

The Information Architect: A Missing Link?

Ricardo Baeza-Yates	Miguel Nussbaum
Dept. of Computer Science	Dept. of Computer Science
Universidad de Chile	Catholic University
Santiago, Chile	Santiago, Chile
E-mail: rbaeza@dcc.uchile.cl	E-mail: mn@ing.puc.cl

Abstract

Motivated but many of the problems faced by computing professionals and what we teach, we propose a framework for designing curricula related to technology. In particular, by using this framework, we argue that a new professional, which we call the information architect, is needed. We propose a specific set of courses, which can also be justified in more standard computer science or information systems programs.

1. Introduction

Computing professionals today suffer many problems related to their education and background. The most important of them, in our opinion, are: over specialization, misunderstanding of the real world, and lack of communication abilities. One of the causes for these problems is what we teach to our students. Nor computer science neither information systems curricula addresses all these issues, and we believe that cosmetic changes to them will not solve our problems.

Although management and information systems (MIS) curricula is closer in spirit to what might be needed, seems that the differences with computer science are more cultural than technical [IEEE97,IEEE98]. Universities do not only have problems at the content level, but also on how education is carried out. Essentially, little has changed from the traditional European universities of 500 years ago. Nowadays, it is much harder to avoid competition or to not care about what society thinks or expect [FRE97,WEI98,TSI99].

Our main contribution is a simple framework which is implicit in some previous work, but not yet formalized. It is based on three components: people, processes, and technology. Using this framework we propose a curriculum for a new professional: the information architect. This architect should be the bridge between the components of our model. That is, should understand the interactions between them. In some sense, it is analogous to the normal architect, which is the bridge between how people want to live and the technology upon buildings or houses are built.

This paper is organized as follows. Section 2 presents the main problems faced by computing professionals, which are the motivation for the new curricula proposed. Section 3 introduces our framework and gives some examples. In section 4, we argue the need for an information architect and we design and propose an appropriate curricula. The last section presents our concluding remarks and ongoing work.

2. The Problems that We Face

2.1 Over Specialization

A couple of centuries ago, a wise person was the one who managed a great amount of knowledge: we only have to remember da Vinci or Newton. Today, in turn, wisdom is synonymous of specialization, to such an extent that when one person is sick, it is necessary a meeting of super specialists to make a decision. It does not exist a global vision to understand a problem as a whole and look for alternative solutions within different dominions. Students in Computer Science are not foreign to this reality. We prepare them to be

specialists on a very specific domain: to develop software. This type of specialization is fine for writing programs. However, this is just part of the game. The other side is the understanding of reality, i.e., its actors, their relationships and its environments. When systems are developed without a sound and complete understanding of reality, the resulting product not necessarily meets users requirements and expectations. A bottom up approach is technology driven and not problem oriented. History has shown us that this approach, which has been used in the first 50 years of computing history, has not developed what people expected, and we are even loosing the managers confidence. In addition, some of our implementation designs have been transferred to the end user. For example, conceptually, there is no need to make a difference between a file system and a database.

Our professionals have a specialized preparation on a narrow scope, the development of software tools. A complement is missing, the top down view. Although with object oriented programming, a top down design is more natural, that does not mean that we understand the problem to be solved. We need to complement the dominion knowledge with a global vision of the problem and provide the capacity of knowing how to be specialists on topics related to their work. It is harder to obtain the global vision than the knowledge about a topic, a methodology or a specific tool.

2.2 Science vs. Tools

If we analyze what we understand by science we can observe two lines. One, that studies the nature of elements from different points of view: physics, chemistry, biology. The other one studies tools developed by man: mathematics and computer science. The first line focuses its questions in order to understand the elements that form the universe: material, processes, life. The second line has defined close, consistent and complete worlds, that have been used to understand phenomena that the first one studies. While in the first line any question we make opens us the universe doors, in the second one any question can or can not be a new tool for helping us in the first one. When this is not the case, the second one becomes theoretical science, like people who study chess, which is also a close world with well defined principles.

Understanding and manipulating a consistent close world is undoubtedly very formative. However it is not sufficient for understanding the real world. For example, do we have something to say to society when trying to solve chess games? A concrete example is what is happening with Operations Research in the United States [HOR92]. The employment rate of these professionals has decreased in a 20% during the years 90 and 92, attributed to the little practice these professionals have. In computer science it is not happening yet, but we all known that we have lost terrain in our influence in corporate decisions over this 50 years.

Students, however, are showing a decay of motivation in the field. In one of the two major Engineering Schools of our country, this image deterioration effect can be clearly observed. From 29% of all Engineering students that graduated with a major in Computers Science in 1992, this percentage dropped to 22% in 1993, and 18% in 1994. A vacuum is left of trained professionals for performing information systems work. This space is filled by engineers of other disciplines. 59% of the students, of the same engineering school in the same time range, chose their first work in jobs related to computing even their training was in other disciplines, mainly Electrical, Chemical and Mechanical Engineering. We, as computer science departments, are not completely fulfilling our social mission. Society has a real problem. Computers are almost free and globalization has increased the complexity of the world in such a way that it is impossible to handle almost any job without computer support. In addition, the professionals that link machines and this increasingly complex world are not there, because we are not training them (adequately) [LBM98].

2.3 Insertion in the Real World

Computer technology changes fast. However, change does not always mean improvement, just means a change. Also, the dominant technology is not necessarily the best possible neither the first [NOR98]. In particular, most dominant software today was never planned to be what it is today [REE98]. These are

the main problems to obtain better engineering skills and to generate maturity in our field. Many programmers still use trial and error, which also does not help to improve software development methodologies. Bad design generates error-prone software and people has accepted that software will always have bugs [LEW98]. In addition, real life challenges are harder each day (for example, the interrelations and simultaneity of the Y2K problem or large multidisciplinary projects) and data volume grows exponentially (the best example is the Web). So, the computing professional has to adapt to all these facts. On the other hand, industry always complains that the university do not teach what they really need. That is not strange, as the goals of universities have a time-horizon much larger than industry. The technological gap between the university and the industry is indispensable, but each day is more difficult to achieve. Our students should ultimately be the agents of change (the good ones!) [GGT97], and some even think that we should be really involved in the future of information technology instead of just following a crazy market [REE98].

The lack of communication abilities and hierarchical work also affects the design process. In many projects all the people participates in the design phase or suggest changes. Although this might be beneficial in some cases, it does not work in general. Doing again a parallelism with building houses, the designers are usually few (the architects) and the civil engineers follow most of the specifications, while the final construction is done without changes. Note also that the architect does not have any problem in understanding the requirements of the final users. In our field, many times the language is a barrier and there is no real counterpart to the classical architect between the two parts involved in a software project.

3. A Simple Framework

From the previous sections we conclude that the professional profile we want to obtain is based in three main elements: technology, people, and processes. By technology we understand mainly information and communication principles. People should be understood as social sciences. Finally processes, as the transformation sequence that achieves a given goal by using technology and/or people. Each element is sustained by its own and by their relation with the other two elements. Figure 1 represents this three elements and their interaction. This model is dual in the sense that we can define university as the relation of society with technology, industry as the relation of technology and industry, and the market as the relation between processes and people.

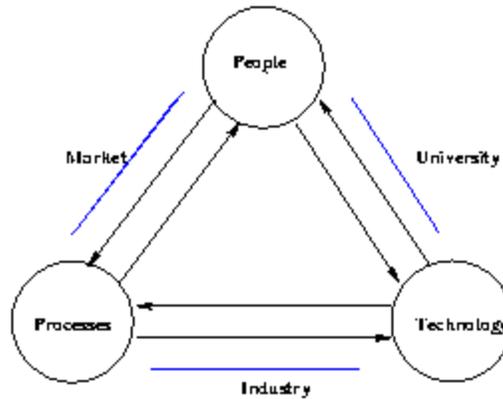


Figure 1. Our dual model based on people, processes, and technology.

We can map in this framework the current computer science curricula. It is focused in technology and the interaction of technology with the other elements. We can see that the reverse interactions are missing, which partly explains the lack of knowledge and language differences between the user and the designer. Therefore, we can use this framework to complement and improve the existing curricula, or to design a new curricula. Although our framework is generic, we now focus on a specific example, which complements computer science and fills the gap between the real world and the specialists. Although we could say that there is an optimal curriculum that fulfills all the needs of the real world, there are two

problems: the time needed to cover well all the contents exceeds the length of any reasonable curricula or the coverage would be too shallow. Figure 2 shows how CS, MIS or sociology would fit in this model.

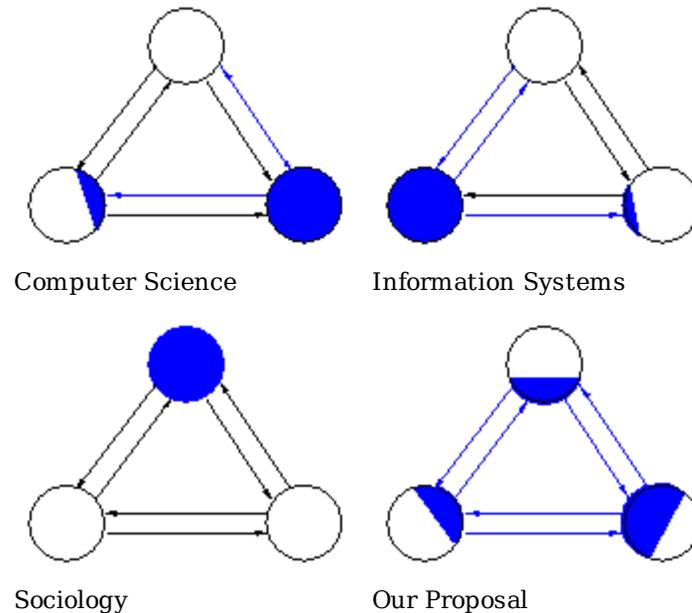


Figure 2. Some professionals in our framework.

From these examples we can see that in all of them we miss part of the interactions between these elements. The proposal in the next section tries to partially solve this problem.

4. The Information Architect

Our proposed professional is also a specialist, but on interactions (also shown in Figure 2). Interactions allow to understand the global picture and are mainly based in extracting and relating information. This professional as an architect, has to be able to understand users requirements and map them to another dimension, in our case, systems. For this reason, our (information) architect is a high level designer that uses information as building material.

The proposed curricula is analyzed following the structure defined in Figure 1. First the three pillars are discussed (processes, technology and people) and then the interrelating elements are analyzed from one pillar to the other which allows situating the viewer in a given perspective. For example the bridge from People to Technology has been chosen as Design and Cognitive Ergonomics which gives a psychological and sociological perspective of the human relation to machines, while vice versa, i.e., the bridge from Technology to People is studied in Interface Design, where the same problem area is analyzed from a Computer Science perspective.

Processes: to be able to understand the world of business, where our professionals will finally work, we have to provide them the frame, the knowledge and the available tools to understand and solve the problems they will find. Three topics have been selected for achieving this purpose: System Modeling, Optimization, and Project Evaluation and Control.

System Modeling: formal description of human process interaction through an abstract description: a model. Different models can be introduced as game theory, human behavior, control theory, etc. The aim is to provide the student the capability to see the world through abstractions. This includes knowledge representation.

Optimization: once the student understands the world in abstract terms it can be optimized.

Mathematical language is used to construct abstract models from real optimization problems. Emphasis should be given to modeling more than the algorithms behind the mathematical procedures since we are looking for professionals that have to specify solutions using the available tools. Standard heuristic solutions as tabu search, simulated annealing, etc. have to be studied for complex problems.

Project Evaluation and Control: the evaluation of viability and profitability of projects is a key issue for any professional related to the business world. Management tools and financial concepts have to be considered. Different control procedures have to be studied as PERT, CPM, Gantt charts, etc. Software productivity as a problem of project viability and project control has to be raised, as well as all the risks involved.

The relation of Processes and Technology is studied in Operations Management. How technology is used to support the organizations requirements of logistics, transport, production, stock control, maintenance and quality control is shown. The underlying models and processes have to be studied.

A Management course is the bridge between Processes and People: to understand the business world where the processes occur it is necessary to consider the way corporations and markets organize and behave. Planning and control strategies are studied showing how information systems are the main tool. This course should also include notions of economy, accounting, and auditing.

People have the problem, want the solutions and decide if the systems satisfy their expectations and/or requirements. If we cannot communicate with people they will not get our message and we will not obtain the information we require for building our systems. If we do not understand how people behave individually and in groups we will not construct systems that perform as expected. Three topics have been selected for achieving this purpose: Cognitive Psychology, Sociologic Paradigms and Communication Aspects.

Cognitive Psychology: the student should have the basic elements on how we think and reason. Human information processing is analyzed by studying feeling, perception, attention, memory, reasoning and operability. This allows first, to understand the user as an individual that has some requirements that have to be understood and second, to have a human model that can be used as test bed for the systems' human interface requirements.

Sociologic Paradigms: since our professional will highly interact with groups of people, to study different perspectives of the social phenomenon allows to better understand human behavior. Action, system and communication theory are tools to better understand group dynamics, interpersonal perceptions and social representations. This includes the development of leadership and collaborative work skills.

Communication Aspects: engineers are well known for their communication deficiencies. Since our professional has to be a specialist in interactions he has to master interpersonal communication. Aspects as formal and informal communication and the interrelation between communication, culture and language have to be known. Not only spoken abilities have to be studied but also written ones since reports are his main tool. Between the spoken abilities, negotiation skills are very important.

The relation of People with Processes is studied in Legal and Ethical Issues. The student has to comprehend the written and unwritten protocol of behavior, i.e., law and ethics, as a professional in his reality. This issue has come to light with the Y2K problem on one side and viruses problem on the other. However this issues are much deeper than these two topics.

The bridge between People and Technology is studied in Design and Cognitive Ergonomics. Our professional should be able to evaluate working environments (think just on the banking teller machine), from its physical and mental requirements. Human reliability and capacity have to be understood for designing usable systems.

Our last pillar is **Technology**. As we are interested in developing better software systems, a bias to technology is given. Each pillar was formed by three courses, but this one is defined by double as much since for us these is what we understand are the basic topics that our professional should manage. The

six courses we propose are Data Structures and Complexity of Algorithms, Computer Architecture, Operating Systems and Distributed Systems, Software Engineering, and Data Bases. The bridge from technology to people is given by Interface Design, while the bridge from technology to processes is given by Information Systems. We are not going to discuss in detail these courses since their contents are well known in the field.

5. Concluding Remarks

The communication gap between the user and the computer specialist, that leads to a misunderstanding of requirements, is one of the key issues of software project risks [CLS98]. For example, while the first sees that he has not received the requested information for deciding which items to buy and sell and how much money he has available now and next week, the second claims that the files were incompatible, registers were missing and the data base was inconsistent. Narrowing the gap with a professional that understands both worlds is a necessity today.

The general model shown in Figure 1 where a curriculum is based on three main pillars, technology, processes and people, is not restrained to computer science. Any professional related to technology works in a multidisciplinary world from where he has to obtain the specification of his system. and therefore has to interact with people. Complementary, any engineering system undergoes a modeling processes to understand its feasibility and economic viability.

Software liability [VMKV97] has been an issue which is risen specially with the Y2K problem. We are preparing professionals sitting on a liability time bomb. It is difficult to determine who is accountable when a piece of software does not meet expectation or fails in certain conditions. Projects tend to grow out of control since at the specification stage sometimes we just see the tip of the iceberg, occasionally finishing in lawsuits. Studies questions the contribution of information technology to productivity [BH98,DK98]. Complementary, technological changes are dramatic while organizations have a different dynamic. We have to prepare professionals that not only understand the problem, the organization and their responsibilities in the process, but also the organization capacities of assimilating and managing technology.

Most curricula proposals are focused in technology and contents, and not much in methodology, as ours. We can mention the ACM-IEEE curricula [ACM91] (the next one is planned for 2001), and the UNESCO-IFIP proposals [UNE94] for computer science, and the ACM-AIS-AITP recent proposal for MIS [ACM97]. That is, there are many proposals for the content (what), but not really for teaching (how). As computer scientists, education is not one of our strengths. Although we do not propose any particular methodology on how to teach it. Its multidisciplinary approach at a macro level should be not forgotten at a micro level. Each course should not be seen as a close world, with almost no relation to other courses. Over specialization and encyclopedic teaching hinder constructive thinking. Our professionals should be able to change context easily and understand the reality they are confronted in the terms that technology has provided us. On the other hand, some of the courses proposed here could complement well some existent CS or MIS curricula on the weaknesses that our careers have.

The methodological problems do not belong particularly to CS. The way that we teach has not changed much in the latest centuries, in spite of having new tools and media. However, our field is less static than other fields, in particular mathematics or physics. As a subproduct of specialization, the original relations (either functional, historical, etc.) of different areas are omitted. Each course is a black box with little communication to other courses. Sometimes even the same topic is covered with a different point of view in different courses. The final result is that integrating different pieces of knowledge is more difficult. Another effect of this black box problem, is that students need to do many context switches (by using a CS analogy) between different topics and views, without a real motivation for them. Real integration of topics should help on motivating and giving the students the reasons of why they should learn A and/or B. To reinforce this, education should be more interactive, emphasizing participation and cooperative work.

References

- [ACM91] ACM-IEEE Curricula Recommendations for Computer Science, volume I, <http://www.acm.org/education>, 1991.
- [ACM97] ACM/AIS/AITP Curriculum for Undergraduate Degree Programs in Information Systems, <http://www.acm.org/education>, 1997.
- [BH98] Erik Brynjolfsson and Lorin M. Hitt, Beyond The Productivity Paradox, Communications of The ACM, volume 41:8, August 1998, 49-55.
- [CLS98] P. Cule, K. Lyytinen, and R. Schmidt, A Framework for Identifying Software Project Risks, Communications of the ACM 41, November 98, 76-83.
- [DK98] Sanjeev Dewan and Kenneth L. Kraemer, International Dimensions of the Productivity Paradox, Communications of the ACM 41(8), August 1998, 56-62.
- [FRE97] Peter Freeman, Elements of Effective Computation, IEEE Computer, November, 1997.
- [GGT97] David Garlan, David P. Gluch and James E. Tomayko, Agents of Change: Educating Software Engineering Leaders, IEEE Computer 30(11), November 1997, 59-65.
- [HOR92] Peter Horner, Where do we go from here?, OR/MS Today, April 1992, 20-27.
- [IEEE97] IEEE Special Issue, Status of Software Engineering Education and Training, IEEE Software, Nov/Dic 1997.
- [IEEE98] IEEE Special Issue on the Future of Computing and Software Engineering, IEEE Computer, January 1998.
- [LEW98] Ted Lewis, ``Joe Sixpack, Larry Lemming and Ralph Nader'', IEEE Computer, July 1998, 107-109.
- [LBM98] D. Lowther, T. Bassoppo-Moyo, G. Morrison Moving from computer literate to technologically competent: the next educational reform, Computers in Human Behavior 14 (1), 93-109.
- [NOR98] Donald A. Norman, The Invisible Computer, MIT Press, 1998.
- [REE98] Karl Reed, Why the CS should help chart the Future of IT. IEEE Computer, July 1998, 77-78.
- [TSI99] Dennis Tsichritzis, Reengineering the University, Comm. of the ACM 42(6), June 1999, 91-100.
- [UNE94] UNESCO-IFIP, A Modular Curriculum in Computer Science, UNESCO Publications, 1994.
- [VMKV97] J. Voas, G. McGraw, L. Kassab, and L. Voas, A "Crystal Ball" for Liability, IEEE Computer 30, June 97, 29-39.
- [WEI98] Mark Weisz, Dilbert University, IEEE Software, Sept./Oct. 1998, 18-22.