number of coauthors per article tends to be lower for theory-oriented work and higher for experimental work, but in either case they remain quite low as compared to the standards of the natural sciences, and higher than in mathematics. Compare:

- *Nature* over a year[5]: the maximum number of coauthors per article was 22 and the average 7.3.
- *American Mathematical Monthly*[6]: maximum 6, average 2.
- For both OOSPLA and POPL in 2007 (representative of high-quality software conferences): maximum 7, average 2.7.

In disciplines where numerous coauthors are the norm, researchers have developed elaborate author ordering conventions to suggest the extent of individual contributions. No such culture exists in computer science Apart from easily recognizable standard cases, such as a paper on a PhD student's thesis topic cosigned with the student's supervisor, it is not always easy to adjudicate contribution; there is a conflictbetween the frequent encouragement to embrace a collaborative research style and the possible dilution of one's contribution at the time of individual assessment.

The danger here is to distort evaluation by attaching importance to characteristics do not deserve it. Hence the next recommendation:

> 4. In computer science, the order in which a publication lists authors is generally not significant. In the absence of specific indications, it should not be used as a factor in evaluating researchers.

[5] Issues from 6 December 2007 to 29 January 2008. From http://www.nature.com/nature/archive/.
[6] All 2007 issues, articles only (excludes notes and correspondence). From www.maa.org/pubs/monthly.html.

4. Bibliometry

Under increasing pressure from political authorities and the public to show results, university administrations worldwide are drawn to the use of quantifiable results — often called as *bibliometry* — including basic and derived measures.

Basic measures include:

- Raw publication counts.
- Publication counts weighted by publication value (determined through some official ranking of the prestige of each kind of publication, for example specific journals and conferences).
- Citation counts, measuring not output but impact, estimated from the number of other works citing a given publication.

Derived measures such as the h-index (basic or normalized) and g-index are computed from formulae involving citation counts, as detailed below.

The very idea of using such indicators for researcher evaluation has triggered some negative reactions, such as an article[7] by David Parnas, a famous computer scientist, and a collective letter[8] of 93 Swiss computer science professors.

It is, however, unrealistic to discard the idea of numeric criteria entirely. Even if one ignores the political context (the need for universities to show tangible results, and the attraction of numbers), it is not clear that the suggested alternatives are always better.Many computer scientists cite peer review as their favorite evaluation method; but it is not without its own issues:

- The results of peer review are highly dependent on the choice of evaluators and their availability (the most competent evaluators are often the busiest).
- If peer review were to become the sole evaluation mechanism, researchers would spend most of their time evaluating others rather than doing research.
- Most fundamentally, peer review has the limitations of any process based on human judgment. In particular, since one can in most cases solicit only a small number of reviews, the result is highly dependent on the choice of reviewers.

The solution appears to be in a combination of peer review and objective indicators. These indicators should be assessed for relevance and reliability, as discussed next; and they should always be used through human interpretation, subject to critical analysis.

5. Numerical impact measurements, such as citation counts, have their place

[7] David Parnas: *Stop the Numbers Game — Counting papers slows the rate of scientific progress*, in *Comm. of the ACM*, vol. 50, no. 11, November 2007, pages 19-21, available at http://tinyurl.com/2z652a. Parnas mostly discusses counting publications, but deals briefly with citation counts.

[8] Letter to CRUS regarding Bibliometric Evaluation, at http://www.sarit.ch/crusletter/.

but must never be used as the sole source of evaluation. Any use of these techniques must be subjected to the filter of human interpretation, in particular to avoid the many possible sources of errors. It must be complemented by peer review, and by attempts to measure impact of contributions other than publication.

In defining acceptable numerical measures the first observation is that what is worth measuring is not volume but impact[9]. Publication counts, raw or weighted, are not relevant to assessment except as an indicator of overall activity and energy. Giving them any other role encourages publication inflation (a serious problem in research, the publication glut making it harder to spot interesting contributions), "write-only" journals with authors and no readers, conference sessions attended only by the speakers, and Stakhanovist research profiles based on quantity rather than quality.

6. Publication counts, weighted or not, must not be used as indicators of research value. They measure a form of productivity, but neither impact nor research quality.

Some publication counts take into consideration the value of the publication venue, according to some ranking of publications. This variant has the advantage of accounting for the peer recognition that follows from acceptance into a prestigious venue. The problem, however, is in defining the ranking. Most of the issues of individual researcher assessment apply to publication assessment as well. Many research agencies have predetermined rankings, which cause disputes and frustration and are particularly delicate to maintain in a fast-evolving discipline such as computer science. Rankings are often given a mantel of respectability through *publication impact factors*, which for being automatically computed are not necessarily more believable. Even the editor of *Nature* — one of the publications most favored by impact factors — has come out forcefully[10] against the concept. The only way to obtain meaningful publication rankings would be a process involving experts of a discipline. In the absence of such community consensus, weighing publication counts by journal rankings does not make them substantially better as an assessment criterion.

Citation counts are more relevant since they do assess impact. They are made possible by citation databases such as: the ISI Web of Science, which (as detailed below) is inadequate for computer science; CiteSeer, which attracted considerable attention when it was launched but seems no longer to be actively maintained; the Digital Library of the ACM (the main computer science society), promising but in its infancy; and Google Scholar, probably the most usable resource today but based on proprietary criteria.

[9] Beyond impact, the factor of real interest is research *quality*. A measure of quality, however, can only result from human judgment. Impact is one of the inputs to that judgment, and is more amenable to approximation through measurement.

[10] Philip Campbell: *Escape from the impact factor*, in Ethics in Science and Environment Politics, Vol. 8, January 2008, available at http://www.int-res.com/articles/esep2008/8/e008pp1.pdf

Citation counts too are subject to serious criticism. Probably the prime reason for the negative reactions cited above is the inconsiderate attempt to use the ISI Web of Science on computer scientists, as discussed in the next section. Even with more reasonable databases, however, concerns remain:

- Focus. What should be evaluated is research quality. Publication quality is just one aspect of quality; impact is just one aspect of publication quality; citations are just one aspect of impact.
- Identity. Some researchers' names are frequently misspelled in the record, leading to citations being lost. Conversely, contributions from different authors are lumped together, especially since it is common practice to retain only the first letter of first names, so that "J Smith", "J Schmidt" or "J Dupont" will have high counts. First and last names of Chinese or Hungarian authors are often switched. All databases— shockingly —have trouble dealing with diacritical marks (accents, umlauts), so that Professor Fröhr's publications may be allocated between authors Fröhr, Froehr, Frohr, and (writing jointly) Mr. Fr and Mr. Hr.
- Other errors. An INRIA report on bibliometry[11] cites nine different renderings of the affiliation of just four INRIA authors[12]. Another example is ETH Zurich, whose members come up under many different affiliations, some not making the difference with EPFL in Lausanne (as both are sometimes identified as "Swiss Federal Institute of Technology"). This requires particular caution in using the citation databases to compare institutions, as when ETH and EPFL compete for funds —although this has not prevented highly publicized assessments such as the "Shanghai ranking" from relying on them.
- Language. A frequent complaint about existing databases is their bias towards documents in English. This is less of an issue in computer science than in other fields such as mathematics and the humanities.
- Distortions. Surveys tend to be widely quoted by article introductions, to refer the reader to an overview of the general topic. Rather than the article that introduced a breakthrough concept, in a way that after the fact may be seen as incomplete and perhaps hard to read, followers may cite a posterior work that improved the presentation. The article that defined NP-completeness, a fundamental concept of computer science, is cited far less often than a later, more pedagogical presentation[13].
- Misinterpretation. Citation does not always imply positive recognition; scientists may cite a paper to criticize it or point out an error. A famous paper[14] describing a

---

[11] A-M. Kermarrec, E. Faou, J-P. Merlet (editor), P. Robert and L. Segoufin, for the INRIA Validation Committee: *Analysis document – What do bibliometric indicators measure?*, 12 September 2007. See also an interview of Jean-Pierre Merlet (in French) at www.inria.fr/actualites/inedit/inedit59_actu.fr.html.

[12] Including INRIA, INRIA ROCQUENCOURT, INFIA ROCQUENCOURT, INST NATL RECH INFORMAT & AUTOMAT ROCQUEN COURT, NAT RES INST COMP SCI & CONTROL.

[13] Dror G. Feitelson and Uri Yovel mention this example and many others from the CiteSeer database in *Predictive Ranking of Computer Scientists Using CiteSeer data*, in *Journal of documentation*, 60(1), 2004, pages 44–61 available at www.cs.huji.ac.il/~feit/papers/CitePred04JDoc.pdf.

[14] Roger Needham and Michael Schroeder: *Using encryption for authentication in large networks of computers*, in *Communications of the ACM*, 21(12), December 1978.

protocol contained an error, and has been cited by many publications on program verification to show that specific verification tools detect the error.

- Time effects. Citation counts are most meaningful for older contributions (and, as a consequence, established researchers). Newer work and newer authors may not have had the opportunity to get cited yet[15]. Reinforcing this concern is the observation that the value of some important work takes time to be recognized.
- Size effects. Citation counts are absolute numbers, but impact within a given scientific community is relative to the size of that community. A seminal publication in a specialized area will get cited less than an incremental contribution in a fashionable field with high publication activity.
- Networking effects. Even without any consciously unethical behavior, groups of authors with kindred interests tend to form Mutual Citation Societies.
- Political bias. Some authors hope (through a practice that is ethically questionable) to maximize their chances of submission acceptance by lavishly citing works of program committee members.

The last two observations lead to another common criticism of citation counts: the possibly perverse effects on research. According to this criticism, researchers who are evaluated through a specific quantitative criterion will adapt their activity to maximize its value, in this case citations, at the possible expense of research quality.

A number of publications analyze in further detail the flaws of available citation databases; even a quick look at Friedemann Mattern's work[16] suffices to temper any temptation to trust automatically collected measurements blindly.

The overall lesson is that the quality of an evaluation process based on data cannot be better than the quality of the data. Unfortunately the problems have not been alleviated since the citation sources came into being, and no organization appears to be working on a solution. One notable exception is the DBLP site at the University of Trier, maintained by Michael Ley[17], which unlike the other sources cited makes it easy to contact a human to correct an error; but this site lists publications, not citations. The recent ACM Digital Library effort, mentioned above, is also showing a willingness to react to criticism and correcting errors promptly.

Critical analysis and assessment of the indicators assumes that the method for collecting the data is transparent:

> 7. Any evaluation criterion, especially if it yields a quantitative result, must be based on clear and published criteria.

[15] Models have been proposed to replace the actual citation count by a predictive value for an author. See Feitelson and Yovel, note 13.

[16] *Bibliometric Evaluation of Computer Science — Problems and Pitfalls* (slide presentation); available at www.vs.inf.ethz.ch/publ/slides/Mattern-Bibliometry-SARIT06.pdf.

[17] www.informatik.uni-trier.de/~ley/db/.

In the present state of the main bibliometry databases this last requirement remains wishful thinking. The method by which Google Scholar and ISI select documents and citations are not themselves published, or subject to public debate. Decision-makers must consider this limitation when using quantitative data in their assessments of individual researchers.

One important caveat on the use of any such indicators is the wide difference between disciplines. Patterns of publications vary considerably across areas of science and engineering; the earlier comment that it would be inappropriate to judge one according to the rules of another particularly applies here[18]:

> 8. Numerical indicators must not be used to compare research or researchers across different disciplines.

5. The ISI case

An issue of concern to computer scientists is the common tendency to use as reference for publications and citations the database of Thomson Scientific's ISI Web of Science. This resource was devised for the natural sciences; while the corresponding community seems to be satisfied with its applicability there, it is inadequate for computer science.

The principal deficiency of ISI comes from its arbitrary classification of what is or is not worthy of being counted. The selection criteria are arbitrary and opaque. Most conferences are not listed; books are generally not listed; conversely, some references are included indiscriminately.

The results make any computer scientist cringe[19]. For Niklaus Wirth, a famous computer scientist honored by the Turing Award (the highest honor in computer science, commonly referred to as the Nobel equivalent of the discipline) and known in particular for his design of Pascal, the ISI database lists a number of minor papers that happen to have appeared in indexed publications, but not his 1970 "Pascal User's Manual and Report" (with Kathleen Jensen), published as a book and one of the best known references in all of computer science. Ask any computer scientist what is *the* most influential publication in the discipline, and most will cite Donald E. Knuth's *The Art of Computer Programming* book series, which has acquired legend status; that reference does not figure in the ISI database. (On Google Scholar it gets over 15,000 citations, an astounding number.) Of the many articles that Knuth — also a Turing Award winner — has published, the three most frequently cited according to Google Scholar, each with about 1000 citations, do not even appear in the ISI records.

[18] This is one of the recommendations of the INRIA document by Merlet et al (note 11).

[19] All ISI results obtained from Web of Science searches on 4 February 2008, through entire database (options: Timespan=All Years. Databases=SCI-EXPANDED, SSCI, A&HCI, IC, CCR-EXPANDED).

Evidence of how ISI collapses for computer science is "internal coverage": the percentage of citations that cite a publication in the same database. Whereas ISI's internal coverage exceeds 80% for physics or chemistry, it is only 38% for computer science[20].

An example of the arbitrariness of ISI criteria is Springer's *Lecture Notes in Computer Science* (LNCS), which ISI for until recently classified as a journal, whereas it is simply a book series offering numerous conference proceedings and some monographs. Lumping all LNCS publications into a single journal category was inappropriate, especially since many high-quality conferences not published by LNCS are not listed[21]. For example:

- The International Conference on Software Engineering (ICSE), the top conference in a field that has its own special ISI category, is considered a premier publication venue by anyone in the field; it is not indexed by ISI.
- Any software engineering workshop published in LNCS, the kind of venue where an author would typically try out an idea *before* it is ready for submission to ICSE, is indexed by ISI[22].

As another example, ISI indexes *SIGPLAN Notices*, a publication of the Programming Languages group of the ACM (one of the two major professional societies in computer science). SIGPLAN Notices is actually an *unrefereed* publication, used in its ordinary issues to publish drafts, notes, letters; but it devotes special issues to the proceedings of some of the most prestigious conferences such as POPL and PLDI. Unlike those to ICSE, contributions to these conferences will appear in ISI, but treated in the same way as an informal reader's note in a regular issue.

The database has little understanding of what constitutes computer science. The 50 most cited references in computer science according to ISI[23] include such entries as "*Chemometrics in food science*" (#13), from a journal called *Chemometrics and Intelligent Laboratory Systems*", a topic and a publication entirely alien to computer

[20] From Moed H.F., Visser M.S. *Developing Bibliometric Indicators of Research Performance in Computer Science: an Exploratory Study*, in ISSI 2005 (Proc. 10th Intl. conf of Intl. Soc. for Scientometrics and Infometrics), Karolinska Univ. Press, Stockholm, 2005, pages 275-279. Cited by Merlet et al (note 11).

[21] The LNCS publishers themselves are careful not to misrepresent their offering. The official LNCS site presents the series as: "*a medium for the publication of new developments in computer science and information technology research and teaching — quickly, informally, and at a high level*" (www.springer.com/computer/lncs?SGWID=0-164-6-73659-0).

[22] An example among hundreds: *Proceedings of the 9th International Symposium on System Configuration Management*, LNCS 1675, 1999. The theme is configuration management, a subtopic of software engineering. An ICSE paper on this topic, arising from a revision of a contribution to this workshop, would not be listed in the ISI Web of Science.

[23] ISI query including all "*COMPUTER SCIENCE*" topics except one: *TS=(COMPUTER SCIENCE, INFORMATION SYSTEMS OR COMPUTER SCIENCE, HARDWARE & ARCHITECTURE OR COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS OR COMPUTER SCIENCE, THEORY & METHODS OR COMPUTER SCIENCE, SOFTWARE ENGINEERING OR COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE) Databases=SCI-EXPANDED, SSCI, A&HCI, IC, CCR-EXPANDED*. The omitted category is "*COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS*", to avoid a bias against core computer science; predictably, including it yields even more outlandish results.

science. This is not just an isolated example; most of the entries on the list, even those which are related to computer science (usually from specialized areas rather than the core of the discipline) are not recognizable to a computer scientist as milestone contributions. The cruelest comparison is with the list of most cited computer science works on the CiteSeer site, devoted to computer science[24]; while imperfect like any such selection, the CiteSeer lists many articles and books familiar to all computer scientists. It has *not a single entry in common*[25] with the ISI list[26].

Merlet et al.[27] note that the top-ranked ISI journal is 195th on CiteSeer, and the top CiteSeer journal is 26th for ISI. While some might be tempted to use this as a reason to dismiss rankings altogether, examination of the differences shows that they simply reflect how far off ISI is from the general understanding of computer scientists.

The ISI list of "highly cited researchers" reflects the database's ignorance of computer science. Wirth, Parnas and Knuth, all iconic names in the field, do not appear. Of the ten Turing Award winners between 2000 and 2006, only one is listed (Ronald Rivest, the R of the RSA cryptographic algorithm), but not, for example, Adi Shamir (the S of RSA), another revered figure of theoretical computer science[28].

Although one might indeed expect ISI to give better results for Theory work, closer than Systems to mathematics (a long-established discipline), the difference is marginal because of the fundamental deficiencies in the process.

Since ISI indexing is based on an opaque process with no room for assessment or appeal of decisions, the situation is unlikely to improve.

Alternatives such as CiteSeer and Google Scholar are subject to criticism as well. While CiteSeer attempts to eliminate self-references, Google Scholar does not; neither project publishes its precise inclusion criteria[29]. These deficiencies, however, are negligible when viewed against those of the ISI Web of Science:

> 9. In assessing publications and citations, the ISI Web of Science is inadequate for most areas of computer science and must not be used.

---

[24] citeseer.ist.psu.edu/articles.html, consulted February 2008; data from August 2006.

[25] One author (Scott Kirkpatrick) appears on both lists, for different articles.

[26] [Note applicable to this draft: Goodrum, McGain, Lawrence and Giles, in *Scholarly Publishing in the Internet Age: A Citation Analysis of Computer Science Literature*, in *Information Processing Management* 37, pages 661-675, available at wotan.liu.edu/dois/data/Articles/juljuljiqy:2001:v:37:i:6:p:661-675.html, found (in 2000) a small but non-empty common subset. I am investigating this discrepancy, in particular whether the ISI queries used in the preparation of this article are the right ones.]

[27] See note 11.

[28] Of the just announced three 2007 winners, two are in the ISI list.

[29] For a general comparision between Google Scholar, ISI and other sources (including Elsevier's Scopus) see Anne-Wil Harzing, *Google Scholar — A New Data Source for Citation Analysis*, available at www.harzing.com/resources.htm#/pop_gs.htm. Anne-Wil Harzing is the author of the "Publish or Perish" tool discussed in the next section. A detailed study: is Lokman I. Meho and Kiduk Yang, *A New Era in Citation and Bibliometric Analyses:Web of Science, Scopus, and Google Scholar*, to appear in *J. of Amer. Soc. for Information Science and Technology*, available at arxiv.org/ftp/cs/papers/0612/0612132.pdf.

Alternatives, imperfect but preferable, include Google Scholar, CiteSeer and (potentially[30]) the ACM Digital Library.

Anyone in charge of research assessment should be aware that attempts to use ISI for computer science will cause massive opposition. Announcements of such plans have led some computer scientists to reject all measurement-based techniques, as in the examples cited above. This is an overreaction; but decision-makers and scientists from other fields must not try to impose on computer scientists a scheme that is demonstrably inapplicable to their discipline.

Beyond the specific deficiencies of ISI, all systematic studies of citation databases show wide variations between the results they yield, in particular for computer science. This has led some authors[31] to suggest systematic reliance on *several* databases, a commendable practice that seems hard to impose in practice. Perhaps software will appear that performs this automatically. In the meantime it is again essential to remember the limitations of data quality in today's databases, and avoid any career-affecting decision based on metric indicators whose validity has not be checked thoroughly.

6. Assessment formulae

A recent phenomenon, not addressed in earlier studies such as the CRA report, is the growing reliance on numerical measures of an author's impact, derived from citation databases. The most commonly cited formula is the **h-index**, defined as the highest n such that C $(n) \geq n$, where C (n) is the number of citations of a publication by the author. The justification for this formula seems to be that it correlates well with other measures of success such as Nobel prizes, although to our knowledge the supporting studies did not involve computer science research. Variants of the h-index include:

- The **individual h-index**, obtaining from the h-index by dividing it by the number of authors, with the goal of better assessing individual contributions.

- The **g-index**, for which the value is the highest n such that that the top n publications received (together) at least n^2 citations.

The g-index corrects a significant deficiency of the h-index: that it does not recognize extremely influential individual publications. If your second most cited publication has 100 citations, it does not make a difference to the h-index whether your top publication has 101 citations or (as in Knuth's case above) 15000. The g-index corrects this.

The "Publish or Perish" site[32] makes it possible to compute these indexes for any scientist in a few seconds.

[30] At the time of writing, the Digital Library citation analysis mechanism is still in pre-release.
[31] Lokman E. Meho: The Rise and Rise of Citation Analysis, Physics World, January 2007; available at eprints.rclis.org/archive/00008340/01/PhysicsWorld.pdf.
[32] www.harzing.com/resources.htm#/pop.htm.

It would be as counter-productive to reject these techniques as it would be to use them blindly to yield a single magic number defining a researcher's value. In computer science as in other fields, there is no substitute for a careful evaluation process involving many complementary sources of information. Peer review is one of these sources; properly qualified and interpreted numerical measures can be another valuable one.

7. Assessing the assessment

Negative reactions to assessment formulae often elicit in return the reproach that the complainants are sore losers or refuse to go with the times. This is generally unfair. Any scientist, as noted at the beginning of this report, is accustomed to evaluation as a constant fact of life. What causes irritation is reliance on inadequate assessment methods.

Scientists are taught to use rigor in their own work: to submit any hypothesis to scrutiny, any result to duplication, any theorem to independent proof. They naturally assume that assessment processes affecting their own careers will be subjected to high standards too. Just as they do not expect, in a discussion with a PhD student, to impose a scientifically flawed view on the sole basis of seniority, so will they not let university management impose a flawed evaluation mechanism on the sole basis of authority.

The principle of collective self-assessment, which has been instrumental in the development of modern science, must continue to apply even as technology brings about new tools of evaluation.

The first step is to ensure, as noted above, that evaluation criteria are public. But they should also be justified on rational grounds, and subject to constant reassessment. This is particularly true of computer science because of the fast evolution of the discipline.

> 10. Evaluation criteria must themselves be subject to assessment and revision.

Openness and adaptability are the price to pay to ensure a successful process, wholeheartedly endorsed by the computer science community.