Towards a strategy to fight the computer science (CS) declining phenomenon

Marcela Porta, Katherine Maillet, Marta Mas-Machuca, Carmen Martinez-Costa

Chapter #

TOWARDS A STRATEGY TO FIGHT THE COMPUTER SCIENCE (CS) DECLINING PHENOMENON

Marcela Porta[1], Katherine Maillet[2], Marta Mas[3] and Carmen Martinez[4]

[1] *Telecom & Management SudParis,* ***9 rue Charles Fourier 91011 Evrey, France. Marcela.morales@it-sudparis.eu***

[2] *Telecom & Management SudParis,* ***9 rue Charles Fourier 91011 Evrey, France. Katherine.maillet@it-sudparis.eu***

[3] *Universitat Politècnica de Catalunya,* ***Avda. Diagonal, 647, pl. 7. 08028, Barcelona. Marta.mas-machuca@upc.edu***

[4] *Universitat Politècnica de Catalunya,* ***Avda. Diagonal, 647, pl. 7. 08028, Barcelona. Mcarme.martinez@upc.edu***

Abstract: European students have give reasons why they reject computer science (CS) as a program of study in higher education [1]. The constant decrease in the number of students choosing to study this subject has had consequences in different European sectors, such as the economic sector or the education sector, among others [2]. Some of the reasons for this rejection are related to the degree of difficulty of the skills that are needed to master CS. This study aims to identify these skills by firstly comparing CS curricula across several European universities and then comparing these curricula with same level programs in other areas of study. It will highlight some misconceptions students have concerning CS programs and will demonstrate that Technology Enhanced Learning (TEL) may have a major role to play in combating the decline of CS professionals by providing dynamic learning environments; where students can acquire the knowledge and skills which are unique to the field of CS.

Keywords: Technology Enhanced Learning; Computer Science Education.

1. INTRODUCTION

The results of previous investigations have determined a declining interest in CS studies in Europe and presented the consequences of this decline [2]. Furthermore, other investigations will identify the reasons that retain students to enroll in CS [1].

Among this reasons figured: the degree of difficulty of the subjects required to master CS (fear of mathematics), a lack of understanding of the career at the moment of choice, or misconceptions about the role played by a computer scientist in society and the economic sector.

In order to clarify these ideas, this study will discover the generic content in European CS programs, aiming to prove that this cause should not represent a reason to reject the career.

The comparison of the CS programs of several European universities will make it possible to identify the knowledge and skills needed to become a computer scientist. Then, these skills, or common fields of study, are compared to same level programs in other areas of study to identify which of them belong exclusively to CS. The results of this investigation are confronted to job offers in the market.

Future work will suggest the usage of TEL tools and solutions to acquire the knowledge and skills needed to become a computer scientist and therefore attract and retain people to this field of study.

2. BACKGROUND AND RELATED WORK

Official European statistics (Eurostat) and the Computing Research Association (CRA) indicated in 2008 that the number of students that have access to tertiary education in Europe is increasing in a sustainable way [4, 5]. Unfortunately the increase in tertiary studies is not reflected in all disciplines. Graduates in science, mathematics and computing have decreased each year, reflecting the lack of motivation to follow scientific careers from the part of students.

Some countries, having a similar economic development as in Europe (like United States and Canada), have shown similar tendencies in the number of students enrolled in technology domains. As a result, decreasing numbers of student enrolments, graduates and CS courses offered in curricula have set off a general alarm [5]. Some experts called this situation "*a serious warning sign*" [6] as they discover that fewer schools are offering CS classes, which means fewer students are being formed with CS skills. Other related concerns are teacher certification levels and the lack of solid information to help understanding and fighting this problem [7].

In order to understand the importance of the CS decline in Europe, it is primordial to know what consequences can take place if the decline continues. Previous studies have identified economic and educational consequences as their main concern [2].

From the economic point of view a rising in the local CS development can be expected. As explained in the economic model of elasticity, the price of a product is determined by its offer and demand in the market [8]. When the number of CS developers decreases, then, their products will become more expensive and scarce. Thus, it is crucial to maintain a large number of students and experts in CS in order keep technological development costs down and in order to sustain the growth of this industry in Europe.

Another economic impact is the migration of the industry to countries outside Europe. The increasing prices of local CS development and the decreasing numbers of professionals to satisfy the demands from industry have driven European companies to redirect or migrate their labor forces to foreign countries where CS development work is not only as efficient as in Europe, but also, where there is a greater available production capacity at a lower cost [9]. Predictions about job losses related to shifting high-technology work to low-wage nations with strong education systems, like India and China, were greatly exaggerated. As remarked by Lohr, S. 2006, "*The concern is that misplaced pessimism will deter bright young people from pursuing careers in computing, and, in turn, would erode the skills in a field that is crucial to the nation's economic competitiveness*".

From the educational point of view, there is an increasing need to learn other languages to succeed in negotiations with other countries. However, the cultures and languages of the countries which are involved in technology development are not related to the European. Therefore, European languages such as English, French, German and Spanish (the most widely spoken languages in Europe and in the world) may become less useful to European industry than Chinese or Russian and possibly no longer be needed. Many studies pointed out the importance of learning languages to facilitate the migration for business. "*As access widens, unique educational modules, courses and programs are being designed and evaluated throughout other regions, evidencing issues, challenges, opportunities and initiatives related to this education*" [10].

Other educational changes are reflected in CS curricula, as it should be adapted in order to offer courses of study that will train students for the kind of language and management skills that will be needed by competitive European industries playing in global markets [11].

3. UNDERSTANDING THE REJECTION OF CS STUDIES IN EUROPEAN COUNTRIES

The World Congress of Engineering and Computer Science (WCECS) published in 2010 an investigation to identify the motivations from the part of student to choose or reject CS as a career in university [1]. The study called "Dec-CS: The computer Science declining phenomenon", was designed to better understand the social perception of CS and to identify how these perceptions influence a student's choice.

An analysis of the responses made it possible to list the reasons that motivate the students to reject CS. Among the obtained answers: the perceived degree of difficulty represents a barrier, the misconception of the social function also matter, the unclearness of the domain, etc., The reasons for rejecting CS in university are represented in Figure 1 and detailed below [1].

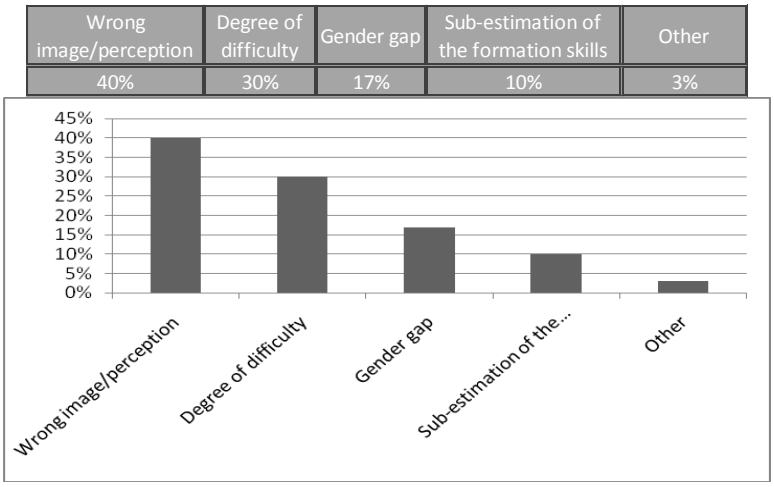

Figure # - 1. Dec-CS: The computer science declining phenomenon; "Reasons to reject computer science as a career in university".

Wrong Image perception of the career: One of the main reasons that retains mayors from following technology studies, is the image reflected by the CS program. Between the answers, words like "geek", "nerd" or "lab rat" were applied in order to identify the perceived image from the part of students, meaning that the person that follow this studies is identify as not very successful in social life. This stereotype is not always true, but still, is changing the mind of high school students, who as teenagers will discard the idea of becoming social-rejected in university. Other related complain is the difficulty to identify the role of a computer scientist in society, or either real

or fiction characters as examples. "*Other careers like medicine, military and even a builder are more clear to us as they are represented with lots of examples and defend their role in society in a better way"*[12].

High degree of difficulty: The image of CS requiring a strong background in mathematics and algorithms is a major fear which prevents students from enrolling. However, the professionals in this field suggested during this investigation (Dec – CS) that this difficulty is easy to overcome and that resources are provided to improve the student's level in these subjects to help them affronting their fear of CS. It can be concluded that the difficulty reflected in the CS program might be overestimated by the students before making the career choice, but from the part of people that have chosen CS it does not represent a justifiable reason to reject the career. Some references agree that the reason why a student does not choose CS or does not feel attracted to technology as a field of study is related to the degree of difficulty these careers reflect, presuming that the decision is determined by the content and quality in the discipline [13, 14].

Gender Gap: Women still feel that some scientific careers related to CS present gender issues. They pointed out that the time requirements, the lack of other women that will accompany them and the negative image the career has when it comes to feminine motivation are some reasons to take into consideration at the moment of choosing a career. It is important to remark that strategies designed to attract women to CS represent a great opportunity to increase the total number of students in this field. Other investigations are making an effort to increase gender diversity in engineering degrees, because they estimate that the number of women enrolled is still very low [15].

Underestimation of the formation skills: New technologies allow a person to easily acquire CS knowledge such as HTML or JavaScript thanks to useful autodidactic tools. These skills are recognized as important for the investigation's participants, e.g. for including them as proficiencies that can represent much in their curriculum. However they think is not necessary to dedicate a complete career to learn them. Experts in the CS faculty are concerned with this attitude from the part of students when these ones perceive that "*computer science is just programming*;" and that "*Faculty must consider ways to move students toward the idea that the work you do in computer science in the real world requires a lot of creativity, not only programming and that it can be dynamic*" [16].

4. TOWARDS A STRATEGY TO FIGHT THE CS DECLINING PHENOMENON

As presented before, students have several reasons to reject CS in university. Some of these skills are related to:

- The fear of affronting the degree of difficulty reflected in the CS program,
- The difficulty of identifying themselves with the CS engineer stereotype.

It is hard to determine, how students get to this conclusion, when they pointed out other facts like:

- The misunderstanding of the computer scientist role,
- The difficulty to identify the job they will perform after pursuing these studies.

This contradiction makes us get to the conclusion that they are overestimating the degree of difficulty (they might be fearing what they don't understand), and that the image of the career can be improve to provide some solutions.

As a strategy to help solving this problem, we propose the identification of the CS knowledge and skills in a European CS program. By identifying these common subjects, it will be possible to:

- Determine the degree of difficulty of the program: observing if it should represent or not a general fact of rejection.
- Provide the description of a Computer scientist, clarifying its role as a professional.

To confirm the truthiness of these common subjects, they might be matched to skills needed in job descriptions, corresponding to CS demand.

5. DISCOVERING THE NEEDED SKILLS TO BECOME A COMPUTER SCIENTIST

The following information represents the analysis of different curricula in several European universities. As CS can have a lot of roots and each university and country have modified programs and content according to different needs, we took System Engineering (SE) as the basis of this investigation. This analysis will identify the skills in common that represent the real challenge for a student that follows CS in university. The comparison between curricula has been done in 6 universities of different European countries. The names of the universities are listed below:

1. *Universitat Politècnica de Catalunya, Spain*[1]
2. *Université d'Evry Val d'Essone, France*[2]
3. *Tallin University of Estonia*[3]
4. *Catholic University of Leuven, Belgium*[4]
5. *University of Reading, United Kingdom*[5]
6. *Heidelberg University, Germany*[6]

Furthermore, the list of skills was also compared to curricula in other fields of study in order to determine which competencies belong exclusively to CS. The compared domains where:

Management: As a career that is having great impact in our days; to observe if the CS curricula propose some management skills.

Graphic design: As a domain that can sometimes include computing systems and programs related to CS e.g. animation and design tools.

Construction Engineering (civil engineering): to take a popular engineering as a reference.

Electronic Engineering: Because often, people that are not related with CS find confusing this domain and compare them both. In other words, it is difficult to identify where the system engineer work starts or finish when he or she has worked in collaboration with an electronic engineer.

The objective analysis of this comparison permitted to list the different unique knowledge and skills required to become a computer scientist. By "unique", we understand that these skills can only be obtained by pursuing a CS degree, and that without these competencies, a professional will not be able to perform CS work.

The identified knowledge and skills, resulting from this investigation, are explained below:

Mathematical Logic: Is the most common domain between CS curricula and represent a competency that needs to be mastered by the student in order to become a computer scientist. As a general use, this knowledge will teach a

[1] http://www.upc.edu/
[2] http://www.univ-evry.fr/fr/index.html
[3] http://www.ttu.ee/en
[4] http://www.uclouvain.be/
[5] http://www.reading.ac.uk/
[6] http://www.uni-heidelberg.de/index_e.html

student the bases of programming and also the understanding of logic structures [17]. Mathematical logic is a subfield of mathematics with close connections to CS and philosophical logic [18]. The field includes both the mathematical study of logic and the applications of formal logic to other areas of mathematics. The unifying themes in mathematical logic include the study of the expressive power of formal systems and the deductive power of formal proof systems. In conclusion, the mathematical content in the CS program is different from the one learned in high school; dough, it is a positive aptitude to have a good mathematical degree of understanding, it is not necessary to have a superior score in math during school to succeed as a computer scientist.

Algorithms: The second important content in the CS curricula is directed to understand the usage of algorithms. Contrary to an erroneous perception, this scientific area is more related to logic than to mathematics itself. An algorithm can be an effective method for solving a problem expressed as a finite sequence of steps. Algorithms are used for calculation, data processing and many other fields [19]. This competency has the characteristic to be very simple or very complex. It all depends on the teaching methods and objectives of the university program.

Figure 2 shows the representation of a simple algorithm that aloud a student to understand the concept of it.

This is an algorithm that tries to figure out why a lamp doesn't turn on and tries to fix it using different steps. A more specific approach of this skill is to be able of measuring programs executions, performance and time required to use it. Each algorithm represents a list of well-defined instructions for completing a task.

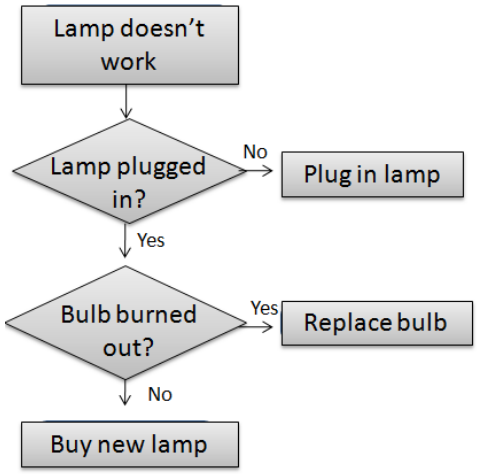

Figure # - 2. Representation of a simple algorithm

Programming: In simple words, programming is the process of designing, writing, testing and maintaining the source code of computer programs. This source code is written in programming languages and may be a modification of an existing source or something completely new.

The purpose of programming is to create a program that exhibits a certain desired behaviour (customization). The process of writing source code often requires expertise in many different subjects, including knowledge of the application domain, specialized algorithms and formal logic [17].

With good programming bases, a student is able not only to create his or her own programs, but also to understand others, written in various programming languages. After acquire this skill, the computer scientist can adapt the knowledge to all kind of progress in technology. For example, is capable to understand JAVA without necessary having learned it in a classroom [18]. Another example of the advantages of a good programming bases, is being able to use new technology integrated to older one learned before (e.g. CSS in HTML) and without having followed this course.

When students learned to program in a tool called Pascal, ultimate knowledge is transferred by the Massachusetts Institute of Technology (MIT) in another tool called SKIM. This fact, sustain the theory that knowing the basis of programming is better than learn how to use a specific programming tool. As Skim, there exist other tools created with the objective to program software i.e. C++, JavaScript, C# etc.

Data Bases (DB): This skill represents a complex part of the CS programs. A database consists of an organized collection of big amount of data for one or more uses, typically in digital form [19]. The databases in CS can be taught in two different forms, the first one consists in understanding one or several examples of databases e.g. Oracle, Access etc… The second more complete, consist in understanding the mathematical bases and logic of the system, permitting the capacity to analyze any database after that [18].

Data modelling: Data modelling is a method used to define and analyze data requirements needed to support the business processes of an organization. The data requirements are recorded as a conceptual data model with associated data definitions. Actual implementation of the conceptual model is called a logical data model [20].

Operative systems: Is another subject that can be taught in different ways (depending on each university system). It consists in learning one or several operative systems like Windows, Linux or UNIX. However, a university method cannot teach a student to use every operative system in the market. Therefore, a good preparation will consist in helping the students understanding the conception of them by creating their own operative systems.

Other information: After analyzing the curricula related to CS and other domains (management, graphic design, construction engineering, and electronic engineering) we conclude that mathematics, as an exact science, is not specially required to become a computer scientist. The part of the curricula related to mathematics (mathematic logic) and the part of the curricula related to logic (Algorithms) can be learned during the university degree. However, a basis of understanding is needed in order to learn these areas in an effective way. Some universities do not take mathematics into consideration while designing their curricula, others, include a mathematic course as an introduction to the career not necessary related to the mathematical logic or algorithms in the program [21].

Another subject that might create controversy during the training to become a computer scientist is the learning of English as a second language (if it's not a native language).

Some CS experts affirm that in addition to English courses to learn the speaking language, they would like to have an engineering English course, with the introduction to subjects and abbreviations for terms written in English [22]. Other CS experts affirm that learning English is important, because a good percentage of books and forums (important to the domain) are available only in this language.

We assume that this is a needed skill to any career and that it's usage in jobs will depend of the type of activity the person performs.

6. TESTING THE INFORMATION: CONFIRMING THE PROFILE

This part of the investigation aims to prove that the information given before is correct. As an example four profiles of System Engineering job offers are taken from a public and famous research system. The online system permit to find and postulate for actual job offers in almost any European country. These examples represent opportunities given in France, no longer than one month ago.

Profile 1: System Manager

"*Knowledge of C, C++ and experience with UNIX and/or LINUX platforms is necessary. Experience working with large existing software systems or development of C++ libraries is highly recommended. Experience in Place and Route Optimization, Timing Analysis or Logic Synthesis is considered a plus. You will need excellent programming and software engineering skills and preferably previous experience in the EDA industry. Experience with multithreaded and/or distributed programming is a big plus. Self-motivation, self-discipline and the ability to set personal goals and work consistently towards them in a dynamic environment will go far towards contributing to your success*".

Profile 2: IT-Project leader

"*Object orientated programming, Oracle database (basic knowledge), SQL database, XML language. Operating systems and tools: UNIX (Linux/AIX - basic knowledge), Windows, Eclipse, Telnet, FTP, VPN, SQL Developer (or comparable DB tools)*".

Profile 3: IT-Developer

"*Object orientated programming - design patterns, Oracle Database (basic knowledge), PLSQL Database (advanced knowledge) and SQL Database (advanced knowledge). Operating systems and tools: UNIX (Linux/AIX - advanced knowledge), Windows, Eclipse, Telnet, FTP, VPN, SQL Developer (or comparable DB tools)*".

Profile 4: IT-Application Engineer (Put into Operation)

JAVA (basic knowledge), object orientated programming, Oracle DB (basic knowledge), Perl, CVS, PLSQL (basic knowledge), SQL (basic knowledge), XML (basic knowledge). Operating systems and tools: UNIX (Linux/AIX - basic knowledge), Windows, Eclipse, Telnet, FTP, VPN, SQLDeveloper (or comparable DB tools).

It is important to highlight that this offers concerned French territory but is originally written in English; though they don't ask English as a required skill, it is needed in order to understand the offer.

7. PROFILE DESCRIPTION

Thanks to the identification of the needed knowledge and skills necessary to become a computer scientist (system engineering) and to the comparison with job offers related to CS, it is possible to design a profile that will better represent the experts in this domain.

A system engineer is the person that understands, develops, maintain and use computing systems. Thanks to unique competencies such as the understanding of mathematical logic, algorithms, programming, DB, data modeling and operative systems; this person is capable to answer to a specific demand in the market: the creation, use and maintenance of software.

Other authors affirm that in order to achieve qualifications and work-ability in CS, one may seek technical or professional certification (diploma that testifies the person's level in each skill). "*To become certified as a database management, for example, is one standard step on the path to becoming a database administrator*" [23].

8. CONCLUSIONS AND FUTURE WORK

This investigation presented the different knowledge and skills needed to become a computer scientist, more specifically system engineer. As an important part of a bigger research work, this research aims to reduce the negative consequences in the CS field due to the low number of students enrolling in this domain. Highlighting the misconception of the competencies in the CS program from the part of students, this study shows how the low number of students is affecting the development of CS and the European technology industry.

The study determined that mathematical logic, algorithms, programming, data bases, data modeling and operative systems represent the necessary

competencies to develop CS work. Other subjects as English and a good mathematical bases are to be taken into consideration. Some random profiles of job offers will confirm that companies are willing to find these competencies in the profile of people demanding a CS job.

The information presented in this paper lead us to ask if technology can provide some help that will enhance the teaching and learning methods of the needed skills to become a computer scientist. After this, a longer investigation will analyze how technology is contributing in the teaching and learning methods of these skills by providing dynamic learning software. The usage of this software can help a student losing the fear in front of logic mathematics; algorithms etc… and therefore attract him or her to follow CS studies in university, contributing to the development of the technology industry in European countries.

The idea of relating technology as a conduct to learn these skills is due to two reasons:

The first one is that the usage of technology can motivate a student to lose fear of technology by having personal experiences with computers.

The second one is that nowadays technology is being used to teach almost any domain in an effective and dynamic way [24]. As a source, the development of techniques such as Technology Enhanced Learning (TEL), representing teaching and learning methods with the usage of technology, are being developed by European communities in the name of education.

Other studies suggest the use of technological environments to learn CS and assume that the problem of low number of students enrolled in these studies can be controlled by creating interfaces and different kind of devises [25].

Initiatives are already aiming to attract and retain talent to CS by sharing technology experiences [12] and by creating teaching and learning methods to support students with the help of information technologies such as effective video clips for learning Web-languages (like JavaScript) [26]; or using 3D animation environments to teach introductory CS courses [11].

9. REFERENCES

1. Porta, M., Maillet, K. And Gil, M. “Dec-CS” The s Declining phenomenon. Lecture Notes in Engineering and Computer Science. Proceedings of The World Congress on Engineering and Computer Science 2010, WCECS 2010, 20-22 October, 2010, San Francisco, USA, pp1173-1178.
2. Maillet, K. and Porta, M. Consequences of the declining interest in computer science studies in Europe; IEEE Engineering Education, Educon Conference 2010. Madrid, Spain.
3. Karat J., Karat C.M. and Ukelson J. Motivation, and the Design of User Interfaces. Communications of the ACM 2010. Vol.43, No. 8 pp. 49 – 52.

4. Eurostat Official European Statistics (Research Report number 200-6.97) 2007. Retrieved from http://epp.eurostat.ec.europa.eu/portal/page/portal/eurostat/home/
5. Computer Science Teachers Association & Computer Research association. High School Computer Science Survey 2010. (Research Report No. 47B32). Retrived from CSTA Website http://csta.acm.org/Research/sub/CSTAResearch.html
6. Nagel, D. Computer Science Courses Show Steep Decline. Retrieved from THE Journal 2009. http://thejournal.com/articles/2009/09/09/in-brief.aspx
7. Computer Science Teachers Association (CSTA) and Computer Research association (CRA). 2010. High School Computer Science Survey.
8. Henderson, Hubert D. The theory of elasticity in price 1946. Hancourt Editorial.
9. Lohr, S. Study Plays Down Export of Computer Jobs. NYT Harvard Business Review 2006. Retrieved from http://www.nytimes.com/2006/02/23/technology/23outsource.html.
10. Bonk, C.J., and Reynolds, T. A Special Passage through Asia. Paper presented at the meetin of AACE conference 2009, Vancouver, Canada.
11. Williams A. Lessons in Mandarin Chinese-language school near reality. New York times August 12, 2009. pp.15.
12. Cussó R. et al. Sharing technology experiences of information from high school to university. International Review of Engineering Education 2009. Vol 2, pp 1173-1178.
13. Koppi T. et al. The crisis in ICT education: An academic perspective, Paper presentd at the meeting of Ascilite 2008. Melbourne.
14. OCDE. Evolution of the young interest for the scientific and technological studies – evolution 2006. http://www.oecd.org/home/0,3675,fr_2649_201185_1_1_1_1_1,00.html.
15. Gil, Marisa et a. Real projects to involve undergraduate students in CS degrees. Paper presented at the meeting of the Annual Global Engineering Education Conference 2006. Madrid, Spain.
16. Lewis A., Lichele H. and Waite W. Student and Faculty Attitudes and Beliefs about Computer Science. Contributed Articles, Communications of the ACM 2008. 53,5.
17. Oxford notes in computing, fourth edition. Library of Congress Cataloguing in Publication Data available 1996, ISBN 01928
18. Burali-Forti, C.A question on transfinite numbers, reprinted in van Heijenoort 1976, pp. 104–111
19. Blass, A. and Gurevich, Y. Algorithms: A Quest for Absolute Definitions. Bulletin of European Association for Theoretical Computer Science 81. Dickson 1998. Californi, USA.
20. Ling L. and Tamer M. Concepts and relations. Encyclopaedia of Database Systems 2009, pp.3- 60 ISBN 978-0-387-49616-0.
21. Wing, J. Computational thinking. Communications of the ACM Journal 2006. pp.43, 33–35.
22. Webster Bruce . Making IT work since 1974. Online report retrieved from Oxford notes in computing, fourth edition 2008. ISBN 01928
23. Computer Science, Technology and Database Administration Careers and Jobs, Bureau of labor statistics, U.S. Department of Labor 2010. Career overviews.
24. Windschitl M., and Andre T. Using computer simulations to enhance conceptual change: The roles of constructivist instruction and student epistemological beliefs. Journal of Research in Science Teaching 1998. Vol 35, pp 145–160.
25. Mantey, P.E. and Nolan, R. Computer Music to attract students to Computer science engineering. Presented at the Frontiers in Education confrence 1996, Salt Lake City, Utah, USA.
26. Kobayashi, T. and AL. Effective Video Clips for Web Language. Paper presented at the meetin of AACE conference 2009. Vancouver, Canada.