Computer Science's Digest

Volume 3

Jesús Insuasti, Manuel Bolaños, Nelson Jaramillo

Authors: Jesús Insuasti. Manuel Bolaños. Nelson Jaramillo

University of Nariño. Systems Department. Galeras.NET Research Group

Computer Science's Digest. Volume 3.

ISBN (volume): 978-958-8958-35-4

Number de edition: 1

Cover design: Cover Art by Pixabay. Creative Commons License.

Proof reading: Katrin Riddle

Layout, printing and finishing: Editorial Universitaria - University of Nariño

Printed and made in Pasto, Colombia.

Insuasti, Jesús
Computer science's digest / Jesús Insuasti, Manuel Bolaños, Nelson Jaramillo. – 1ª. ed. –
San Juan de Pasto : Editorial Universitaria : Universidad de Nariño, 2017.
v. : il. gráficos. tablas

Incluye bibliografía
Contiene : v.3
Índice de contenidos: v.3. Professional issues in computing ; Programming the internet--

ISBN: 978-958-8958-35-4

1. Seguridad informática 2. Delitos por computador 3. Cifrado de datos
I. Bolaños, Manuel II. Jaramillo, Nelson

004 I599 - CDD- Ed. 21 Biblioteca Alberto Quijano Guerrero

About the Authors:

Jesus Insuasti, MSc.

Jesus Insuasti is an Assistant Professor in the Systems Department at University of Nariño. His specialties include software design and implementation, embedded systems and game development, amongst others. He especially enjoys working with Microsoft technology. His academic career is as follows: B.S. Systems Engineer (University of Nariño - 1999), MSc. in Higher Education Teaching (University of Nariño - 2010), English Proficiency (San Jose State University - Silicon Valley - 2010), MSc. in Internet Systems (The University of Liverpool - 2014). Currently, Insuasti is finishing his PhD on Education in Colombia.

Manuel Bolaños, MSc.

Manuel Ernesto Bolaños González is the Director of the Systems Department at the University of Nariño. He has a DEA in Computer Engineering from the Pontifical University of Salamanca, an MA in Information Sciences and Communication from the District University (Colombia), and he specialized in Systems Auditing, Higher Education Teaching and Systems Engineering. He has been a full-time professor at the University of Nariño since 1999.

Nelson Jaramillo, MSc.

Nelson Antonio Jaramillo Enriquez qualified as a Systems Engineer at the University of Antioquia, specialized in Systems Auditing at Antonio Nariño University and gained a Masters in Educational Communication at the Technological University of Pereira. He is Dean of the Engineering Faculty at the University of Nariño, and has been a full-time professor at the department since 1997.

Acknowledgment

The authors want to thank the Systems Department at the University of Nariño for their support in the creation of this series of textbooks. This series is dedicated to the students of the Systems Department, to give them reading material related to computer science in a second language. This book covers Professional Issues in Computing and Programming the Internet.

Preface

This series of textbooks was created for the students of the Systems Engineering Program at the University of Nariño. They have been intentionally written in English to promote reading in a foreign language. The textbooks are a collection of reflections and workshops on specific situations in the field of computer science, based on the authors' experiences.

The main purpose of these textbooks is essentially academic. The way in which the reflections and workshops were constructed follows a didactic structure, to facilitate teaching and learning, making use of English as a second language.

The Authors.

CONTENTS

1. PROFESSIONAL ISSUES IN COMPUTING 8
1.1. REFLECTIONS 8
1.1.1. Internet Governance 8
1.1.2. Projects Gone Wrong 10
1.1.3. The Case for Freedom of Speech Online 11
1.1.4. Data Protection 14
1.1.5. Intellectual Property: A Case Study 16
1.1.6. The World and Computers 17
1.1.7. Outsourcing in the workplace 20
1.1.8. Data Ownership 21
1.1.9. Privacy and Data Protection in Colombia 23
1.1.10. Professional Responsibility 25
1.1.11. Future Computing Professions 27
1.1.12. Pros and Cons of New Ways of Communication 28
1.1.13. ROI in the workplace 30
1.1.14. Software Risks in the Workplace 32
1.1.15. Intellectual Property 33
1.1.16. Why does Risk Management fail? 35
1.2. WORKSHOPS 37
1.2.1. Hand-In Assignment – Seminar 1 37
1.2.2. Hand-In Assignment – Seminar 2 41
1.2.3. Professional Ethics: Legal Transgression on the part of the Customer 44
1.2.4. Protecting Intellectual Property: Software 47
1.2.5. Hand-in Assignment – Seminar 5 49
1.2.6. Hand-In Assignment – Seminar 6 52
1.2.7. Hand-In Assignment – Seminar 7 55
2. PROGRAMMING THE INTERNET 60
2.1. REFLECTIONS 60

2.1.1. The "Deep Web" 60
2.1.2. Analysis of Protecting Code in Client-Side Scripting 61
2.1.3. **Data Warehousing Online** 63
2.1.4. DOM 65
2.1.5. Threats to JavaScript 68
2.1.6. The Future of Open Computing 69
2.1.7. HTML: Past, Present and Future 71
2.1.8. JavaScript and Interpreted Code 72
2.1.9. XML Web Services 75
2.1.10. Design Patterns 77
2.1.11. Objects on Web Pages: A JavaScript Perspective 78
2.1.12. AJAX and SOA 80
2.1.13. Using Cookies via PHP 81
2.1.14. Web-based Applications: AJAX Model 82
2.1.15. Web Forms and Workflows 84
2.2. WORKSHOPS 86
2.2.1. 1st Draft of a Proposal: Web-Based Academic Grading System 86
2.2.2. Advanced PHP and Database Issues 90
BIBLIOGRAPHY 109

1. PROFESSIONAL ISSUES IN COMPUTING

1.1. REFLECTIONS

1.1.1. Internet Governance

In the models of Internet Governance as stated by Lawrence B. Solum, the second model defines the internet as a global arena, in which transnational institutions and international organizations establish the guidelines and regulations (2008). The Internet is borderless by its nature, completely globalized, and therefore the transnational institutions and international organizations, which are responsible for regulating the Internet, should engage in close relationships with governments from around the world.

Arguably, this second model could be implemented more easily than Solum's other proposals for three main reasons. Firstly, the current regulations surrounding the Internet's infrastructure; secondly, the existence of large organizations that control data protection, privacy and human rights; and finally, governments' own development plans concerning ICT. These will be given in more detail as follows.

1. The current Internet infrastructure is controlled by international organizations such as IANA and IETF (Adams &McCrindle, 2008: 182). These institutions assign blocks of IP addresses to networks and watch the Internet protocol updates respectively. Thanks to the development of these institutions, the Internet's resources, sites and users have all grown rapidly, which is testimony to their scalability and high performance.

2. Organizations that watch for data protection, privacy and human rights are making enormous efforts to detect violations of fundamental principles in different contexts. These organizations include WIPO, FTC, EUDPD, Amnesty International, and Human Rights Watch, among others. Technology is part of human life; and the Internet, as a means of carrying out this kind surveillance, is powerful.

3. Given the evolving lifestyle of societies all around the world, it is unlikely that any current government does not have an ICT budget. According to Zheng (2009: 66), an important indicator of growth in terms of social development is the level of implementation of ICT. Each government therefore, has plans for ICT development, which are closely related to Internet access. This is the perfect scenario in which international organizations and governments could begin to establish agreements in order to promote wider Internet usage under certain parameters.

Taking the above into account, Solum's second model is the most feasible of all the models when it comes to practical application. However, there are several issues which remain untouched upon, or are not explored in depth; for instance, the regulation of free speech and the Internet content regulation. It could be that action on the part of governments and global organizations would go against human rights (Nair, 2007: 17-19). The ideal would be to establish minimum agreements between the parties involved to create rules that control activities without violating rights (although this sounds a bit utopian). That is why is so difficult to create a *status quo* for Internet Governance.

1.1.2. Projects Gone Wrong

Finding the balance between liberty and safety in society is a time old struggle. The conflict between individuals needing freedom and privacy, without jeopardizing other's right to act as they want, and the concern that privacy for the individual sacrifices safety on the part of the whole. However, those two concepts are not isolated ideas (Zipkin, 2006: 324). The right of privacy is known world-wide (but unfortunately not always respected) and stated in Universal Declaration of Human Rights. In Article 12, it states that nobody can interfere with another person's privacy (UN General Assembly, 1948), which means that every human being has the right to privacy and it this right can be exercised according to the will of the individual.

In spite of the advantages of RFID, one of the biggest concerns about this technology lies in privacy. An RFID-based application can store personal information in tags, and its mobility feature allows the tracking of items (CPSR, 2005). This could put people's privacy at risk, by revealing their location and personal data. In the same way, some technical risks have been noted when using RFID technology for security purposes, research in the field of RFID-based system communication revealed some vulnerability (Paraschiv, Melinte, &Pricop, 2009: 89). A common attack which occurs with this kind of technology is sniffing, which means that an intruder (an RFID reader) can access the information stored in RFID tags. A common way of hacking RFID systems is by cloning the transponders, compromising the system's integrity. In order to prevent these kind of attacks, tag authentication and data encryption must be configured.

Another way of attacking RFID-based systems is through buffer overflows, code insertion and SQL-injection. In buffer overflows, free access to the unprotected parts of memory is an open

door for hackers. This kind of attack leads to system collapse, due to unauthorized manipulation of the data stored in memory in real time. When the information in a RFID card is written using scripting languages such as JavaScript, hackers are able to write malicious code on the tag. The same situation can occur when hackers inject SQL code, in order to see the inner structures of databases (Paraschiv, Melinte, &Pricop, 2009: 90).

This risk causes a huge dilemma in supply chains. By installing one RFID tag into every single product, all of a customer's private information could be accessed by those who have access to RFID readers. Every single product bought by customers could be traced and this would mean that the location of each item is known by the producers of those products, which is a very serious threat to people's privacy.

1.1.3. The Case for Freedom of Speech Online

In some regions where censorship is extremely strict, those who exercise their freedom of speech can be held in prison. International freedom and human rights movements call these cases prisoners of conscience (Amnesty International, n.d.). In such cases, prisoners have been deprived of their liberty due to their beliefs, political opinions, religion, ethics, ethnic origin, sexual orientation, or social class, among other factors. These situations, in which the right to free speech has been violated, all have one thing in common. That is, that the entity that promotes certain extreme censorship mechanisms, usually governments, are taking refuge in the excuse of "national security" to justify such acts of censorship.

In Latin America, some countries have imposed serious restrictions on people's freedom of speech, in Venezuela and Cuba for example. More specifically in Cuba, which is the largest

island in the Caribbean and the only Latin American country under a socialist regime with Republican status, which it is has had since the Cuban Revolution started in 1953. Governmental social policies are all aimed at achieving the ends of the Revolution, and all the conditions or restrictions placed on peoples' lives are to guarantee social welfare (Sixth Congress of the Communist Party of Cuba, 2011: 53).

In spite of the improvements in the fields of education, access to medical care, culture, and sports, among others, the Cuban regime has a very hardline policy when it comes to accessing information. Like the governments, in China and some Arab states, direct criticism of the government's policy, in this case the Socialist Regime of the Revolution, is considered willful social disorder and a breach of the peace. That is why people who criticize the Cuban government publish their thoughts online from outside the island (especially via blogs). In response to this, the regime censors these publications, so people living on the island have no access to the information (Corsa, 2008: 33).

One of the most famous cases is that of the "Generación Y" blog (in English "Generation Y" which refers to Cubans born in 70's, when Russian-style names were widely used). Unlike other blogs that spoke out against the Cuban Government whose authors lived outside the island, Mrs. Sánchez who writes "Generación Y" lives in La Habana. In order to update the blog, she has to pose as a tourist in Internet cafes around La Havana, and thus manages the content of her system hosted on a server in Germany (Corsa, 2008: 34). This blog emphasizes in the utopian status of the Cuban socialist regime. According to its author, the blog system allows her to say what is forbidden her in real life (Sánchez, 2011).

In essence, “Generación Y” promotes freedom of speech and talks about the quality of life in Cuba. In addition, an important function of the blog is to ask for support or donations in order to maintain and grow it. It asks people entering the island as tourists to help, to bring items which support the technological infrastructure of the blog, for example: USBs, hard drives, digital cameras and even laptops are all useful in continuing the development of the blog. Furthermore, the author has no internet access for long periods of time, so she encourages people to donate access cards for her to use the internet in public places. She says that Cubans have to pay high prices to get internet access in cybercafés or hotels. She also tries to get digital recordings of political prisoners giving their opinions about the revolution.

People who feel oppressed by their governments seek to assert their fundamental right to freedom of speech, enshrined in the Universal Declaration of Human Rights. According to article 19, people should be able to express their opinions freely without any kind of interference (United Nations: 1949). Unfortunately, this right is denied (or is hindered) in some places. International organizations, such as Amnesty International, are making great efforts to uphold these fundamental rights, but so far this has not been enough as they have not been able to change several governments’ policies regarding free speech. Those people who expose the reality of their living situation and give evidence must be supported and heard as much as possible.

1.1.4. Data Protection

Data protection, or the safeguarding of information, has always been a concern for individuals and societies, even before the computer age. In second half of the 20th century, developments in computing have enabled the creation of strategies and mechanisms which safeguard the information stored on computers.

These computing solutions are based on algorithms specifically designed to protect data; and there are several techniques involved, such as cryptography and steganography (Adam, McCrindle, 2008: 212). Nowadays, several professional computer-based solutions are available on the market (open source and proprietary) to deal with the complex task of protecting data, to name but a few: communication protocols, encryption algorithms, encoding techniques, and user applications.

According to Schartum, the current situation as regards data protection (or at least that of most places) is a complex one, in which 3 main factors influence the development of technology and its application in society (2010: 7). The relationships between these three factors is shown below:

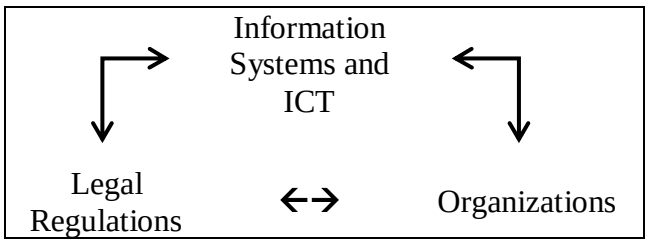


Figure 1.1 Factors which effect the development of technology

The interaction between these 3 factors influences how data protection is managed, and this varies from society to society. As individuals of a community, each person has personal data to protect, and some communities have established their own legislation concerning personal data. Such is the case of European Union and the Charter of Fundamental Rights; personal data is defined therein as social, economic, or academic information related to an individual, or information relating to their daily life (European Commission, n.d.: 3).

On the other hand, at an international level, committees such as the Organization for Economic Co-operation and Development (OECD), designed international guidelines called The Transborder Flow of Personal Data. In this document, the OECD outlined a set of weaknesses associated with "current" technology, although the proposal was made 30 years ago. Some authors argue that these guidelines should be updated to fit current individual legislation in each country, and need to address the difficulty of reaching a minimum requirement between countries (Wright, De Hert, Gutwirth, 2011: 119).

The current global Data Protection situation does face problems. Hypothetically, if we are unable to establish minimum requirements to protect our own data, data protection is left up to the open market exclusively, and the private sector would be wholly responsible for safeguarding personal data. In this scenario, those engaging in any online activity requiring data protection would have to sign up for a Data Protection service with a specific company. However, in such an eventuality there would probably exist a very full range of offers and contracts on the market (because "market" does not mean only one company). In this vein, it is also possible that some offers would be published as open-source solutions.

One way or another, a situation in which the market exclusively governs the protection of data, would make payment on the part of individuals and companies inevitable, even if some companies offer their services via open-source solutions. In addition, it would probably be available for a service fee, but in any case, people would have to pay for Data Protection.

1.1.5. Intellectual Property: A Case Study

The definition of Intellectual Property (IP) according to Adams and McCrindle (2008: 402), is a collection of regulations such as patents, trade secrets, trademarks, and copyrights, among others, which grant the ownership of a specific product to a person or company. This particular property case relates to a specific software product which was the result of time and intellectual efforts on the part of one person.

All jobs related to software construction must be very clear about the terms of ownership in a contract. In this case, with Paul Software, it is essential to know the terms of its contract (which were not available in the proposed problem).

Software reusability on the other hand is comprehensively explained in several documents as a software engineering methodology. Reusability is matter of domain engineering and the ins and outs are covered by the IEEE Computer Society. One of the standards developed by IEEE Computer Society, number 1517, outlines the relevant aspects to be considered in the development of the software life cycle from a reuse-oriented approach (IEEE-CS, 2010: 1).

According to the above document, a *reuse asset* can be developed in order to integrate the computer-based solutions. However, the main issue is related to the ownership of that software

asset. It depends of the contract terms between NewLeaf, The Sofa Barn, and Paul. NewLeaf develops custom software which implies the desire to have ownership on the client side. However, there are rights on both sides concerning the process of construction, and those rights should be shared between Paul and NewLeaf by establishing which parts belong to whom.

1.1.6. The World and Computers

The term computer is actually far older than people imagine. People often associate it with the technological revolution of the past five or six decades. However, the etymology of the word computer goes back further; it comes from the verb to compute meaning to solve. Ergo, a computer is a machine which is able to calculate or solve problems through calculations, and as such, there is a controversy surrounding the creation of the first computer.

There are those who claim the first computer was the abacus. However, as it cannot make calculations by itself, an abacus always needs human assistance, it is just a tool that facilitates working with numbers. Computer scientists state that the first computer was conceived by Charles Babbage in 1837, and he called it "The Analytical Engine" (Green, 2005: 35). Although Babbage was unable to build his machine due to financial problems, it is widely considered the first computer design because it was implicitly automated, which means it would have been able to make calculations by itself (Savage, 2010: 18).

Computers could therefore be said to have been present for more than a century in human life, and it stands to reason that these machines have impacted several (it could be argued almost all) aspects of society. Nowadays, computers are used in many different contexts, and the majority

of workplaces use computers to manage information. Computers are everywhere and we interact with them constantly.

Assumptions are made that if a technological catastrophe hit, society would survive because human kind lived 200,000 years without computers. This notwithstanding, there are many contexts in which society depends on computerized assistance. For example, banking systems, money transactions, and real-time communications would collapse without computers, which would create utter chaos for the economy. In this hypothetical situation in which a technological apocalypse occurs, our planet would probably return to a lifestyle similar to the eighteenth century. Every company, institution or organization that manages information with computers might have serious problems trying to keep working without them. In addition, modern vehicles with embedded computers which control various essential systems such as brakes, fuel consumption, and security would surely be a threat to people's safety if their computers failed. Almost all electronic devices that support the lives of patients in hospitals depend on computers; many people's lives would be at risk if these machines stop working. In this fictitious, but not impossible situation (USA Today, 2010), the list of problems caused by the absence of computers would be almost endless in this modern world.

It is extremely difficult to identify who the current direct beneficiaries of computers are, because many so people interact with computers daily at work and in their private lives. As such, many sectors benefit from the use of computer-based systems: government, economy, science, food production, trade, banking, communications, education, entertainment, transportation, health, military, security, and so on, and of course, there are the benefits to individuals and their private lives.

Our digital age has brought with it a reduction in paper consumption, as a lot of paperwork is now in digital format instead of in print. Although computers have helped in some way to reduce paper use, electronic waste (e-waste) has become a serious global environmental problem. Some of the components manufactured for computers are highly toxic. Oswald and Reller explain how deep the environmental impact is on developing countries in Asia and Africa, as developed countries contract out the processes which cause e-waste overseas. It has been said that taking technology production to developing countries closes the digital divide between developed and developing countries. However, the environmental effects of technology production on the landscape, and the inhuman conditions in workplaces and unacceptably low wages paid to the locals surely outweigh these benefits (2011: 46).

Another negative impact of the rise of computer-based systems is the loss of manual labor jobs. Countless jobs have disappeared as they have been replaced by computer-based systems. In spite of the increase in new employment opportunities in computing, many manual labor posts have been replaced. For example, evidently nowadays there are fewer personnel working in postal and courier services due to the massive rise in email-based systems. Similarly, the telecommunications sector has changes, professions such as telephone commutator, telegrapher, orthographic reviewer, etc. have all but disappeared. Computers have also led to the automation of industrial processes, applications in robotics have caused massive layoffs at production line factories.

1.1.7. Outsourcing in the workplace

The University of Nariño is a public higher education institution. It was founded in 1904 in Pasto, southern Colombia (Álvarez& Guerrero, n.d.). Nowadays, the University of Nariño has a small uptake of just over 8,000 students. It has 34 academic departments whose programs are offered as certificates, undergraduate, and graduate (master and PhD) degrees. As it is public it is a nonprofit institution.

Currently, the authors are professors within the Systems Department, and as such are responsible for courses in computer programming and software design. The department staff have also actively participated in the construction and maintenance of the university software platform by building custom programs. This has led to cooperation between students and professors to meet the challenges of the construction of the custom software required by the university. At the end of the project, the people involved were employed formally by the university and are now part of the information technology services' staff.

In terms of Information Systems, the university has never outsourced. It is a practice commonly adopted by public universities in Colombia, building software from within the organization (using university staff) helps to reduce high costs of investment in information systems.

In other areas such as ICT infrastructure, the university establishes contracts with other companies (e.g. ISPs) to install, maintain, configure and upgrade infrastructure. Contracts are made on an as needed basis, but in no case are made via outsourcing approach. As previously

stated, the university has never engaged in outsourcing in any field. This is due to internal policy regarding the reduction of expenses, essential in the running of a public university due to the low budget allocated them by the Colombian government.

The university does well on its small budget, especially considering it is not as large or complex as other higher education institutions. Therefore, there is no reason to establish new contracts based on outsourcing approach as the budget would not allow it, and the university staff are capable of fulfilling their software needs.

1.1.8. Data Ownership

Database protection is regulated in various parts of the planet, as data contained therein is mismanaged could have serious implications on people's lives. To cite one example of database regulation, the European community, through its parliament and its council, has proposed a directive that establishes the legal parameters for databases protection. This directive clarifies Europe's relations with international policies on data protection and privacy rights stipulated in the World Declaration of Human Rights and other entities (The European Parliament and the Council of the European Union, 1996).

Article 15 of the directive states that it is only possible to establish the rights to the contents of a database if its authorship is known. In this sense, there is a direct relationship between the origin of the data and ownership. The Directive further states that the holder of a database should prohibit unauthorized extraction of data, by regulating demonstration of ownership (Nettleton, Obhi, 2004: 268).

This would lead one to believe that the owners of the databases (in some way or another) are responsible for the data, even if the data does not strictly belong to them. New databases can be created after extracting data, using modern mash-up services and systems integration. This is where establishing data ownership can be more complicated, but nevertheless essential to avoid violating owners' rights (Nettleton andObhi, 2004: 200).

The data ***per se*** should be owned by the individual who generated it. For instance, in the case of personal data such as names, addresses, phone numbers, etc., each person has the right to his/her data. That is, everyone has the right to access, modify / update, delete their personal information, so as long as it does not interfere with the policies regulated by the organizations that maintain the information in their databases. Therefore, in reality people and organizations have shared ownership of data. In addition, organizations that manage databases must establish clear policies regarding their use of people's data.

The above refers to personal data; however, financial data and sensitive data are another matter. In these cases, organizations should responsibility for data. These organizations must be regulated by the government under local and international data protection guidelines and privacy policies. In these cases, organizations own the data. Finally, other types of data, such as information resulting from the work of academic and scientific research, experiments, art production, and commercial production, among others, must be maintained and protected directly by the authors, organizations, councils, memberships, groups, etc. that produced them. This is the case of academic-scientific communities, journals, digital libraries, companies of the production sector, individual authors, and so forth. In these cases, those entities are free to decide whether the data will be published or not, whether to pay for them or not, whether data will expire or not, etc.

In conclusion, it is very important to distinguish the nature of different databases, the use for the data, and other considerations when looking at a universal protection policy (Bagby, 2003). Unfortunately, this is not an easy task due to the various and contrasting points of view around the world. Arriving at a minimal agreement concerning data ownership seems to be a utopian task.

1.1.9. Privacy and Data Protection in Colombia

As in other countries, in Colombia there are specific laws concerning privacy and data protection; in fact, the law has its origins in the Colombian Constitution, which gives all citizens the right to access, modify and delete personal data. In the Colombian Constitution of 1991 (*Carta Magna* in Spanish), Article 15 states:

> All persons have right to personal and family privacy and their good name, and the State must respect and support them. Similarly, they have the right to know, update, and rectify information gathered about them in the databases and files of public and private entities. Freedom and the other guarantees stated in the Constitution will be respected in the collection, processing, and circulation of data (AsambleaNacionalConstituyente, 1991: 3).

Although the definitions, scope and limitations of that article are rather broad, fortunately, a law, popularly known as "Habeas Data", explains relevant aspects of Privacy and Data Protection in Colombia. In general terms, this law regulates the handling of information in personal databases, especially financial, credit, trade and services related data, and information originating from third party countries (Congreso de la República, 2008: 1). Some key facets of this law are described below:

- **Definitions**: In this section, concepts such as Information Owner, Information Source, Information Operator, User, Data Types (Personal, Public, Half-Private, and Private), and Trade Information Agency are defined.
- **Principles**: This outlines the principals involved, such as integrity and commitment to quality.
- **Duties**: The roles of responsibilities of the individual.
- **Requirements**: The schema which defines functional specifications.
- **Sanctions**: Sets forth the punishment which offenses against the law should incur.

Despite the main intention behind this legislation being to regulate privacy and data protection, "Habeas Data" very much revolves around the management of information in the financial sector. There is in fact a whole entity called "Datacrédito", which is responsible for managing virtually all of Colombia's financial information.

In comparing the European model of privacy and data protection with the Colombian one, they generally display the same features. However, the list below shows the criteria that are poorly defined in Colombia. This information comes from doctoral research on the protection of personal data carried out in the Law Department at the Pontificia University Javeriana in Colombia (Remolina-Angarita, 2010: 519).

Criterion	Colombian Law	European Law
Sensitive Data	Not regulated	Regulated
Right of Opposition	Not regulated	Regulated
International transference of data	Vague definition	Partly defined

Table 1.1 Summary of Colombian versus European Law.

In conclusion, Colombia lacks the legal framework which would allow it to be considered a country with an adequate level of data protection according to European standards.

1.1.10. Professional Responsibility

Throughout history, different disciplines have developed their own codes of ethics, which act as guides, defining the responsibilities, rights and duties of professionals. For instance, there is the Hippocratic Oath in medicine, and legal science has certain ethical norms which resolve themselves into a code of conduct. And computing is no exception.

ACM (Association for Computing Machinery), an educational and scientific computing society, has developed its own code of ethics and professional conduct. The society covers several aspects, including morality, leadership, and professionalism, in its recommendations on good practice in computing. The following concepts will be looked at in more detail below: contribute to society and human well-being, avoid harm to others, and honor contracts, agreements, and assigned responsibilities (ACM, 1992).

Professionals in computing have ethical responsibilities to the wider community in their work just as doctors and lawyers do. Humans are social beings; community life is part of human nature. For Karl Marx, man is to the whole of their social relations (1845: 6th thesis). Professionals in different disciplines play meaningful roles in their society, where ancient values demand that we seek the common good for the development of human potential but, unfortunately, some individuals do not think this way. Both statements "contribute to society and human well-being" and "avoid harm to others" outlined by the ACM fall under the basic instinct that individuals should have to better the life of the community (global or otherwise), and clearly entail more specific responsibilities.

Due to rapid technological growth in computing, more and more applications are being incorporated into our routines, and computing solutions now pervade almost every sphere of daily life. Therefore, the social implications and responsibilities of creating computing technology have also increased, making an explicit code of ethics for computing professionals essential. Estell and Christensen put forward an interesting proposal on how to introduce new professionals to the ACM Code of Ethics (2011). However, the proposal could be said to be somewhat superficial, because a mere graduation rite could be taken without the individual really feeling or understanding it. Ethics should not only be a part of the syllabus, but should be a transversal formation strategy across syllabi, to reinforce the real meaning of ethics through formal education.

Finally, ACM stated that professionals should "honor contracts, agreements, and assigned responsibilities". This aims to encourage professional behavior and increase productivity in the computing community. The ACM also states however, that this obligation to fulfill contracts only applies if the contract does not contradict the other regulations set forth in the code of ethics, as this third obligation comes under the cadre of business ethics.

In conclusion, every development in computing impacts society for better or worse. The real question is, what is the world of academia doing to create professionals with an awareness of the impact their work has, and an ethical code?

1.1.11. Future Computing Professions

Computing itself is a rather broad concept, so it stands to reason that there is no one single type of computing professional. According to the ACM Joint Task Force on Computing Curricula, five disciplines can be identified, through their distinct (but integrated) goals. These are Computer Science (CS), Software Engineering (SE), Information Systems (IS), Information Technology (IT), and Computer Engineering (CE) (ACM, IEEE-CS & AIS, 2005: 12).

Several professions related to computing have been created in order to solve specific problems in these specialized fields of knowledge, although some professionals have similar duties and responsibilities even though they have different job titles. For instance, in Colombia, systems engineers are equivalent to computer analysts, or even computer programmers; computing professions in Colombia lack well-defined boundaries. Although computing is huge field, computer programming is the focus of this chapter, because it is where the fastest growth has been most commonly seen in recent years, and computer programming is the common denominator of CS, SE, IS, IT and CE.

The common feature of the above disciplines is programming. Programming is implicit in every single aspect of computing; not everybody is able to create computer programs, but in one way or another everyone interacts with them. This was recognized by the ACM Joint Task Force on Computing Curricula when they stated that programming has one of the highest relevance

scores when comparing the weight of computing topics across the five kinds of degree programs (2005: 24). Many current professionals have born witness to the rapid progress of computer programming, from punched cards to visual development and mash-up techniques. Now, some organizations are thinking about removing their IT departments (IT Now, 2010). This could be caused by the arrival of new computing solutions that work via outsourcing and new approaches to SaaS (Software as a Service) via the Cloud (White, 2011: 11). Clearly, the future for programmers looks worrying; computer programming is becoming a trivial task. Nowadays, it is the skills required to design systems and software architectures that are highly valued, and these skills do not necessarily imply programming.

In conclusion, computing will continue to multiply in the future to face the new challenges, features, requirements and technology-based support. Nonetheless, the specific field of Computer Programming (if that name continues to be used) will suffer drastic changes in terms of concept foundations and techniques. The next generation of programmers in 25 years will think that using Microsoft .NET Framework, as we do now, seems prehistoric.

1.1.12. Pros and Cons of New Ways of Communication

Communications have evolved rapidly in recent decades which has impacted several aspects of society, none more so than computing. Following the advent of the telephone in the late 19th century right up to the invention of modern IM and SMS technology, the way human beings communicate has changed radically as technology has evolved (Adams, McCrindle, 2008: 566).

One of the main concerns about these changes relates to the transformation of language itself. For some researchers, such as David Crystal, the impact of technology on language is quite

a complex issue. Studies on the influence of texting on language show both pros and cons, as shown in the following table: (Crystal, 2008).

PROS OF TEXTING	CONS OF TEXTING
Texting allows us to send messages faster	Texting uses new and nonstandard syntax.
Texting helps shy teenagers to communicate	A new generation of adults will inevitably grow up unable to write properly.
Texting is cheaper than other forms of communication	Texting ruins children's ability to spell, punctuate and capitalize correctly.

Table 1.2 Pros and Cons of Texting

The birth of a new communication style and the death of written language seem to be happening simultaneously. It could simply be the natural evolution of language; nonetheless, every one of us, and perhaps academics in particular, should pay special attention to this phenomenon to avoid the loss of our linguistic heritage.

On the other hand, the younger generation is fluent in new communication technologies. However, IM and texting do imply some (non-language related) risks. For instance, there are infinite reports of car accidents due to people (teenagers especially) texting while driving. Texting could be considering risky because of its addictive nature.

In Colombia, a Spanish-speaking country, the reality of the texting and IM phenomenon is clear. For instance, these are some samples of commonly used contractions in texting:

Spanish (Correct Form)	Spanish (Texting)	Translation
¿Hola cómo estás tu?	olacmotas?	Hi, how are you doing?
Yo te quiero mucho	Tqm	I love you so much
¿Qué te parece?	q tpareC?	What do you think?
¿Para qué?	pa q?	What for?
¿Dónde estás tu?	ontas?	Where are you?

Table 1.3 Sample of texting in Spanish

Young people perhaps do not like to waste their time when writing messages. However, this leads to orthographic errors according to the Real Academia Española (the official institution that regulates the correct usage of Spanish). Some argue that contractions are vulgar, and make messages more difficult to understand, sometimes messages seem even cryptic. However, there is no way of policing grammar; texting belongs to the younger generation, and they will adapt it as theirs.

1.1.13. ROI in the workplace

The University of Nariño is a small public university, and a non-profit organization. The authors are currently employed as professors of computer science, working for the systems department. The professors in the systems department design software, program computers, and more recently participated in the construction of technology solutions for the university's information systems.

The university has a very clear policy with respect to operating costs. In short, the general budget of the university is very limited; ergo, when implementing new IT, the university must reduce keep staffing and technology investment costs low. That is why it usually taps into the labor potential of its own students, involving them in internships.

As a non-profit organization, the University of Nariño seeks to contribute to peace, coexistence, social justice, and academic education and research, as well as being committed to regional and intercultural development in the community (Unidad de Informática y Telecomunicaciones, n.d.). As such, the return of investment on IT projects is measured through indicators of quality of service (education) and academic goals achieved.

The planning and development office in the University of Nariño stated that the management of information and technology comes under "Support Processes" (2010). These kinds of processes are the main processes at the university, in other words the mission processes, which are academic formation, research, and social projection.

Therefore, everything related to the field of ICT must be justifiable in accordance with the university's mission processes. This is the current way in which the university measures the relevance and efficiency of their technology investments. In other words, the university is only interested in goals which contribute to academic formation, research and social projection. There are no aims which justify economic profit; therefore, ROI must be measured in terms of the final product, and how it betters education and social responsibility.

1.1.14. Software Risks in the Workplace

The authors, as previously mentioned, work for the University of Nariño in southern Colombia, and had the opportunity to construct software for the institution alongside colleagues and students. The software required to manage academic information and administrative processes in a public university is quite complex. Several modules are integrated into the complete information system, including: Academic, Finance, Human Resources, Physical Resources, and Document Management (Insuasti, 2009: 14).

The university has its own computing center which is in charge of the design and development of computer-based solutions within the institution. As the staff in the computing center is sometimes not able to meet all the university's information systems requirements, some computer programs have been made fire-fighting style. This means that few software engineering criteria were considered when building some of the modules. That is why it is fairly difficult to integrate all the modules, which puts the scalability of the whole system at risk.

With this in mind, the software development processes are obviously very complicated at the university. However, there is one problem that is significantly worse than the others, that is the lack of good documentation practice concerning software construction. This is not an uncommon problem; in fact, in many places employers do not demand thorough documentation, which leads to several problems further down the line (Parnas, 2010: 27). According to the authors' experience, it is a problem that is consistently overlooked, and not well managed within the organization. It is one of main causes generating huge issues in terms of maintaining software support during its lifecycle.

Unfortunately, the staff working at the computing center is paid a low wage. This means the staff turnover is high, as they leave and find better opportunities elsewhere. The high turnover could be countered using a policy of economic incentives to ensure staff retention. As a strategy to mitigate the severity of this problem, a minimum standard for documentation must be enforced; in this way, those who work in the center could follow these regulations to create well-documented software material.

Finally, an improvement in the contracts of the computing center staff, in terms of minimum duration, wages and economic incentives is essential. It might sound straightforward, but a big problem in all public universities is the small budget allocated them by the government. In conclusion, an alternative solution could be to outsource an external company which provides the development and maintenance of information systems.

1.1.15. Intellectual Property

In terms of hardware, the production of computers (which are tangible products) involves material costs which influence the value of the product. Unlike hardware, the value of software (an intangible product) can only be estimated based on the intellectual work required to make it (Wiederhold, 2011: 68).

Each software product must have an economic value and there is one figure which represents its value (be it millions, thousands, or zero). As a product, software is the result of the efforts of one or several people whose job represents time and intellectual production. In general terms, intellectual property is defined by a collection of regulations such as patents, trade secrets,

trademarks, and copyrights, among others, which grant the ownership of a specific product to someone (Adams &McCrindle, 2008: 402). These regulations seek to vindicate the work of the author, and contribute to the protection of the use and reproduction of such property by granting rights to the owner, and restricting the rights of others.

The word "restriction" is sometimes demonized, and some people argue that while "restriction" exists, freedom cannot be achieved. However, the word "freedom" is a complex one, as people have always lived in a society with restrictions in terms of ethics, morality, civil and criminal law, economics, trade, and so on. Restrictions do have compensations, such as levies and value-added taxes which generate profit and such mechanisms tend to be granted by local governing bodies, meaning that the extra income is not handled by middlemen, but by governments. From this perspective, it would be beneficial to the majority to impose taxes on software products. However, it is also important to ensure the creators share in the profit made from their product; it is a just recognition of their work and investment.

In conclusion, strategies such as levies and value-added taxes on products are good for common welfare. However, this alone is not enough as the authors deserve to profit from their software products. It is a job like any other which involves sacrifice, time, dedication, effort and emotion. Although software is arguably many things (progress, support, tool, etc.) it is essentially a business. That is why the software industry exists; software construction has particular features such as malleability, negligible duplication costs, and low distribution costs, but in the end, software construction is a business environment like any other (Favaro&Pfleeger, 2011: 25).

1.1.16. Why does Risk Management fail?

Risk management can be traditionally carried out in four ways (Gómez, R. et al, 2010: 110):

1. Generate controls to mitigate risks.
2. Accept risks as they are, because it is impossible to control them.
3. Eliminate risks by removing the business processes that generate them.
4. Transfer risks to another party, for example, insurance companies.

According to above, the ways in which risks are managed within organizations all require inner policies to be established to face whatever eventuality may arise. This in itself is one of the main causes of risk management failure. Unclear risk management policies are considered a threat to the working of an organization, and can cause large financial, marketing, staffing and management losses, among others.

Furthermore, good risk management policies alone are not enough. According to Landier, Sraer and Thesmar, it is important to establish a corporate strategy that allows these risk management policies to be aligned with concrete actions, specific workplaces and jobs; this corporate strategy should follow the guidelines of widely-tested models in real market environments (2009: 454).

In the University of Nariño, the lack of clarity as regards risk management policies is striking. Unfortunately, there is lack of leadership from the upper levels of the university concerning strategy coordination. Currently, there are serious problems when it comes to the

implementation of new information systems, due to risk management failures. One of the main reasons that causes risk management to fail at the university is the lack of strategic planning in terms of risk estimation. Unfortunately, in some cases, risks arise which were not previously considered or identified in the initial planning of projects. The worst instance of this occurred when some risks were obvious to the professionals working in the computing center, and yet they did not take into account, when planning some scalable software projects, issues related to data storage overflow.

Finally, Lam (2006: 6) stated that some independent risks (risks related to corporate dependencies) could meaningfully affect the general risk profile of a whole organization. In the case of the university, the planning office is at present working hard to identify independent risks for each dependency (unit or department) which could affect the whole picture. Some critical weak points are: the computing center and the vice presidency of administrative and financial matters.

1.2. WORKSHOPS

1.2.1. Hand-In Assignment – Seminar 1

1. Computing Functions Which Improve Technology

Computing impacts practically every aspect of human life, people interact with computer-based devices, instruments or machines every day and in every sphere; entertainment, health care, business, transportation, and education, among others, are supported by computing innovations and adaptations around the world. In one way or another, computing is breaking the boundaries between disciplines while enabling their growth. Some computing functions have a direct relationship with the transformation of the world today, of which 5 will be examined below: miniaturization, storage, intelligent behavior, ubiquity, and handling. These concepts can be defined by their relevance in real daily contexts:

Miniaturization: People are demanding demand smaller and even invisible devices (Mills, 2011: 24). This is best achieved through miniaturization, which is the technological capability to decrease hardware size in order to make it more portable, flexible and easy to use.

Storage: A previous prediction about the amount of digital information available by 2011 was 1.8 zettabytes, that is to say 1.8×10^{21} bytes (Berman, 2008: 52). Storage is a technology feature that allows enormous quantities of information to be managed efficiently. Nowadays, it is the basis of practically all information storage, via database systems.

Artificial Intelligence: Computers are very useful machines; however, computers can normally solve problems using algorithms. Artificial intelligence in computing solutions however,

goes further than that. The concept revolves around "smart" agents, defined as independent entities which carry out autonomous actions to achieve their goals (Wooldridge, 2000: 1).

Ubiquity: The omnipresence of computing is also called Ubiquitous computing. This is the concept of constant manipulation of possible actions and space within a complex system (Waller, Johnston, 2009: 130). Ubiquity is the feature that is moving us from the distributed system approach, towards a world in which computing is as close as possible to inseparable from human life.

Handling: Over time, we have been looking to create computing solutions which solve problems more efficiently. The management of large amounts of data, and the complexity of information systems, have allowed the development of new approaches which facilitate people's jobs. Handling deals with human-computer interactions (HCI) and it is a feature which is constantly growing. HCI works with several principles, such as responsiveness, user adaptability and feedback, low mental load, intuitiveness, and comfort, among others (Wachs et al, 2011: 62, 63).

2. Issues with implementing Artificial Intelligence

Privacy:

Privacy goes hand in hand with control and safety. Privacy can be compromised if there are flaws in security or control of autonomous systems is lost. Therefore, the security issues in our current systems would be the same in artificially intelligent systems.

Ownership:

Complex systems powered by intelligent technology would have the same implications as those of current intellectual property. Currently, there are regulations that protect the ownership of technological products, such as patents and copyrights. People can expect to face similar legal issues with artificially intelligent technology as they do now.

Control:

In science fiction there are many examples of apocalyptic futures where artificial intelligence takes control of all aspects of life. In a sense, it is true that machines equipped with intelligence might consider the human beings a threat or vice versa. The film industry has recreated this idea time and again with movies such as Terminator, Matrix, and I Robot, among others. In real life however, people have total control over AI implementation so far, which leads to the question, is AI entirely autonomous? If autonomy is the ability to self-govern, then, is the work of the academic and scientific community on AI thus far valid?

Accuracy:

Artificial intelligence works using purely one of the multiple types of intelligence: logical. Humans however, go beyond numeric reasoning alone; feelings, emotions, attitudes, and so forth, are considered complements to logical intelligence processes according to Gardner (2006: Preface). In conclusion, in terms of objective precision, AI works fine, however that precision could have some impacts in human life. In a hypothetical situation: a machine could decide to terminate a patient's life due to numeric survival chances versus economic spending; however, it would have no concept of whether it would be ethically or morally acceptable.

Security:

According to the tragic predictions of science fiction, our security could be affected by the implementation of intelligent behavior if key systems are out of human control. In the case of robotics, a code of responsibility for machines could be established based on Isaac Asimov's famous three laws. In the same way, intelligent weapons could be a threat to peace in the case of a technical error; the same could happen with intelligent solutions, which support life in health care and patient attention: in these contexts, an error can have catastrophic consequences.

3. Possible IT policies derived from the implementation of Artificial Intelligence.

In accordance with the above, implementing AI based technology would incur additional responsibilities. In general terms, research into AI in computing looks to develop responses to problems with human-like solutions. With this in mind, academic and scientific communities should direct their efforts toward defining an intelligent system's limits of autonomy; that is, to what extent intelligent systems should be able to govern themselves.

Professional responsibility goes beyond complying with certain labor standards or duties in business and contracts; it transcends the workplace and has a direct impact on society. In computing, in order to maintain values and professional responsibility, education is essential in raising awareness. Obviously, education takes time, and the results cannot be seen immediately; only time will tell how responsible today's computing students are as professionals.

1.2.2. Hand-In Assignment – Seminar 2

Electronic Monitoring in the workplace.

IT Consultant Jesus Insuasti, System Engineer. MsEd.

Professor of the Systems Department Galeras.NET Academic Cell

Mentor - Microsoft Registered Partner

Universidad de Nariño - Campus Torobajo - Block 6 - Office 109

Website: insuasty.pbworks.com

E-mail: insuasty@udenar.edu.co

Phone: +57 (2) 737-0765 Pasto – Colombia

SUMMARY: This report encapsulates the different perspectives on electronic monitoring in the workplace. It presents theoretical approaches to organizational settings, and the ethical implications of the use of electronic monitoring and a professional recommendation.

Theoretical Approaches in organizational environments

When Frederick Taylor quoted President Roosevelt on, "the larger question of increasing our national efficiency" (Taylor, 1911: 5), he laid the foundations for his own theory on scientific management which came from his vision of prosperity. In order to maximize natural resources, Taylor considered that human effort should not be wasted under any circumstances. In his quest, Taylor was looking for the most competent men to join a task force in industry, in order to reach the highest possible levels of production.

According to Taylor, the U.S. was suffering due to the inefficiency of its people; he therefore proposed a scientific approach to management, to increase efficiency through clear rules, laws, and quantitative principles. His famous phrase: "that maximum prosperity can exist only as the result of maximum productivity" (Taylor, 1911: 12) encapsulates his vision of high performance by minimizing waste (time and energy).

A controversial aspect of Taylor's theory is related to the direct relationship between wages and dedication. In general terms, the theory leaves open the possibility of using any mechanism possible to minimize the waste of labor and time. Thus, one possible mechanism available is the monitoring of employees, which would be a useful tool to verify that their workforce and their time are not wasted.

On the other hand, two theories of productivity have been put forward with a focus on motivation. Theory X and Theory Y are two mutually exclusive ways of perceiving human behavior, taken by managers to motivate employees and high productivity. Theory X focuses in hard work and Theory Y focuses in pleasant work. McGregor stated that Theory X is a mere traditional view of direction and control, and Theory Y responds toward an integration of individual and organizational goals (McGregor, 1960: 43). Two possible scenarios arise from these disparate visions: the first one in which people are a cost that must be monitored and controlled, and the second one in which people are an asset that should be valued and developed (McGregor, 1960: XLVI).

In spite of the differences between the two McGregor theories, both focus on the stimulus-response modality. Both Theory X and Theory Y consider the attitude of the worker. In theory X they are considered lazy, with poor attitudes and enterprises should threaten them with punishments to increase their productivity. In this scenario, pressure is the stimulus which causes workers to lose their "natural" behavior, which is avoiding duties and responsibilities. Theory X therefore, justifies the use of monitoring techniques as a control measure to increase productivity.

In contrast, Theory Y promotes a collaborative environment based on the enhancement of workers' abilities. In order to achieve this goal, Theory Y uses motivational strategies based on the idea that workers have good intentions and positive attitudes towards their jobs. As a consequence, motivation and opportunities to grow are the stimuli which cause workers to exploit their potential, which takes place in a suitable and comfortable labor environment, increasing productivity

.

Ethical Implications of Electronic Monitoring

Electronic monitoring is a reliable, technology-based solution for employers who want to ensure that workers are using their time efficiently. However, it can invade people's privacy (Nord, McCubbins, Nord, 2006: 74). The report outlines some of the ethical implications of using Electronic Monitoring in the workplace, in this case, in software production.

One of the problems that arises with monitoring is the concern that employees are being monitored against their will (ACM, 2006: 10). This can lead to a negative atmosphere in the workplace. On the other hand, employers might find that monitoring leads to a more positive attitude, as interruptions and distractions, such as non-work related software (chat and messaging) are removed, which invigorates employees and helps them focus; when blocking is exercised in moderation of course (Bock, Ling Ho, 2009: 128).

The question that remains unanswered however is, how ethical is it electronically monitor employees? An interesting study about the ethical implications of electronically monitoring employees showed that employees have no problems with electronic monitoring if they are forewarned by their employer. They only consider it unethical when it is carried out secretly or without warning (Taylor, 2007: 144).

Monitoring in the workplace is not a new trend; in one form or another there have always been restrictions and ways of observing employees, for instance: drug testing, psychological testing, searches etc. Nowadays, monitoring with technology can be done by checking someone's e-mail, voice mail, messaging, phone calls, chats, internet history, and so forth. The ethical issues are the same surrounding how far an employer has the right to know about their employees' private lives. Around the world however, there are laws and regulations which preserve privacy, but in some cases, they clash with company policies.

Professional Recommendations

Taking into account the above pros and cons, it can be concluded that electronic monitoring of the employees' activities within the workplace is a good policy, so long as there are clearly defined rules relating to its use. For example, the following variables should be considered:

- International agreements on privacy: for instance, the World Declaration on Human Rights.
- Global trade organizations and work organizations: for instance, International Labor Organization, World Trade Organizations.
- Local legislation: country's constitution or chart, state or province legislations
- Enterprise policy: the legal policy in the workplace

An important point that must be emphasized is that electronic monitoring should only be established when employees have been forewarned. Furthermore, it would be advisable to establish a

clear system of economic incentives for productivity to motivate employees. Of course, the monitoring policy and the incentive policy must be laid out in the manual of procedures in the workplace.

With human resources, enterprises should take the positive points of the above theories and adopt them without going to extremes. Control is good to a certain extent, but in excess feels oppressive, and equally freedom is positive to a certain extent, but in excess might cause chaos. It is a question of finding the right balance.

1.2.3. Professional Ethics: Legal Transgression on the part of the Customer

In their book, Adams and McCrindle pose an interesting ethical dilemma, in which any software programmer could be caught and their ethics tested. If a professional suspects that their customer is carrying out illegal activities, related to false accounting for example, and the customer asks for technical support to avoid the software constraints created by the professional, what course of action should they take? (Adams, McCrindle, 2008: 326). In order to examine the situation more closely first the following questions must be answered.

How far should you be willing to help your customer, aware that you may risk aiding and abetting tax fraud?

In this case, the customer is committing fraud, and not through ignorance but consciously. In the oath taken by professionals at the graduation ceremony, there is a clear statement about the social responsibility that they have not to create software that could be used for illegal purposes. As such, it is impossible to fix software so that it can be used fraudulently for two main reasons:

Firstly, it would go against personal and professional values. In accordance with the Alma Mater oath, as a software writer you can be accused of unethical behavior and charged by a government authority. For the authors, in Colombia, the principles of the public universities are

widely known, and it is public knowledge that anyone can report the illegal practices of public university professionals to the state. It is also against the principles of public teachers as it represents a total negation of their aim to better the community.

Secondly, the legal sanctions against fraud in Colombia are extremely harsh, which will be looked at in more detail later. This kind of fraud falls into the cadre of falsification of financial evidence, according to Bernate (2007: 52), and the manipulation of information in accounting is considered a crime, and Article 5, Law 43, 1990 states that the Colombian government exercises vigilance in accounting and financial matters (Congreso de la República de Colombia, 1990: 3).

Taking all this into account, computing professionals should evaluate each situation separately, so as to avoid contradicting their ethics, values and principles.

Would the software programmer/company be personally and/or criminally liable if they fulfilled the customer's request?

Of course, as mentioned above, the crime is first degree falsification of financial statements. Unfortunately, Colombia has a history of fraud in drug trafficking; thus, the first step legally is to carefully analyze the fakes order to determine why financial information has been falsified. If the reason was tax evasion, it may be less serious, if the main reason was money laundering however, or linked to drug trafficking, there is no chance of bail and a jail sentence is certain. That is why in Colombia, as in many countries, accounting fraud is a very serious issue.

This is the legal situation that a company involved might face. The owner of the company however, also shares in the responsibility for the crime. In other words, the company and the individuals involved will face complicity charges. The charges against employees would depend

on the degree of responsibility of all those involved; generally speaking, in Colombia, all employees who had prior knowledge and participated consciously could go to jail.

What does due diligence require in this case?

Firstly, the computing professional is responsible for warning customers that the actions they are performing with their software are unlawful. Furthermore, they must firmly refuse to alter their software. Finally, it would best to talk a superior about the case, explaining the whole situation in detail.

The problem does not end there however. If one or more of the customers in question are professional graduates from a public university, they should be reported to a government authority. That is the public duty of any individual, but there could be personal reprisals. Unfortunately, in Colombia, whistle blowing could lead to your life and those of your family being in danger. Due diligence is therefore not easy to define, as some of the personal consequences can be risky.

In general, what responsibilities do computing professionals have in situations like this?

In essence, all types of professional production in computing must comply with local, national and international regulations. Computing professionals have a social responsibility to work, and ensure their employees work, within the law. In this particular case, a computing professional should warn people if they are engaging in illegal activities.

Is it not justifiable to simply produce (as employees and as companies) what the client requests, if the alteration itself is legal?

No, this case is a counter-example of the classic motto: “The Customer is Always Right”.

1.2.4. Protecting Intellectual Property: Software

As university teachers in Latin America, the authors have witnessed a growing interest in the creation of knowledge transfer offices and technology, and the management of intellectual property. In the case of universities, colleagues, particularly in universities in the U.S. and Europe, have made great efforts to generate income through research.

The successful cases of those universities which carry out research that has a significant impact on society all have one thing in common: effective management of intellectual property. According to Manderieux, universities (public ones or private ones) with the ability to innovate, create and invent should establish their own clear policies around intellectual property (2011: 3).

Intellectual property policy is the branch of law that protects the ownership of ideas. Intellectual property covers creations, physical or otherwise: inventions, ideas, literary and artistic works, symbols, names, images and drawings.

Intellectual property is divided into two categories: industrial property, including inventions, patents, trademarks, industrial designs and geographical indications of origin; and copyright, which include literary and artistic works such as novels, poems and plays, films, musical works, works of art such as drawings, paintings, photographs and sculptures, and architectural designs.

There are important processes which protect intellectual production. Those mechanisms are: patents, copyrights and trademarks among others; some definitions are provided below.

Patents:

Patents are official documents that give an individual recognition of ownership of an invention and the rights this entails. Samuelson stated that this kind of protection mechanism prevents competitors from copying, using or adapting the invention (2010). A patent could protect a software product and grants practically all the rights over it to the owner. One of the most important features of patents is the right to temporarily prevent others from manufacturing, selling and/or commercially using the patented invention.

The following features are used within the patent arena:

Invention: Creating something new to solve existing technical problems.

The patent holder can:

- Decide who may or may not use the patented invention during the period in which it is protected.
- Grant permission to third parties to use the invention, subject to conditions set by mutual agreement.
- Sublicense, which is a legal agreement between two parties in which the owner of a patent grants an operating permit to another person or company for a period of time.
- Sell the rights of the invention to a third party, who then becomes the new owner of the patent.

Finally, when a patent expires, the protection also expires and the invention enters the public domain, that is, the owner no longer has exclusive rights over the invention, and it becomes available for commercial exploitation by third parties.

Trademarks:

A trademark is a sign that is used to identify companies and distinguish goods or services on the market. A trademark, that serves to distinguish goods or services as belonging to a company, may be one of its most important assets. It impacts the consumer's choices, based on their knowledge of the quality of a company they may or may not choose a product or service, giving successful brand names an edge on the competition when introducing or endorsing new products.

Trademark protection only covers very small pieces of "information", such as a word, a motto, or an icon (Adams &McCrindle, 2008: 403). Advertising slogans are words, groups of words or phrases that contribute to strengthening commercial products and services and can also be identified by a trademark.

Copyrights:

Copyright can be defined as a set of privileges that the legal system recognizes and give the creators of original intellectual works in literary and artistic fields, in its various manifestations, such as writing, music, drawing, painting, sculpture, and printing, among others.

Computer programs or software, as well as other newer creations not originally covered by copyrights and related rights, are facing serious problems in terms of distribution and marketing, within and outside of cyberspace. Software marketing, using traditional and current digital media, must follow the licensed mode of use, this type of marketing has evolved with technology and current market needs.

1.2.5. Hand-in Assignment – Seminar 5

1. Explain the meaning of the terms "fixed assets" and "current assets", illustrating your explanation with suitable examples.

Current assets are assets which are likely to become cash in a short period of time, roughly less than a year (Goldberg, 1996: 31). Examples of these kinds of assets, besides cash and money in the bank, are short-term investments, portfolios and inventories.

At the university, the authors are responsible for computing equipment, such as computers, PDAs, and GPSs, all useful for research purposes. These items all come under the category of current assets. The university usually sells items in its inventory which are not immediately academically useful; as such the aforementioned items could be transformed into cash at any time.

Fixed assets mean those assets that are not to be turned into cash in the immediate future, and include physical infrastructure, real estate, large equipment, and furniture, among others (Mykolaitiene et al, 2010: 143).

In the university, for example, the office and its furniture, which belong to the faculty building, is part of the university campus; ergo, the office is one of the university's fixed assets.

2. Describe how the two types of assets are valued for balance sheet purposes, using the following assets owned by a company that writes and sells software packages:

a) The company has a stock of 500 user manuals for version 1 of a package—version 2 of which is to appear shortly. The company paid $5,000 to have 1,000 manuals printed and has been selling them at $25 per copy.

b) The company has a file server costing $15,000 that is used by the software development teams.

In the previous example, the company clearly has an inventory. This inventory is part of the current assets of the company; in this case, the user manuals which are items which will be sold shortly. However, if the main objective of the company is to write and sell software, user manuals cannot be considered current assets.

The inventory system of the company probably has more items, such as office equipment (although it is not entirely known whether this company has offices). Those items could be considered the current assets of the company. However, the user manuals are part of the final product, software, even if those manuals can be sold independently. Therefore, user manuals cannot be considered current assets.

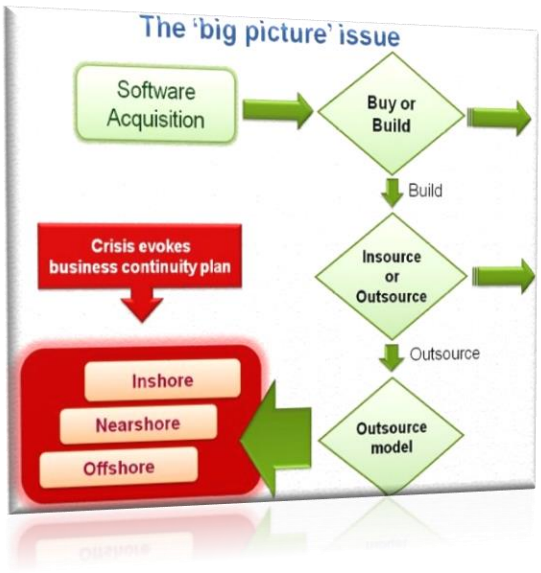
On the other hand, the company has a file server which can be considered a current asset, because it could feasibly become cash if sold. In addition, hardware in IT needs to be updated constantly as it evolves so quickly. Hardware is not durable, a server more than a year old can be considered obsolete. This is an argument for renewing the hardware infrastructure.

As the description of the company is incomplete, it is unclear whether the company has a physical infrastructure. This company could be operating without a physical office, and therefore it is unclear whether the company has fixed assets or not. Some people would put the file server in the category of manufacturing equipment (and it would therefore be a fixed asset); however, it is more likely to be a current asset for the reasons stated above.

1.2.6. Hand-In Assignment – Seminar 6

Sometimes organizations outsource specialized services, in particular, where software development is required. However, there can be grave consequences when outsourcing software development fails.

The "big picture" of IT outsourcing is defined by Dr. O'Connor. In general terms, Professor O'Connor established a basic algorithm to explain how IT outsourcing is carried out, see Figure 1.2.



In short, everything starts with the need to acquire software. First off, the organization takes the most important decision, whether to buy or to develop. Following this decision, the second choice is to establish who will build it: existing staff or external contracts. When the organization decides on an outsource model, the approach must be defined: inshore, nearshore and offshore.

Finally, Dr. O'Conner stated that a crisis in outsourcing leads to planning to protect business continuity.

Figure 1.2. The "Big Picture" Issue with IT outsourcing (2011)

Taking all this into account, Professor O'Conner's proposal is based on the cost/benefit relationship. The pros of IT outsourcing include lower cost, increased flexibility, faster development speed, coverage of taxation, and accounting tasks. However, the cons of IT outsourcing include hidden costs, legal issues, plus the list of risks below (among others):

- Unexpected transition and management costs
- Lack of experience and expertise of the client with the activity
- Lack of experience with outsourcing
- Costly contractual amendments
- Technological discontinuity
- Contractual disputes and litigation
- Measurement problems
- Lack of experience and expertise of the client and/or of the supplier
- Interdependence of inner activities
- Performance ambiguity
- Lack of experience and expertise of the supplier with the activity
- Supplier financial stability
- Lack of experience and expertise of the client with contract management
- Loss of organizational competencies

According to his research, the above situations are ones that could cause IT outsourcing to fail. It must be noted that the most common risks relate to human error. In fact, lack of experience in the field is one of the most common precursors to failure.

It could be argued that human error is mostly responsible for failure (and equally for success) in IT outsourcing, in spite of other factors such as money, time, contractual terms, etc. all influencing the outcome. Selecting the right staff to deal with IT outsourcing within the organization is key; a common mistake relates to making management staff responsible for the control of IT outsourcing processes, rather than the organization's IT staff.

However, Dianne Frank, Editorial Manager for the CIO Executive Council and a contributing editor to CIO magazine and CIO.com, asserts that one of the most common causes of IT outsourcing failure is the lack of inclusion staff other than the IT personnel. In this vein, she states that organizations should not have IT personnel alone in charge of outsourcing. She suggests including people from other parts of the business, as close to the top as possible (2008: 68).

The authors do not have any experience with IT outsourcing; all the information systems within the university were designed and developed by existing staff. Nonetheless, there have been serious problems related to software construction due to lack of previous planning processes. Currently, the university has computing needs in terms of information management, but unfortunately the computing center has a fire-fighting approach to development. There are no scalability criteria, which is why the integration of the existing modules is a significant task.

At the university, it is certain that the staff in the computing center are well-selected based on their skills and proven experience, in this case the problem is not their skills, or their level of expertise. Therefore, there must be other factors affecting the IT development process. The computing center has never outsourced IT contracts; however, the existing staff are perfectly capable of producing their own software or leading an IT outsourcing process. The problem lies in the lack of clear policies surrounding software construction, which would take into account risk

management and focus on well-documented products, and furthermore the loss of personnel due to low wages and lack of economic incentives.

In conclusion, a clearly defined IT policy is paramount when trying to mitigate the previously identified risks. This policy should have regulations about hiring IT personnel, good strategies aimed at staff retention (which could be through improved wages and economic incentives). This policy should also include clear methods for constructing well-documented software, or how to lead IT outsourcing processes. As stated above, the human factor is what matters most.

1.2.7. Hand-In Assignment – Seminar 7

In order to identify the potential risks involved when making modifications to an existing application, the possible variables surrounding a specific case have been identified in the list below:

- 30% probability of delay in receipt of resources - cost $50,000. (R1)
- 20% probability that the resources will be $10,000 cheaper than planned. (R2)
- 25% probability that there will be a problem integrating with existing software, cost to fix $3,500. (R3)
- 30% probability that the development may be simpler than expected, savings $2,500. (R4)
- 5% probability of a design defect causing $5,000 of rework. (R5)

In order to calculate the net expected value for this project's risks, the recommendations stated by the National Institute of Standards and Technology have been followed (NIST, 2002). The first step is classifying the provided risks in categories:

Risk	Probability of Occurrence	Rating	Effect
R1	0.30	Not very likely to occur (Low)	Negative
R2	0.20	Not likely to occur (Low)	Positive
R3	0.25	Not likely to occur (Low)	Negative
R4	0.30	Not very likely to occur (Low)	Positive
R5	0.05	Almost sure not to occur (Low)	Negative

Table 1.4 Risks by category

There is a direct relationship between the likelihood of a risk occurring, and its economic implications, whether positive or negative. The main purpose of this exercise is to create a contingency plan which includes enough in the budget to cover all the potential eventualities related to the risks.

In general terms, the probability of any of these situations occurring is relatively low. With this evaluation done, the next step is to produce a list of possible controls; the following items have been suggested:

- Create the conditions to produce positive effects (Related with R2 and R4)

Risk	Possible Action	Savings
R2	Do research into resource providers to find the best possible prices; these studies should produce strong relationships in trade with the company	$10,000
R4	Promote good software construction practices through extensively tested methodologies; use well-documented software assets under the reusability approach in order to save time and effort.	$ 2,500

Table 1.5 Actions for R2 and R4

In the best of both cases, it would be possible to save $12,500 total.

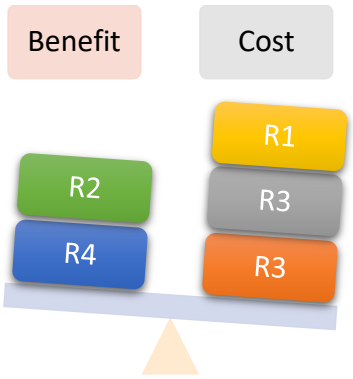
- Create a strategy to minimize the impact of negative effects (Related to R1, R3, and R5)

Risk	Possible Action	Costs
R1	Have a Plan B for the acquisition of resources; it is possible to see if other companies can offer the resources required in order to avoid delays. If the resources are related to money, bank loans could be included in these alternatives.	$50,000
R3	The same as risk 4.	$ 3,500
R5		$ 5,000

Table 1.6 Actions R1, R3, and R5

In the worst of all these cases, it would be possible to lose $58,500 total.

A cost-benefit analysis shows that three risks (R3, R4, and R5) could have the same solution to achieve the best results. Therefore, establishing a clear software engineering process in the company could minimize the risk associated with going over budget, and maximize the opportunities to save money:



	Money (accumulation)	Probability (Average)
Benefit	$12,500	25%
Cost	$58,500	30%

The potential costs of the project outweigh the potential rebates. The risks' budget should have enough to cover this.

Figure 1.3 Cost-Benefit relationship

Finally, some remaining questions relating to this particular case:

How much should you plan for your contingency reserve budget based on the above?

Benefit	**%**	**$**	**Equivalence**
R2	0.20	10,000.00	2,000.00
R4	0.30	2,500.00	750.00
		Total:	**2,750.00**

Cost	**%**	**$**	**Equivalence**
R1	0.30	50,000.00	15,000.00
R3	0.25	3,500.00	875.00
R5	0.05	5,000.00	250.00
		Total:	**16,125.00**

Reserved Budget for Risks:	**18,875.00**

Table 1.7 Cost-Benefit relationship

How much would you allocate for the management reserve? And what are your assumptions about these reserves?

It would be advisable for management to handle 10% of the total risk budget as an AUCC (Administration, Utilities and Contingencies Costs).

2. PROGRAMMING THE INTERNET

2.1. REFLECTIONS

2.1.1. The "Deep Web"

The term "Deep Web" refers to content which is not accessible through a search engine due to storing techniques; that is, content that is not stored in HTML files, but in databases for instance (Computer Desktop Encyclopedia, 2011).

According to the definition above, there is an inaccessible "invisible web" for people who cannot access certain information systems. Authentication processes protect the deep web as the information contained therein is sensitive: person data, financial information, and industry secrets, for example. One of the biggest concerns today is safeguarding information. Unfortunately, access to sensitive information, due to insufficient data privacy protection in certain information systems, has aided acts of terrorism (Stein & Harrell, 2011). People are more aware now of the fact that not all information should be available to everyone, which is the main principle behind privacy policies.

The "Deep Web" as a concept was conceived for both commercial and academic/scientific contexts. All kinds of information in business, monetary transactions for example, needs to be handled with care and individual privacy maintained (for instance, credit cards' transactions, loans, payments records, etc.). Similarly, in academic/scientific scenarios, the management of some personal and private data must be secured as it is strictly private (for instance, grades, assessment material, intellectual property, etc.).

The implications of search engines creating public access to “Deep Web” material are considerable. However, international organizations, such as the entities in charge of data protection and privacy in EU and US, have clear guidelines and regulations in their directives. Hillyard and Gauen stated that attempts to access private information via search engines, such as Google, AOL and Microsoft, have sparked wide criticism, as the complications of giving the general public access to private information would be catastrophic (2007: 121).

According to international, national and local regulations concerning privacy and data protection, search engines should not be able to access “Deep Web” material, as this would infringe data privacy. In terms of exposing private financial information, some potential risks were found in connection with Google and Visa credit cards in terms of exposing private images (Bartholomew, 2009: 56). Unarguably, search engines share in the responsibility of organizations to protect people’s privacy and uphold data protection policies online.

2.1.2. Analysis of Protecting Code in Client-Side Scripting

In programming, there are several approaches to constructing computer-based solutions to face specific problems; one such is scripting. Unfortunately, there is some confusion between the terms “script” and “program”, and although similar there are subtle differences in their meanings.

Firstly, programming is best-known as the act of writing computer programs. Logically, this would lead one to believe that scripting is the act of writing computer scripts. Therefore, what needs to be established is the difference between computer programs and computer scripts. Scripting languages are a kind of programming language with a set of particular features. Since the introduction of the term “script” in early 1970s in UNIX shell environments, scripting has been

considered simply a sequence of commands to reach a goal, which run and are read by a language processor (Barron, 2000: 4).

Nowadays, scripting has three meanings (Barron, 2000: 5):

1. A fast programming approach which is able to integrate complex components in a professional programming environment.
2. An interface between users and existing systems which can manipulate, customize or automate tasks.
3. A shell oriented to administrative tasks in complex systems.

Online, scripting happens both on the client and the server side. In client-side scripts, various scripting languages are frequently used in web development; however, the most common scripting language is JavaScript. In spite of JavaScript being able to run on the server-side, it is widely known and used on the client-side, as it can maximize the full potential of the resources in client's web browser. Furthermore, due to use of the Document Object Model (DOM), the client's web browser allows scripters to work with existing components (Goodman et al, 2010: 105).

However, the benefits of using JavaScript in client-side, such as distribution of the server's workload, higher performance, and the use of rich graphic user interfaces (for instance, AJAX implementations), are only available online: the code is available for access via a web browser. This feature is not present in server-side scripting because the scripts are located in a (supposedly) secure server and their code is interpreted by the server's scripting processors (i.e. PHP, JSP, ASP.NET, etc.). That is to say, information in server-side scripting is hidden, but client-side scripting's code is exposed.

JavaScript is considered standard in client for web scripting (Deitel and Deitel, 2008: 198) and it has to be interpreted by the in client web browser's built-in engine (JVM). As such, a lot of programming production happens with this kind of technology, which means its source code is widely available. Nonetheless, there is a way of protecting the source code written in JavaScript, which is so widely used. The technique is called "JavaScript obfuscation" and it stops people from being able to understand the program code when they look at it (Byung-Ik, Chae-Tae & Hyun-Chul, 2011: 20). However, there are also several browser plug-ins and extensions that "de-obfuscate" the code, meaning obfuscation is not 100% effective in hiding the source code.

It is extremely difficult therefore, to establish parameters for the protection of intellectual property when it comes to source code written in JavaScript. It could be possible if JavaScript changed its inner model, that is to say from interpreted language to compiled language, but that seems unlikely as it goes against the philosophy behind JavaScript.

2.1.3. **Data Warehousing Online**

A data warehouse is a computing system which provides effective business intelligence solutions for companies, enabling them to improve their efficiency and succeed. (Chenoweth, Corral & Demirkan, 2006: 115). A data warehouse is established within large companies with lots of information to store. In addition, a well-designed data warehouse should follow the definition of normalized tables with a dimensional approach to storing data, in order to achieve good business intelligence results. In spite of their many benefits in business, a common problem with warehouses is that as they expand as more data and more data is added, the hardware infrastructure's ability to process is impaired (Costa et al, 2011: 1). The fact is however, that companies are using data warehouses more and more, to build up their business intelligence.

According to Bhowmick et al, the main problems with data warehousing are lack of productivity and managing historical data (Bhowmick, S., Madria, S., & Ng, W. 2011: 2). The former is due to the complicated processes associated with searching in the chasm that is the web; the information gathered is sometimes in several pieces, and needs to be re-assembled as a whole. The latter arises when trying to locate information pertaining to past behaviors, trends, and results of businesses; information that should be freely available, but in some cases is quite difficult to access.

An important aspect of data warehousing to consider is the quality of data used by the business. Corporations or organizations should initially establish clear policies to ensure that the data generated and stored in the warehouse is of high quality. According to Singh and Singh, the establishment of data quality parameters has implicit consequences on the completeness, consistency, validity, conformity, accuracy and integrity of the data warehouse (2010: 41).

Singh and Singh state in their research that there are 52 causes of data quality problems (2010: 44). Organizations should therefore make themselves aware of these causes if they want to improve their business intelligence. Maintaining data quality has an enormous impact on the "traditional" functions of a business, because several traditional practices must be altered to meet the data warehouse's design requirements.

Organizations do not necessarily need to undergo a complete overhaul in order to implement a data warehouse system. They do however, have to establish clear policies on inner data management, in particular as far as concerns sources, such as Legacy Systems, OLTP/operational systems and flat/delimited files. According to Singh and Singh, the origin of the toughest problems in business intelligence is the data extraction phase.

2.1.4. DOM

DOM (Document Object Model) is defined as both a platform and a language interface that manages documents and their content. According to the DOM activity statement, it is the partnership between the platform and the language interface that produces the new application programming interface, DOM. It gives access to the components of a document and the document itself, and has an object-based approach which can be used with scripting (W3C, 2005).

Today, DOM specification is defined by the W3C, but this has not always been so. The first DOM model came from Netscape, they produced an object model with their browser, Navigator 2. After that, other companies, such as Microsoft, implemented their own versions, which led to problems with browser incompatibility. Some of these problems were solved when the W3C outlined DOM recommendations (Goodman et al, 2010: 503). In essence, DOM organizes objects according to their properties, attributes, methods and events, into hierarchical structure within a document schema. It is important to note that DOM is an object-based model, where objects can be managed via JavaScript and other scripting languages in client-side scenarios.

In terms of software development, there are several well-known DOM models which are used by programmers, for instance the Basic Object Model (the simple DOM), Basic plus Images (specialized for imaging), NN4 Extensions, IE4 Extensions, IE5 Extensions, and W3C DOM I and II.

The advances and improvements that have been made in each model are summarized below:

Basic Object Model	This model is the lowest-common denominator in document object hierarchy. This structure includes: window (1st level); history, document, location (2nd level); link, form, anchor (3rd level); and the last level, in which all the inputs, such as text, radio, checkbox, buttons, etc. are located.
Basic Object Model plus Images	This incorporates the <*img*> tag with properties, event handlers and methods for its generated objects which is a great improvement for enhanced UIs.
Navigator 4-Only	Navigator 4 added a new event capture model related to the keyboard and mouse. A new specification was successfully tested; Dynamic HTML (DHTML). Another, radical addition was made to manage containers of document objects. Those containers were called Layers; however, the W3C did not agree with the <*layer*>tag.
Internet Explorer 4+ Extensions	Its most significant improvement was the management of arrays or collections of objects. Scripting statements such as *document.all* allow easy control of each object by following its *elementID*. This model introduced the managed control of cascading style sheets via scripting, and promoted event bubbling - some events can be reached through the containment hierarchy. However, this last feature was incompatible with other browsers.
Internet Explorer 5+ Extensions	This focused on DHTML behaviors using several CSS features. The concept of HTML Application (HTAs) was coined in this model. New HTML elements with XML syntax appeared and extended functionality, for instance: *hta:application*. HTAs had higher security privileges on client than regular programs, which caused security issues.
W3C DOM	The battle between Netscape and Microsoft made life difficult for Developers. Ergo, the W3C stepped in to regulate. In order to unify DOM criteria, W3C divided its DOM specification in two sections: Core DOM and Compatible Features.

Table 2.1 Improvements on models

Following this, levels of DOM compliance were established. Although some things stayed the same, such as the object collection in the original model (level 0), other things are still not universally available, such as the "layer" concept and properties that facilitate dynamic content, such as *innerHTML, innerText, outerHTML*, and *outerText*. For instance, in event propagation approaches, the W3C DOM (level 2) can deal with the event models NN4 (trickle down) and IE4 (bubble up) in spite of their conflicting relationship. Since the contribution of the W3C, there have been new HTML practices, such as the XML-based syntax, and new DOM concepts, such as techniques for element referencing via the ID of each HTML element, and built-in functions, for instance getElementById() (Goodman et al, 2010: 519).

The impact of these improvements is striking when it comes to writing code to create dynamic content on client-side. For example, when the first version of the University of Nariño's home page was written in 1995, two separate versions had to be constructed in order to cater for the most popular web browsers: Microsoft Internet Explorer and Netscape Navigator, due to there being dynamic client-side content.

Now programming is different; there are still some incompatibilities depending on which browser is used, but not to the same extent as before, as the latest web browser engines try to follow the W3C's regulations. In terms of user interfaces, CSS and AJAX are leading the way in the construction of rich environments with graphics, animations and multimedia content. With the new features within the latest specifications, such as HTML5, developers have access to an alternative for lighting up web content without using industry components such as Adobe Flash or Microsoft Silverlight. Today, web developers have a wide choice of options when writing cross platform web applications.

2.1.5. Threats to JavaScript

The Internet is an open environment in which lots of people interact with information, particularly via the World Wide Web service. Unfortunately, the web can be a hostile territory with security and data privacy risks, due to the proliferation of malware, spyware, viruses, and worms, etc. (Byung-Ik, Chae-Tae, & Hyun-Chul, 2011: 19).

When the web first started, sites were static. Websites at that time only published information for the end user to interact with; therefore, the first websites focused only on allowing users to navigate inside and outside their content. This kind of basic interaction used to be enough; however, over time webpages have become more dynamic. This is where JavaScript comes in.

By definition, JavaScript is an Object-Based Language that can interact directly with the Document Object Model (DOM) of the HTML Engine in client's web browser (Goodman et al, 2010: 195). In this way, most of the objects that a client's web browser creates are part of DOM (Goodman et al, 2010: 505); and those objects are accessible for the client-side scripting. Some years after its creation, the W3C established a standard specification of DOM from a portability and flexibility perspective.

However, when W3C purposed its specification model, several risks were identified concerning the "manipulation" of objects in DOM. Earlier browsers were more vulnerable to malicious code written in JavaScript; today, modern browsers are designed to counter those threats. There are several kinds of dangers which come from client-side scripting, two of which, cross-site scripting (XSS) and cross-site reference forgery (XSRF), will be looked at in closer detail below (Wadlow&Gorelik, 2009: 43).

Wadlow and Gorelik define XSS as an incident in which, “an attacker embeds commands or code in an otherwise legitimate Web request. This might include embedded SQL commands, stack-smashing attempts, in which data is crafted to exploit a programming vulnerability in the command interpreter” (Wadlow&Gorelik, 2009: 43). They define cross-site reference forgery as, “an attack which is similar to XSS but it basically steals client’s cookie from another tab within the web browser” (Wadlow&Gorelik, 2009: 43). The question is how to avoid these types of scripting attacks. One solution could be to increase the web browser security by limiting DOM access to exposed objects; another could be to restrict JavaScript access to the DOM path. On the other hand, a third party software may be needed to achieve this, rather than simply using the web browser. Although both alternatives are viable, restrictions may interfere with client-side scripting performance.

2.1.6. The Future of Open Computing

The definition of free software essentially focuses on the importance of liberty; the philosophy behind it upholds users’ freedom to run, copy, distribute, study, change, and improve software (GNU, 2011). With free software, users have the opportunity to handle software without any kind of restriction. It is often confused with an open source approach, and the terms are used interchangeably. However, a closer examination of the relationship between free software and open source approaches demonstrates that although these terms are close in their meaning there are differences between the philosophies.

Firstly, free software is a philosophy behind software production and usage, and open source approach is a methodology for creating and installing specific software (Stallman, 2011).

In other words, every production process which follows the principles of free software is in fact using an open source approach, but not *vice versa*. Several open source products are not 100% compliant with free software principles. For instance: there are only 9 GNU/Linux *distros* which comply fully with the free software philosophy in the sense that they follow every single FSF principle, even hardware-firmware drivers. Unfortunately, the most common and well-known *distros*, such as Debian, RedHat, Ubuntu, among others, do not belong to this exclusive list because they allow proprietary components to run within them (FSF, 2011).

Undoubtedly, the philosophy of free software has had a huge impact on our society, having great benefits, but also raising some concerns (Schryen, 2011: 130). In fact, at the beginning of the free software expansion, it was thought that it could be the way all future software construction was going, but in reality, free software has proved to be not so simple (Campbell-Kelly, 2008: 23).

In online contexts so far there have only been open source developments. However, there is a new trend in online free software: SaaS (Software as a Service). "SaaS seems to offer a technological prospect in which both proprietary and open source software can flourish" (Campbell-Kelly, 2008: 23). Although Campbell-Kelly predicts the rise and proliferation of SaaS, it is incredibly hard to predict the future of open computing, especially as there are also several leading proposals in proprietary fields for Internet-based application construction. In fact, proprietary solutions such as Microsoft .NET and Oracle Java EE, have grown rapidly due to their flexibility, easy-to-use policies, and performance.

2.1.7. HTML: Past, Present and Future

Although the Internet has been developing over several decades, in the recent years the WWW service seems to have grown the fastest, producing the greatest technological impact. The World Wide Web's roots go back before 1979, following contributions in the field of electronics and computing. Later, in 1989, Tim Berners-Lee was consulting for CERN (The European Organization for Nuclear Research) where he has the idea to link arbitrary nodes (W3C, 2004).

At the end of 1990, the first applications were created; these applications included a browser, an editor, and a first client-server environment. Then, the first-ever web server was created in 1991 at the CERN laboratories. Its address was http://info.cern.ch/ (it is still available although there is a new version of the site), and a year later, a second web server was created (outside Europe) in Stanford University (CERN, 2008).

In the first HTML draft, the author outlined his initiative for linking information around the globe using HTML (Berners-Lee & Connolly, 1993). This draft contained detailed information about the features of language, its specifications, and syntax. HTML is simply a language, which would be completely useless without a protocol to manage the communications and links. This protocol is called HTTP (HyperText Transfer Protocol).

Nowadays, HTML is widely used in web development; in fact, the W3C, which was created in October 1994, still promotes and specifies its usage through HTML standards (W3C, 2010). Today, every web based development must deal with HTML in one way or another, sometimes when using XHTML, or even when using new rich content approaches such as Adobe Flash or

Microsoft Silverlight. Although these are newer developments, the basis of the original implementation still remains, and the aforementioned applications are all embedded in HTML.

Arguably, HTML still plays a vital role in development. One of the most common criticisms of HTML is its weakness in GUI, however this issue can be solved using CSS and JavaScript, although it is possible that HTML5 will solve this problem altogether. HTML5 is a work in progress with substantial improvements enabling it to respond the demands of new GUIs with flexibility, portability, and scalability (W3C, 2011). Currently however, web designers and programmers can use powerful APIs and class libraries, such as AJAX, to create rich user interfaces, and these resources do not strictly belong to the initial HTML proposal.

In spite of the fact that the HTML5 initiative originated outside of the W3C, the Consortium has its eyes on its development. Predictions show that HTML5 most probably will be able to function without the use of complements, such as Adobe Flash and Microsoft Silverlight, and in that sense enable a better user experience. Clearly HTML will continue to play a key role in development in the future as a language, but with more added functions.

2.1.8. JavaScript and Interpreted Code

Certain programming languages need to work alongside an agent which is able to execute their code; that agent is called an interpreter. In the specific case of JavaScript, web browsers have interpreters instead of compilers, which process the lines of code written in JavaScript (Deitel&Deitel, 2008: 199).

Due to JavaScript's design, its scripting language must be hosted by a specific container. This is a prerequisite because JavaScript was conceived as an embedded language within HTML, so its final container is a web page. Tim Berners-Lee's original idea was that web pages should be available for access everywhere; and this was achieved, in terms of static content, thanks to the HTML engine in browsers (Goodman, 2010: 16). Over time however, the classic web page has become more dynamic, following the inception of JavaScript in early 1990s.

With a new embedded language within web pages, a compilation environment was roundly rejected due to the heterogeneous hardware/software technology of the devices which were to read the web pages at that time (mostly computers). In compiling code in such a way, the engine would have had to create an object/executable code that could run on every single device. Today, the situation is much more complex than before, because there are various different devices which are able to access web resources (desktops, laptops, tablets, PDAs, cell phones, game consoles, TVs, etc.) Each one of these has a different hardware architecture and a different operating system/web browser.

In response to this, JavaScript was designed as an interpreted language, but why designers decided to use interpreted code instead of compiled code deserves closer analysis. According to the definitions of interpreted and compiled codes in computer programming, both have the same objective: to execute program code, but their approaches are different. By definition, a compiler translates code into a form in which it can be executed by a computer (Aho et al, 2006: 1); unlike an interpreter which is a language processor that appears to directly execute the operations specified in the source program on inputs supplied by the user (Aho et al, 2006: 2).

To clarify, a brief comparison between interpreted code and compiled code in web environments is shown below (Chone, 2009):

Criteria	Interpreted Approach	Compiled Approach
Model	Asynchronicity	Client/Server inside the browser
Languages	JavaScript, ECMAScript	C#, Java
Final Production	Embedded Code (Source Code itself)	Assemblies, Object Code
Execution Platform	Language's Interpreter in browser (Engines)	Frameworks
Execution Place	Client	Server
Source Code Access via browser	Available	Not Available
Performance	Higher (it depends on client's resources)	High (it depends on server's resources)
Security	Low profile	High profile

Table 2.2 Comparison between compiled and interpreted code

Looking at the features of these two approaches, it stands to reason that JavaScript was designed with an interpreted approach to take advantage of the computing capabilities on the client's web browser. In this vein, JavaScript's strengths are the events associated with the asynchronous life of a web page when it is fully loaded on the client's web browser. These events include: data management with focused tasks such as validation, caching and portability, the enhancement of the graphic user interface with multimedia support and special effects, among others.

There are enormous advantages to JavaScript because of the way in which it removes the server's workload. Tasks associated with input validation, navigation, and direct interaction with

end users via graphic user interfaces have been powered using JavaScript. However, it must be said that its exposing of source code in web browsers makes it a threat to security, which is its major disadvantage. JavaScript embedded on web pages and can pose a security risk, however it only has restricted access to the operating system's resources. In spite of these restrictions, there are several threats to the security of the operating system on client, for instance: malware, spyware, even viruses, though it is possible that the server-side scripting model could decrease the proliferation of these threats.

2.1.9. XML Web Services

The authors, throughout their career at the University of Nariño, have worked with several different web development technologies. The professional construction of web-based solutions falls into one of two main areas: client-side scripting and server-side scripting. The latter in particular has undergone many changes throughout its evolutionary process; the focus of the authors' work has been with CGI, "Classic" ASP, and ASP.NET since its inception in 2002.

One of the many reasons Microsoft .NET technology was chosen for working with professional web development, is its high level of productivity and its ease of use. It is possible to create complex computing systems to meet modern requirements in less time than with other applications. When looking at Service-Oriented Architecture (SOA), it soon becomes obvious that Microsoft .NET offers a powerful and complete platform to develop and install web services. SOAP (Simple Object Access Protocol), which works within the SOA approach, has marked an

important milestone in the history of web development due to its extensibility, modularity, portability, robustness, and scalability.

SOAP led the boom in the XML Web Service technology. In particular, Microsoft's construction of distributed systems online was made possible by establishing a solid platform with XML Web Services. Wolter stated that an XML Web Service is a powerful way to create distributed systems online by using open standards of communication and collaboration The main features of XML Web Services include the use of standard protocols, such as SOAP over HTTP. Communication between clients and the web is simple, thanks to the parameters specified by WSDL (Web Service Description language), which is the easy way to find resources via UDDI (Universal Discovery Description and Integration) and allows new users to access online resources (Wolter, 2001).

XML Web Services are sometimes referred to as "internet ghosts", because they do not need graphic user interfaces. These "faceless" software units are designed to be consumed by multiple clients, no matter who or where they are. In the creation of an XML Web Service, everything starts with a good design, normally using an Object-Oriented Approach, and it is fairly common to use encapsulation techniques based on the creation of class libraries.

XML Web Services are transforming the way web-based solutions are built. An XML service approach allows the creation of "units of code" which can be considered reusable software assets. One of the huge advantages of using XML Web Services is that several types of clients can be included. Using XML as the language for organization and display increases portability, so much so that today, the most common devices that can consume these services are mobiles devices

such as cell phones, tables, and PDA's, among others (McDonald, 2010: 1484). As such, the same web service can be used with a web application, a desktop application, and a mobile application.

2.1.10. Design Patterns

Design patterns are tools that facilitate the construction of good quality reusable software. Each pattern outlines a problem that occurs repeatedly in certain context, and equally gives the solution to the given problem, and can be used several times. Generally, the elements included in a design pattern are: the name, the problem, the solution, and the consequences (Gamma, 1995: 3).

The name of the pattern encapsulates the problem and solution in one or two words. A good name allows for a design with a higher degree of abstraction, and speeds up the communication process between software designers. The problem specifies when a pattern should be applied; it explains the problem and its context. It could refer to class structures or objects that repeatedly cause inflexible designs. The solution is composed of the elements that make up the design, the relationships between them, their responsibilities and interactions; the solution defines a particular design or implementation. The consequences are the predicted results of applying a pattern; although the consequences are not normally discussed or known at the time of design, they are critical when choosing a solution, and to understand the pros and cons of applying the pattern.

Nowadays, there are many different design patterns to choose from. In e-commerce especially there has been a proliferation of meticulously thought through, well-defined, design patterns, and there is one design pattern in particular that can be implemented in any application that deals with e-commerce: the order list pattern.

The order list design pattern is one of the most famous and widely-used design patterns when making purchase orders. This pattern contains classes of customers, products and the relationship between these actors with the class *Order* and *OrderDetail* (Jewett, n.d.). This is the basis for constructing an online shopping cart. This pattern is widely seen on websites which offer end users the opportunity of buying something.

2.1.11. Objects on Web Pages: A JavaScript Perspective

By definition, JavaScript is a programming language, and more specifically a scripting language (Deitel&Deitel, 2008: 198). Scripting languages are designed to be flexible to make software development easier. Due to their flexibility, scripting languages such as Perl, PHP, Python and of course JavaScript, are used in both clients and servers (Wong, 2007: 57). JavaScript came from Netscape Communication Inc. in the early 1990s, and since its inception, other leaders of industry such as Sun Microsystems and Microsoft Corporation have contributed to its enhancement.

According to Bucknall (2011) JavaScript has the following distinguishing features:

- JavaScript is an object-based language, unlike Java which is a 100% object-oriented language.
- It inherits its basic syntax from Java programming language and it complies with the ECMAScript approach; which gives it one of the most striking features of its syntax, that it is case-sensitive.
- JavaScript is a weakly-typed language. It handles the variable/object concept.

- Its code is available to end users via web browsers, as it is an interpreted language rather than a compiled language.
- It is embedded within HTML. In this vein, web browsers offer ways to include JavaScript by integrating their Document Object Model with the scripting language.

As can be concluded from the first and last features stated above, JavaScript is able to detect "objects" within a webpage (Goodman, 2010: 49). Objects are everywhere within a webpage; some of them are part of the basic text, containers such as divisions and paragraphs, images and graphic elements, and some are the most important data management resources: web forms.

Nowadays, web-based data management is a common feature of online business; online financial transactions for instance, are described by data through web forms. However, this mechanism is not exclusively used for dealing with monetary transactions; online information management is also handled using web forms, for instance: academic records, medical patient history, and messaging systems, among others. Graphic objects are also an essential component when constructing interactive user interfaces; AJAX technology and the latest HTML specifications (i.e. HTML5) for instance are both graphic object developments. These objects are easy to use, thanks to the object model which is available for use with JavaScript. The implementation of graphic objects has led to improvements in user experience with innovative, portable and rich user interfaces via JavaScript.

In conclusion, the benefits of JavaScript's object-based approach online are obvious. Two main aspects stand out: data management which is able to carry out specific tasks such as validation, caching and portability; and the enhancement of the graphic user interface with multimedia support and special effects.

2.1.12. AJAX and SOA

Nowadays, successful integration and even distribution using web service technology is more user friendly than ever, and this is not because the technology itself is better, or simpler, but because its tools and environment have undergone significant advancement (Rosen et al, 2008: 8). This is, at least in part, thanks to SOA, or Service-Oriented Architecture, which is composed of two main elements: approach and implementation. The former concerns service orientation's theoretical background, and the latter is related to the set of strategies which can be put into practice. It is widely known that the most common application of SOA is in web services.

According to the W3C, a web service is a method of communication between computing entities over the web; it is a software system that works over SOAP and uses an XML format to manage information (W3C, 2004). Advances have given rise to the development of software packages which help users comply with the SOA principles, in terms of performance, scalability and robustness. Thanks to web services, software construction has changed from traditional web page creations into rich Internet applications, a phenomenon referred to by some academics and industry members as Web 2.0. The idea behind Web 2.0 is based on enhanced interaction with users; user experience is enriched via flexible spaces where users can interact with interconnected systems (Deitel&Deitel, 2008: 53). Taft stated that, "developers are using AJAX to build Rich Internet Applications as front ends to SOA". In addition, Taft notes that the success of AJAX is based on the consumption of SOA services which enables enterprise mash-ups (Taft, 2006: 18). It stands to reason then, that AJAX is just a tool with which a SOA approach can be used. The relationship between AJAX and SOA is close in the sense that AJAX is a programming tool (some

authors have called it a framework, module, even component) which allows the principles of service orientation to be implemented, thanks to the integration with web services.

AJAX was originally designed to handle information asynchronously; however, over time developers have tipped it towards graphic rich explosion. In this vein, most AJAX environments focus their attention on the user graphic interface. It can be argued that although AJAX is successful in terms of information management, in many cases AJAX also requires a new user interface.

2.1.13. Using Cookies via PHP

Cookies are an in-client mechanism where information can be stored. With PHP, which is a server-side script, it is possible to carry out the following operations using the client's cookies:

1. Create cookies
2. Read values from cookies

The following are some practical examples:

Action:	**Create Cookies**	
	Sending a signal to create a cookie with PHP is simple using the *setcookie()* function. This function sends all the necessary information concerning a particular cookie. All the parameters are optional except for $name.	**<?php** setcookie($name,$value,$expire,$path, $domain,$secure,$httponly); **?>**

Table 2.3 Creating Cookies

Action:	Read values from cookies	
	There are several options when reading the value of a cookie in PHP. One of them is by using $ _COOKIE superglobal variable, which is always available in the code.	**<?php** *//create cookie* setcookie("mycookie","our cookie"); *//reading* if(isset($_COOKIE['mycookie'])) { foreach($_COOKIE['micookie']as$name=>$value) { echo"**$name : $value** "; } } **?>**

Table 2.4 Reading values from cookies

The parameter expiration is essential when "destroying" a cookie and its values.

2.1.14. Web-based Applications: AJAX Model

Web application development has undergone a rapid evolutionary process; at its inception, scientists interacted with research-based hypertext, but now web applications are more complex than simple static web pages. They have been hugely successful in both commercial and information processing contexts. Undoubtedly, web applications owe their growth and expansion to the dynamic languages on the server and client sides.

The fact that the development of computing solutions now tends to be online, and therefore clients only need to use a web browser, has drastically changed the way that people perform daily activities. Most applications have been developed for online use, and as such, local storage is no longer necessary. Information is stored on the servers that host web applications, and some

information is stored not only in one server, but in servers' farms via distributed systems. However, this software construction model does have some issues, for instance, the postback phenomenon.

Postback is when client and server are not completely in sync and the client therefore experiences a time delay. When a client machine loads a web page, the end user can interact with it using the client machine's computing resources (memory, processing time, graphics, etc.). However, when the end user incites a control on the website, the client sends a request to the server and the server responds to the request through the construction of a brand new page, and this takes time. This gap between the request and arrival of information is called a postback.

A single user interface can commonly include several controls which need postbacks (because certain information must be recovered from a database) and traditionally, each interaction with these controls generates a new full-page load. Asleson and Schutta call this, "The Fundamental Problem" and according to them, each interaction with the end user in the graphic interface producing a complete new page prevents the development of fast web applications (Asleson&Schutta, 2008: 12).

However, a solution to "The Fundamental Problem" has been developed: AJAX. The technology, based on Asynchronous JavaScript and XML, was designed to improve communication between server and clients based on asynchronous requests (Deitel&Deitel, 2008: 591). AJAX creates a layer between clients and server which is capable of partial page updates. It makes web applications seem faster to end users, as they see the complete web page without the need to re-load again with each interaction.

Traditional applications (those that need to be installed, set up and maintained on individual computers as desktop applications) have been losing ground with the emergence of AJAX and rich

Internet applications. The latter is simpler to use; it merely requires a web browser in client machines as everything is available through the web application. One of the most conspicuous examples of evolution in this field is Cloud Computing.

An example of this kind of technology is Microsoft Private Cloud. Several complex applications and services can be running in the cloud, such as Microsoft SQL Server, Microsoft Active Directory and Microsoft SharePoint Services, among others, and these can be used and managed just from a web browser, no client software is required. All the information is stored outside the client side, virtualization means better efficiency, performance, security, and sometimes, a decrease in costs (Microsoft, 2011). As developments in this area are currently unfolding; arguably, applications that only require Internet access will continue grow, especially as this approach appears to be more economically feasible than traditional applications.

2.1.15. Web Forms and Workflows

The way that language interacts with the end user, in all of the HTML specifications, is through HTML forms. According to the W3C, a HTML form is a part of a document, a web page, which describes a collection of objects in order to facilitate the final user's input (W3C, n.d.). HTML forms deal with information, by doing calculations and processes with data provided by users. HTML forms allow more profound interaction in addition to the basic specification of the language; without HTML forms, the language lacks mechanisms to work in request-response environments.

The use of HTML forms has allowed the development of web applications that interact with database management systems, which were built to handle large volumes of information.

These systems are available today, in education, transportation, health, public services, finance, online entertainment, research and science, etc. Commerce is one of the most common contexts in which HTML forms are used. In business, the flow of online information takes place inside a HTML form. Most of the interactive HTML substance in a browser exists in text fields, buttons, checkboxes, option lists, etc., representing valuable business information (Goodman et al, 2010: 153).

A common business term is "workflow", a sequence of defined tasks which lead to a goal. Roger et al stated, that workflow within a business involves not only machines, but also tasks which must be carried out by personnel (2010: 2898). The flow of information within a business that controls certain workflows can be managed via a web application, and more specifically it could be managed through HTML forms. When designing organizational processes and workflows, relationships between different perspectives or points of view; such perspectives or views include: the customer, the business, the employee, and the product that needs to be established (Roger et al, 2010: 2898). Each perspective must be able to handle its own information and send/retrieve data to/from others, and this communication can be addressed by HTML forms. This is called a workflow management system, which deals with routes of information, and roles and regulations, and such a system generally is supported by IT.

Following new trends in information management systems, the most common place to have such a system is online and cloud computing. HTML forms, or similar technology will be available in future business workflow developments. New technological requirements are appearing in business, such as 2D barcodes and rich GUI's. A possible answer to these requirements is Xforms: the next generation of HTML forms (Gingrande, 2006: 14). The Xforms initiative may be used in

workflows in business environments, due to its flexibility and its capacity to interact directly with XML Schemas and specifications oriented business rules (W3C, 2011).

2.2. WORKSHOPS

2.2.1. 1st Draft of a Proposal: Web-Based Academic Grading System

LOCATION: The fictitious University OLI Tech

BUSINESS: Higher Education in Technology

BRIEF DESCRIPTION:

The University OLI Tech's main goal is the formation of computing professionals at undergraduate level in the five disciplines stated by ACM Computing Curricula 2005, which are: Computer Engineering (CE), Computer Science (CS), Information Technology (IT), Information Systems (IS), and Software Engineering (SE). Each discipline has its own courses, and there are some courses which belong to more than one discipline. Each discipline should be studied for 4 years. As with all universities, the OLI Tech has its own faculty staff, and of course, students.

The university needs a web-based information system in order to manage grades. The system should manage diverse users with different profiles such as student, professor, and administrator. Furthermore, the main purpose of the system is to enable students to be graded for each course; this must be done through a database management system. Finally, the system allows students to query their grades, produce reports, and facilitates academic administrations for the managers OLI Tech.

WEBSITE'S MAIN STRUCTURE:

1. Home
    a. About OLI Tech
    b. Our Mission
    c. Faculty Staff
2. Academics
    a. ACM Computing Curricula's Disciplines
        i. Computer Engineering
            1. Syllabi
        ii. Computer Science
            1. Syllabi
        iii. Information Technology
            1. Syllabi
        iv. Information Systems
            1. Syllabi
        v. Software Engineering
            1. Syllabi
    b. Courses
    c. Apply online
3. Grading System (this is the core of the web site)
    a. For Administrators (login required)
        i. Education management (assign professors to courses, setting up the syllabi)
        ii. Enabling a study season (summer, fall, winter, spring)

iii. Reports

b. For Professors (login required)

i. Define Grading Criteria for his/her Courses

ii. Save Grades

iii. Reports

c. For Students (login required)

i. Query Grades

DATABASE STRUCTURE:

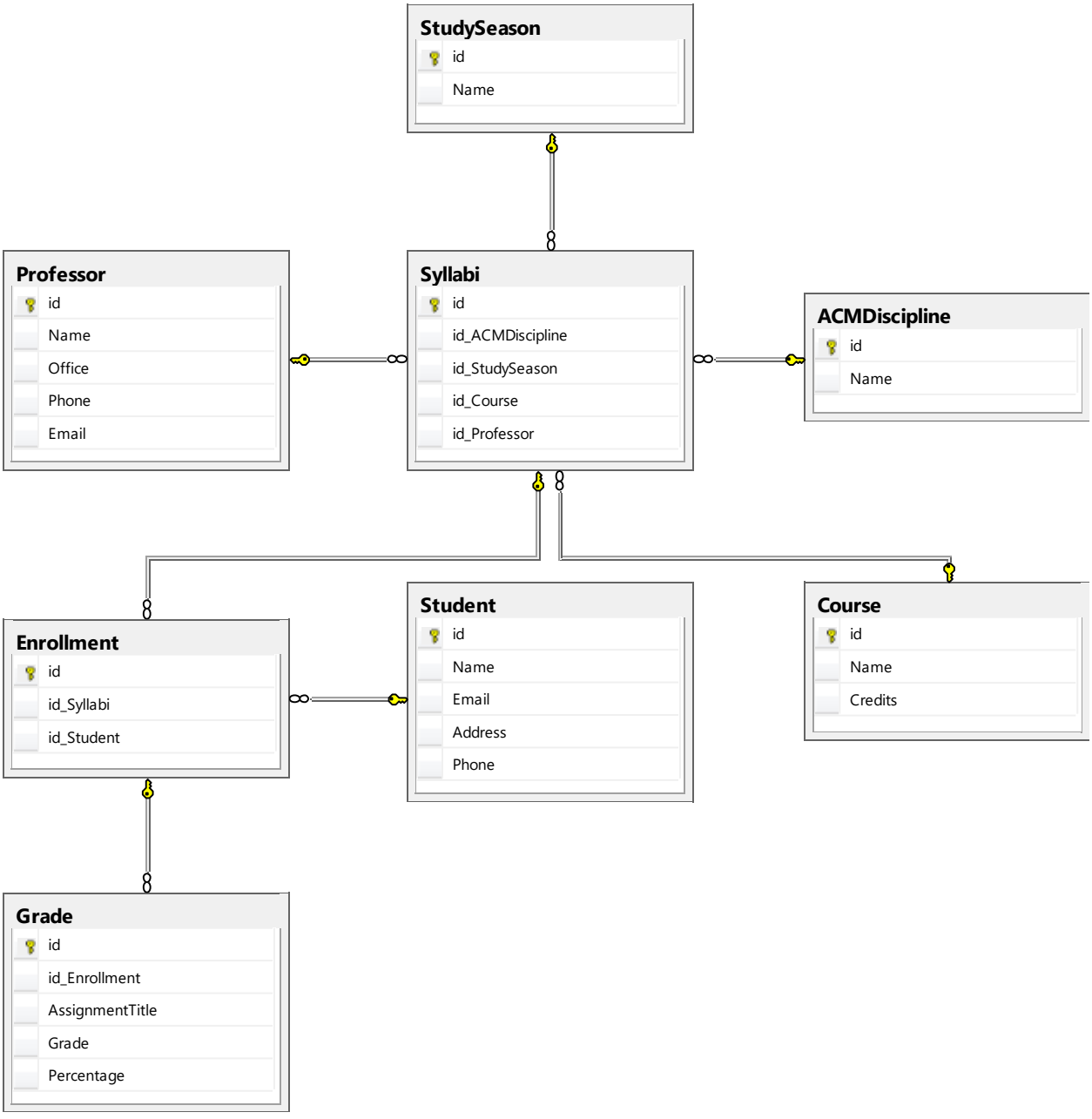


Figure 2.1 Database structure

2.2.2. Advanced PHP and Database Issues

This article further discusses the issue of PHP and databases on the Web. The importance of databases and their use in any application as a data store, or as a tool to drive the application, cannot be understated. In that respect, the web is no different from any other environment from the point of view of a business engaging in eCommerce. The web offers its own unique challenges in so far as application development is concerned, and with larger systems it tends to be even more complicated, particularly with the combined use of so many technologies needed for all the interfacing. However, there are steps that can be taken to try and keep this under control. In this article, the use of PHP in implementing a database interface model to help reduce complexity and increase maintainability in large software systems will be addressed.

PHP has been used as a scripting tool, either by embedding PHP within XHTML, or writing stand-alone PHP scripts for interfacing, and for database connectivity and manipulation. As previously mentioned, PHP is useful because it is versatile, extensible and supports a wide range of database protocols. However, PHP was initially designed to be a quick and useful tool for embedding in XHTML pages to make a given application more dynamic. The examples we have been using are small and programmed in this fashion, e.g. mostly embedded PHP code to provide database connectivity and Data Manipulation Language (DML) statements, (SQL in this instance), for the web pages. However, in any large scale web based application, these techniques simply do not scale-up. As the application size grows, the complexity and maintainability of the web pages as a whole increase with duplication of DML statements throughout the XHTML code. The problems are partly due to the various technologies and stateless nature of the Web, but there are techniques which can be used to combat the problem.

When a web based application attains a certain size, the paradigm being used to build it must be changed. The basic design rules for a couple of XHTML pages with some embedded PHP should be vastly different from a large scale eCommerce application. PHP was designed to be embedded in XHTML, and as such works well, but it becomes an inhibitor as the application grows in complexity (Wong 2003). Embedding DML (SQL) statements within XHTML also causes the same problems as applications grow. It becomes difficult to control access to the database through a central point.

Model View Controller Design Pattern

A design model that can be used to help control the complexity and increase the maintainability of a web based application is called the Database Interface Model (Wong 2003). This is also based on the Model View Controller (MVC) design pattern (Marston 2006, Pattern 2005). "The primary goal of MVC is to isolate User Interface changes and prevent them from requiring changes to the domain logic of the application" (Pattern 2005). The User Interface and domain logic have different rates of change, and change is precipitated by different factors. Thus, it makes sense to isolate them so a change in one does not necessarily mean a change in the other. The MVC model maps the traditional (Input, Processing, Output) roles of the (Controller, Model, View) components, whose relationship is shown below. This pattern decouples the business logic from the presentation, and provides a common place for the pre-processing and post-processing of requests.

Basic MVC relationship (Marston 2006)

A brief explanation of the components is:

Model

A model can be represented as an object representing business data and activity, e.g. a database table and DML. The model is the business process layer

- Models the data and behavior behind the business process
- Responsible for:
    - Performing DB queries
    - Calculating the business process
    - Processing orders
- Encapsulates data and behavior which are independent of presentation (Sun 2006)

View

Presentation layer

- Displays information according to client types
- Displays result of business logic (Model)
- Not concerned with how the information was obtained, or from where (that is the responsibility of the model) (Sun 2006)

Controller

- Serves as the logical connection between the user's interaction and business services
- Responsible for making decisions among multiple presentations

When a request enters the application through the control layer, it (the controller) decides how the request should be handled and what information should be returned (Sun 2006).

A controller is the means by which the user interacts with the application. A controller accepts input from the user and instructs the model and view to perform actions based on that input. In effect, the controller is responsible for mapping end-user action to application response. (Marston 2006).

There are a number of ways of implementing the MVC model for web applications, and following is a simplified version of the one introduced by Marston (Marston 2006). A diagram showing the details of this implementation is shown below.

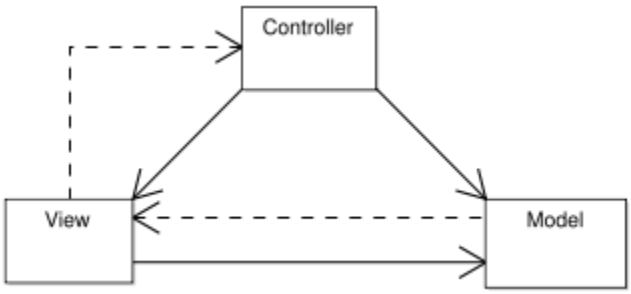


Figure 2.1 Simplified MVC Pattern

Simplified implementation of the MVC pattern

In the implementation shown above the model for a series of business entity classes, where each business entity will correspond to one table in the database, will be put in place. These business entity classes will contain all the logic and DML statements to physically manipulate the specific table in the database to which the business entity corresponds. The controller will be implemented as a series of scripts, and in this simple implementation, there will be one controller script basically controlling the interface with each of the tables through communication with the corresponding business entity class.

The resulting 3 tier system is in the diagram below:

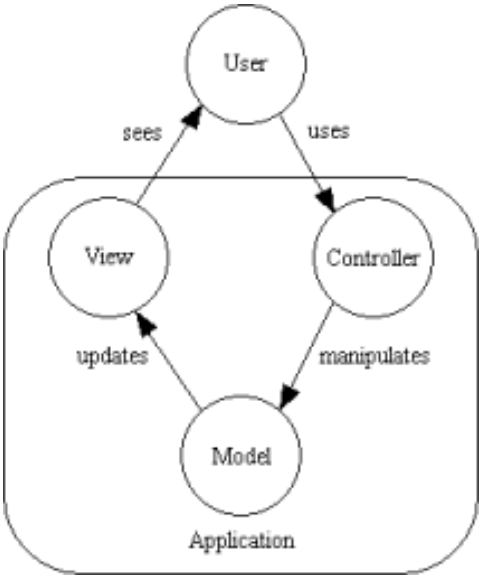


Figure 2.2 MVC Pattern

Utilizing Objects in PHP

The solution that will be constructed to implement the MVC pattern will utilize Classes and Objects in PHP. It will not be too complex, but a basic understanding of PHP is needed. Below is a slightly adapted version of a simple tutorial on using basic objects and classes in PHP. The following is a rendition of a tutorial (Zend 2004), see references section below

Classes are considered by some to be one of the hardest components of PHP to understand, but the reality is that classes are extremely straightforward and easy to use. Problems often originate from programmers not having worked with Object-Oriented Programming (OOP) before. They are little more than a container for variables and functions affecting those variables, but they can be very useful for building small components, almost miniature programs. The difference is that they do not have to be loaded as separate URLs, and they can be easily plugged into most scripts with just a requirement or include statement at the top. A class describes an “object”. An

object is a collection of smaller objects, just like in a class. For instance, in describing a door lock as a class it might contain:

variable $Bolt
variable $Position
functionTurnKey($Direction, $Distance)
functionCheckLock( )

TurnKey would accept a direction value that would in turn determine if the bolt were locked or unlocked. Once the key is turned far enough in either direction ($Distance) the Bolt will either set or clear. The $Bolt variable shows the current state of the door lock, and $Position is used to track how far the key has been turned in any direction. Obviously this could all be done with regular variables and functions, and if only one lock in the code was needed, it would probably work just fine. However, normally more complicated information needs to be stored as classes, continuing with the lock metaphor, more than one lock. Or more details about the lock, or extra information, such as the door in code too. And the lock would just be a subset of the door. Perhaps a whole house is being described. There might be more than one door, and each door certainly needs a lock. This shows just how complex object-related problems can get.

By using classes, each object is defined once, and it is also included in several scripts. That way the object will function the same in all of the scripts. Its good practice to write classes in a separate file usually, that way if it has to be altered it only has to be changed in one location. A good tip is to make full use of PHP's include and require functions, they have many uses.

Class Structures

Below are instructions on how to create a simple page object. The page object encompasses a few of the basic page necessities in building a HTML page. Although there are many more complicated functions that can be done with this object, these are the essentials. Firstly, it is important to examine what the basic elements of an HTML page are. First, the open and closing tags, <HTML> and </HTML>, these will need to be included in the output. Also, the page Title, Keywords for the Meta-tag field, and naturally the main content, or body. Now using these basic components this is how to build a class.

Firstly, the class itself:

```
<?php
        class Page {
        }
?>
```

This is a basic class, without variables, or functions. All class structures look the same with the exception of the "Page" name. Every class/object is assigned a name for reference, it is important to know this when creating new copies of the object, so it is advisable to pick something straightforward and sensible.

Next, the variables are added, a title for the page, keywords, and a content body:

```
<?php
        class Page {
                var $Title;
                var $Keywords;
                var $Content;
}
?>
```

The class at this point is fully functions and ready for use, however it is best to add some functions to make it easier to work with. The first function should be to build the output HTML

which will be called “Display”. Then some simple functions for displaying the Title and Keywords, and also a function for setting the content.

The final class code is:

```
<?php
class Page {
	var $Title;
	var $Keywords;
	var $Content;
	function Display( ) {
		echo "<html>\n<head>\n";
		$this->DisplayTitle( );
		$this->DisplayKeywords( );
		echo "\n</head>\n<body>\n";
		echo $this->Content;
		echo "\n</body>\n</html>\n";
	}
	functionDisplayTitle( ) {
		echo "<title>" . $this->Title . "</title>\n";
	}
	functionDisplayKeywords( ) {
		echo '<meta name="keywords" content="' . $this->Keywords . '">';
	}
	functionSetContent( $Data ) {
		$this->Content = $Data;
	}
}
?>
```

At first glance this appears simple and in reality it is. It is basic PHP code wrapped in a class, however some things do need to be highlighted. VAR - All variables declared in the class must be placed at the top of the class structure, and preceded with the VAR statement. $THIS - $this is a variable that indicates the current object. That way, the object knows how to find itself while running functions contained within it. For instance, $this->Keywords gets the data from the

$Keywords variable in the object. Also, when using variables contained within a class, the $ cannot just be used to reference them, it must be used to reference the object itself.

It is best to save this class file before continuing to anything more complex. A following expel is based on this structure. The following is this basic class using normal script:

```
<?php
include "page.class";
$Sample = new Page;
$Content = "<P>This page was generated by the Page Class example.</P>";
$Sample->Title = "Using Classes in PHP";
$Sample->Keywords = "PHP, Classes";
$Sample->SetContent( $Content );
$Sample->Display( );
?>
```

The include statement is used to bring in the class (page.class) from its external file. The “new” statement is used to create a new copy of the object so it can be worked with. The new copy is stored into a variable called $Sample. Then some variables are set and the objects functions named. For instance:

$Sample->Title = ... to set the content of the $Sample objects $Title variable.

$Sample->Keywords = ... to set the content of the $Sample objects $Keywords variable.

```
$Sample->SetContent( $Content);
$Sample->Display( );
```

To call the functions SetContent() and Display() of the $Sample object the SetContent() function requires a parameter to be passed. In Object oriented terminology, the variables of a class are called the attributes, and the functions are its methods.

At this point the PHP code above should be saved into a file called classes.php, and can be run to test the output. This has already been done, and the code can be viewed by following the URL http://student.ohecampus.com/projects/teacher2/in/sem7/classes.php. The output does not look big, but in the browser from the drop-down menu, at the page's source, all the output exactly as dictated by the functions of the $Sample objects functions can be seen.

For further and more comprehensive tutorials on utilizing objects and classes in PHP, look at the tutorials from the following references, Fuecks (2004), PHPDeveloper (2006), and the PHP online manual at PHP Manual (2011), see references below.

Implementing the MVC pattern

Based on this knowledge of objects in PHP, it is possible to implement a simplified version of the MVC pattern for the flamingo system. The Flamingo database has only one table called 'prodList'. The table has the following fields

prodCodeProduct code – Text field 10 characters long
prodNameProduct name – Text field 30 characters long
prodDescProduct description – Text field 60 characters long
prodPriceProduct price – A number field, 10 digits long

The first thing to be done is implement the code to connect to the database, and place this in a small file called db.inc. The code is shown below:

```
<?php
$dbConn = NULL;
$dbUser = "";
$dbPass = "";
functiondbConnect($dbname) {
	global $dbConn, $dbUser, $dbPass;
	if (!$dbConn) $dbConn = odbc_connect("flamingo",$dbUser, $dbPass);
	if (!$dbConn) {
		return 0;
	}
	else {
		return $dbConn;
	}
}
?>
```

This contains the code to connect to the database, which was removed and placed in a standalone file. There is no need to place it in the business entity classes, as it would be duplicated for each table; instead its best to include it at the beginning of each of the controller scripts. As such, any modifications to this can all be done in one place. The database has only one table, therefore not very much is being saved, but implementing this architecture makes the application scalable.

Following is the code in db.inc.

The lines

```
$dbConn = NULL;
$dbUser = "";
$dbPass = "";
```

Only the global variables that the database connection statements will use need to be defined: the dbConn (database connection object) and the username and password associated with the database. The actual connection statements are placed in a function dbConnect(). This means the function can be named after any of the business entity classes, passing on only the name of the database to be opened.

Before making the connection, a test should be made with (!$dbConn) to see if the database connection is already established. If the connection is already established, it is not opened again, otherwise another attempt should be made to establish the connection. If successful, the connection object is returned, otherwise a zero (0) is returned which can be used in a Boolean condition.

Now, an embedded PHP and SQL code were provided to retrieve a single record from the flamingo database, as well as the code to update a single record. The MVC pattern will be implemented to provide the same functionality. All the code with embedded DML (SQL) statements need to be stripped, and placed within appropriate business entity classes. Since there is only one table, there is only one business class, and this can be put in a file called prodlist.class, corresponding to the name of the database table. The code from the prodlist.class file can be seen below.

```
<?php
classProdList
{
        var $dbname = "flamingo";
        var $tablename = "prodList";
        var $prodCode = "";
        var $prodName = "";
        var $prodDesc = "";
        var $prodPrice = 0;
        functiongetRecord($code) {
```

```
        $sqlQuery = "select * from " . $this->tablename ." WHEREprodCode='" . $code .
"'";
        global $dbConn;
        if (!dbConnect($this->dbname)) return FALSE;
        $dbResult = odbc_exec($dbConn,$sqlQuery);
        if (odbc_fetch_row($dbResult)) {
                $this->prodCode = odbc_result ($dbResult, "prodCode");
                $this->prodName = odbc_result ($dbResult, "prodName");
                $this->prodDesc = odbc_result ($dbResult, "prodDesc");
                $this->prodPrice = odbc_result ($dbResult, "prodPrice");
                return true;
        }
        else {
                return false;
        }
}
functionupdateRecord() {
        $sqlQuery = "update " . $this->tablename ." set ";
        $sqlQuery .= "prodName='" . $this->prodName . "', ";
        $sqlQuery .= "prodDesc='" . $this->prodDesc . "', ";
        $sqlQuery .= "prodPrice=" . $this->prodPrice . " ";
        $sqlQuery .= "where prodCode='" . $this->prodCode . "'";
        global $dbConn;
        if (!dbConnect($this->dbname)) return false;
        $dbResult = odbc_exec($dbConn,$sqlQuery);
        if (!$dbResult) {
                return false;
        }
        else {
                return true;
        }
}
// Standard get and Set methods
functiongetProdCode() {
        return $this->prodCode;
}
functionsetProdCode($code) {
        $this->prodCode = $code;
}
functiongetProdName() {
```

```
		return $this->prodName;
	}
	functionsetProdName($name) {
		$this->prodName = $name;
	}
	functiongetProdDesc() {
		return $this->prodDesc;
	}
	functionsetProdDesc($desc) {
		$this->prodDesc = $desc;
	}
	functiongetProdPrice() {
		return $this->prodPrice;
	}
	functionsetProdPrice($price) {
		$this->prodPrice = $price;
	}
}
?>
```

Listing 1, prodlist.class

All the statements in this listing are wrapped within a class definition - class ProdList.

This is followed by a number of variable declarations, corresponding to the database and table names, followed by the variables corresponding to the table columns. These are used to hold the data which is retrieved from the database, or put there from an input form to be inserted/ updated into the database. There are two main functions getRecord() and updateRecord() which are used to get or update a specific record from the database.

Following this, there are a number of get and set type functions which provide a standardized interface for retrieving or updating the objects' basic fields (corresponding to the database fields). This a fairly standard way of accessing these types of variables within an object.

The two unctionsgetRecord() and updateRecord() contain the basic code embedded in the PHP scripts. However, the code here is slightly adapted to the object environment. The ProdList class then contains all the DML code to manipulate the data that needs to be retrieved and stored to the database according to the business rules. This has now been separated from the connectors, which will be examined, and does not need to be duplicated in any other scripts which might require retrieval or the updating a record from the prodList table.

In retrieving a single record, the main flamingo page, which can be accessed at http://student.ohecampus.com/projects/teacher2/in/sem7/main.html, then called prodsingle.html which is a simple XHTML form used to input the product code, which then called prodsingle.php to retrieve and display the record. When all the DML statements are removed and then placed in the ProdList class, the resulting prodsingle.php now becomes a controller, which is accepting input from the user in the form of the product code, and then calls the model which manipulated the DML to retrieve from the database. The code for the Controller prodsingle.php is shown in listing 2 below.

```
<?php
include "db.inc";
include "prodlist.class";
?>


<html>
<head>
<title>List all Flamingo Products</title>
</head>
<body>
<b><font color="#000099"><font size=+2>Flamingo Product List </font></font></b>
    <img SRC="pinkflamingo.gif" nosave height=85 width=61>
<p>
```

```
<?php
$product = new ProdList;
$code = $_POST["prodcode"];
// OK... get the requested product
if (!$product->getRecord($code)) {
        exit("Error, Product Listing failed .... use the browsers BACK button");
}
else {
        echo("<pre>");
        echo("<p><b>Code Name Description Price</b>");
        echo("<p>");
        echo("<br>" . str_pad($product->prodCode,10) . str_pad($product->prodName,30) .
        str_pad($product->prodDesc, 60) . str_pad($product->prodPrice,10));
        echo("</pre>");
        echo("<p><a href='main.html'>Return to Main</a>");
}
?>
<p><a href="produpdate.html">Update a Product's details</a>
</body>
</html>
```

Listing 2, prodsingle.php

In comparing this to the prodsingle.php listing, there are a few noticeable differences. Firstly, the inclusions of the db.inc and prodlist.class files at the beginning of the file. Before doing anything the php script first creates a new ProdList object with the statement

$product = new ProdList;

Secondly, all the SQL (DML) statements have been removed, and replaced by function calls and references to the ProdList objects variables. The code is significantly shorter, and this controller script can now concentrate on the logic to manipulate the data at the user level, rather than the low level DML statements necessary to physically implement the business logic at the database level.

In a similar fashion, in order to update a specific record, from the main flamingo page link would be made to produpdate.html, a simple script like prodsingle.html whose purpose is to input the product code of the product to be updated. Like prodsingle.html this script does not need to change. Once the product code is entered, it then calls produpdate1.php whose purpose is to display the elements of the database record just retrieved into a form for editing. This script will utilize the ProdList class as the prodsingle.php script above did. It is listed in Listing 3 below.

```
<?php
include "db.inc";
include "prodlist.class";
?>
<?php
$product = new ProdList;
$code = $_POST["prodcode"];
// OK... get the requested product
if (!$product->getRecord($code)) {
        echo("<p> The product with code " . $code . " does not exist in the products table");
        echo("<p><a href='main.html'>Continue</a>");
        die("<br /><br />");
}
?>
<html>
<head>
<title>
Update a Single Product
</title>
</head>
<body>
<b><font color="#000099"><font size=+2>Flamingo Product Update
Form</font></font></b>    <img SRC="pinkflamingo.gif" nosave
height=85 width=61>
<br>
<br>
```

```
<form name="updateForm" action="produpdate2.php" method="POST" onSubmit="return
true">
<pre>
Product code <?php echo ($code) ?>
Name <input type="text" name="prodname" size=50 value="<?php echo($product->prodName);
?>">
Description <input type="text" name="proddesc" size=60 value="<?php echo($product-
>prodDesc); ?>">
Price <input type="text" name="prodprice" size=15 value="<?php echo($product->prodPrice);
?>">
<input type="hidden" name="prodcode" value="<?php echo($product->prodCode); ?>">
<input type=submit value="Submit Update">
</pre>
</form>
</body>
```

Listing 3, produpdate1.php

This listing just makes calls to the ProdList objects functions (namely $product->getRecord()) and variables, concentrating on the high-level logic needed. Once alterations are made by the user, the produpdate1.php controller then calls the produpdate2.php controller which performs the update. Again, the produpdate2.php controller has had the DML and low-level business logic removed and simply calls the prodlist.class model to perform the low-level functions. The code for the produpdate2.php controller is shown in Listing 4 below.

```
<?php
include "db.inc";
include "prodlist.class";
?>
<html>
<head>
```

```
<title>
Update a Simgle Product
</title>
</head>
<body>
<b><font color="#000099"><font size=+2>Flamingo Product Update
Form</font></font></b>    <img SRC="pinkflamingo.gif" nosave
height=85 width=61>
<br>
<br>
<?php
$product = new ProdList;
// store the products new details from the POST in the object
$product->setProdCode($_POST["prodcode"]);
$product->setProdName($_POST["prodname"]);
$product->setProdDesc($_POST["proddesc"]);
$product->setProdPrice($_POST["prodprice"]);

if (!$product->updateRecord()) {
        echo("<p> The product with code " . $code . " does not exist in the products table or
        could not be updated");
        echo("<p><a href='main.html'>Continue</a>");
        die("<br /><br />");
}
?>
<p> The Product details were successfully updated, <a href="main.html">Continue</a>
</body>
</html>
```

BIBLIOGRAPHY

- ACM (2006) 'THE BOSS IS WATCHING', *Communications of the ACM*, April 2006, 49(4), pp.10-10
- ACM, IEEE-CS & AIS (2005), ACM Joint Task Force on Computing Curricula, Computing Curricula 2005: The Overview Report, September 30th 2005. [Online] Available at: http://www.acm.org/education/education/curric_vols/CC2005-March06Final.pdf
- ACM. (1992) ACM Code of Ethics and Professional Conduct. Available [Online] at: http://www.acm.org/about/code-of-ethics (Retrieved August 5th, 2011).
- Adams, A. &McCrindle, R. (2008) Pandora's Box: Social and Professional Issues of the Information Age, Wiley, UK.
- Aho, A., et al. (2006) Compilers: Principles, Techniques, and Tools. 2nd Edition. Pearson – Addison Wesley. Boston. USA.
- Álvarez, M. & Guerrero, G. (n.d.) Universidad de Nariño - RESEÑA HISTORICA, [Online] Available at: http://www.udenar.edu.co/?page_id=5, Retrieved at September 10th, 2011
- Amnesty International (n.d.) FREEDOM OF EXPRESSION, Learn about Human Rights, [Online] Available at: http://www.amnesty.org/en/freedom-of-expression, Retrieved at September 1st, 2011.
- Asamblea Nacional Constituyente (1991) CONSTITUCIÓN POLÍTICA DE COLOMBIA, UpdatedOctober 2009, Presidencia de República de Colombia, [Online] Available at: http://wsp.presidencia.gov.co/Normativa/Documents/ConstitucionPoliticaColombia_20100810.pdf

- Asleson, R., &Schutta, N. (2008) Foundations of Ajax, Heidelberg: Springer-Verlag GmbH, 2008, USA
- Bagby, J. (2003) Who Owns the Data? Research Online - Penn State, January 2003, 24(1), Posted at January 2003, [Online] Available at: http://www.rps.psu.edu/0301/data.html
- Barron, D. (2000) The World of Scripting Languages, John Wiley and Sons Ltd., UK.
- Bartholomew, M. (2009) 'Cops, Robbers, and Search Engines: The Questionable Role of Criminal Law in Contributory Infringement Doctrine', *Brigham Young University Law Review*, 2009(4), pp.783-845.
- Berman, F. (2008) 'Got Data? A Guide to Data Preservation in the Information Age', *Communications of the ACM*, December 2008, 51(12), pp.50-56, DOI: 10.1145/1409360.1409376
- Bernate, F. (2007) Delitos de Falsedad en los Estados Financieros, Colección Textos de Jurisprudencia, Universidad del Rosario, Bogota – Colombia.
- Berners-Lee, T., & Connolly, D. (1993) Hypertext Markup Language (HTML): A Representation of Textual Information and MetaInformation for Retrieval and Interchange, 1st Draft, CERN, June 1993, Genève, Switzerland, [Online] Available at: http://www.w3.org/MarkUp/draft-ietf-iiir-html-01.txt, (Retrieved at October 8th, 2011)
- Bhowmick, S., Madria, S., & Ng, W. (2011) Web Data Management: A Warehouse Approach, Springer, Singapore-USA.
- Bock, G.; Ling Ho, S. (2009) 'Non-work related computing (NWRC)', *Communications of the ACM*, April 2009, 52(4), pp.124-128
- Bucknall, J. (2011) 'JavaScript Uncovered', APC, 07254415, Computers & Applied Sciences, July 2011, 31(7).

- Byung-Ik K., Chae-Tae We., & Hyun-Chul J. (2011) 'Suspicious Malicious Web Site Detection with Strength Analysis of a JavaScript Obfuscation', *International Journal of Advanced Science & Technology*, 26, pp.19-32,
- Campbell-Kelly, M. (2008) 'Historical Reflections: Will the Future of Software be Open Source', Communications of the ACM, October 2008, 51(10), pp.21-23, DOI: 10.1145/1400181.1400189.
- CERN (2008) The website of the world's first-ever web server, [Online] Available at http://info.cern.ch, (Retrieved October 8th, 2011)
- Chenoweth, T., Corral, K., &Demirkan, H. (2006) 'Seven Key Interventions for DATA WAREHOUSE SUCCESS', *Communications of the ACM*, January 2006, 49(1), pp.114-119.
- Chone, J. (2009) 'Compiled Web vs. Interpreted Web', *Bits & Buzz*, [Online] Available at: http://www.bitsandbuzz.com/article/compiled-web-vs-interpreted-web/, (Retrieved at October 22th, 2011)
- Computer Desktop Encyclopedia (2011) 'Deep Web', *Computer Desktop Encyclopedia*, July 2011, pp.1-1.
- Congreso de la República (2008) LEY ESTATUTARIA 1266 DE 2008, [Online] Available at: http://www.secretariasenado.gov.co/senado/basedoc/ley/2008/ley_1266_2008.html
- Congreso de la República de Colombia (1990) LEY 43 de 1990, [Online] Available at: www.oas.org/juridico/spanish/mesicic3_col_ley43.pdf, Accessed: August 23, 2011
- Corsa, L. (2008) 'Notes from the UNDERGROUND', Hispanic, September 2008, 21(9), pp.32-33.

- Costa, J. et al (2011) 'ONE: A Predictable and Scalable DW Model', Data Warehousing and Knowledge Discovery - 13th International Conference, DaWaK 2011, Toulouse, France, August 29-September 2,2011. Proceedings.
- CPSR (2005) 'RFID: What Congress Needs to Know?' Computer Professionals for Social Responsibility, [Online] Available at: http://cpsr.org/issues/privacy/inet_caucus_rfid/, (Accessed September 1st, 2011)
- Crystal, D. (2008) Txtng The Gr8 Db8. Oxford University Press. New York. USA.
- Damasio, A. (1994) Descartes Error: Emotion, Reason, and the Human Brain, Gosset/Putnam Press, New York, USA
- Deitel P., &Deitel, H. (2008) Internet & World Wide Web – How to Program, 4th Edition, Pearson Education Inc., New Jersey, USA.
- Estell, J. & Christensen, K. (2011) 'The Need for a New Graduation Rite of Passage' *Communications of the ACM*. February 2011. 54(2). pp.113-115. DOI: 10.1145/1897816.1897846
- European Commission (n.d.) Data Protection in the European Union, [Online] Available at: http://ec.europa.eu/justice/policies/privacy/docs/guide/guide-ukingdom_en.pdf
- Favaro, J. &Pfleeger, S. (2011) 'Guest Editors' Introduction: Software as a Business', *IEEE Software*, July 2011, 28(4), pp.22-25, DOI: 10.1109/MS.2011.77
- Frank, D. (2008) 'How to Fix a Bad Outsourcing Deal', *The CIO Executive Council*, October 2008, 22(2), pp.66-68
- FSF (2011) 'Free GNU/Linux distributions', GNU Operating System – Free Software Foundation, [Online] Available at: http://www.gnu.org/distros/free-distros.html, (Retrieved: November 4th, 2011).

- FSF (2011) 'The Free Software Definition', GNU Operating System – Free Software Foundation, [Online] Available at: http://www.gnu.org/philosophy/free-sw.html, (Retrieved: November 3th, 2011).
- Fuecks, H., (2004), The PHP Anthology Volume 1, Chapter 2 - Object Oriented PHP, Sitepoint Jan 2004, http://www.sitepoint.com/article/object-oriented-php. Accessed Oct 2006
- Gamma, E. et al (1995) Design Patterns: Elements of Reusable Object-Oriented Software, Pearson Education, USA
- Gardner, H. (2006) Multiple Intelligences: New Horizons in Theory and Practice. Basic Books. USA
- Garvey, P. (2001), Implementing a Risk Management Process for a Large Scale
- Gingrande, A. (2006) 'From E-forms to Xforms', *AIIM E-DOC*, March/April 2006, 20(2), pp.14-16
- Goldberg, L. (1996) 'A Note on Current Assets'. *Abacus*, 1(1), pp31-45.
- Goodman, D. et al (2010) JavaScript Bible, 7th Edition, Indianapolis (IN), USA.
- Goodman, D. et al (2010) JavaScript Bible, 7th Edition, Wiley Publishing, Indianapolis (IN), USA
- Green, C.D. (2005) 'Was Babbage's Analytical Engine Intended to Be a Mechanical Model of the Mind?' *History of Psychology*, 8(1), February 2005, pp.35-45.
- Hillyard, D., &Gauen, M. (2007) 'Issues Around the Protection or Revelation of Personal Information', *Knowledge, Technology & Policy*, summer 2007, 20(2), pp.121-124, DOI: 10.1007/s12130-007-9013-9.

- IEEE Computer Society (2010) IEEE Standard for Information Technology – System and Software Life Cycle Processes – Reuse Processes, IEEE Standard 1517™ - 2010, New York, USA.
- Insuasti, J. et al (2009) PROPUESTA DE PLAN DE DESARROLLO DEL SISTEMA DE INFORMACIÓN INTEGRADO DE LA UNIVERSIDAD DE NARIÑO, SystemsDepartment, Universidad de Nariño, Pasto – Colombia
- IT Now (2010) 'Future of the IT Profession', *IT Now*, January 2010, 52(1), pp.25-25, DOI: 10.1093/itnow/bwq128
- Jewett, T. (n.d.) 'Design pattern: many-to-many (order entry)', DB design, 3rd edition, [Online] Available at: http://www.tomjewett.com/dbdesign/dbdesign.php? page=manymany.php, (Retrieved: November 12th, 20)
- Manderieux, L. (2011) Practical Guide to the Creation and Management of Technology Transfer Offices at Universities and Research Centers in Latin America, WIPO, June 2011, Publication 1026S, Genève, Switzerland.
- Marston, T., (2003), Using PHP Objects to access your Database Tables (Part 1), http://www.tonymarston.net/php-mysql/databaseobjects.html, accesses October 2006
- Marston, T., (2005), RADICORE - A Development Infrastructure for PHP, http://www.tonymarston.net/php-mysql/infrastructure.html, Accessed October 2006
- Marston, T., (2006), The Model-View-Controller (MVC) Design Pattern for PHP, http://www.tonymarston.net/php-mysql/model-view-controller.html, Accessed October 2006
- Marx, K. (1845) Theses on Feuerbach. 1st Publication 1924. Institute of Marxism-Leninism. Moscow.
- McDonald, M. et al (2010) Pro ASP.NET 4 in C# 2010, Apress, USA

- McGregor, D. (1960). The Human Side of Enterprise, M.I.T. Press. USA
- Melnic, A (2010) 'RISK RESPONSE STRATEGIES IN PROJECT MANAGEMENT', *Metalurgia International*, August 2010 Supplement, pp.74-78,
- Microsoft (2011) 'Microsoft Private Cloud', Microsoft Server and Cloud Platform, [Online] Available at: http://www.microsoft.com/en-us/server-cloud/private-cloud/default.aspx, (Retrieved November 26th, 2011)
- Mills, M. (2011) 'Hearing Aids and the History of Electronics Miniaturization', *Annals of the History of Computing IEEE*, February 2011, 33(2), pp.24-25, DOI: 10.1109/MAHC.2011.43
- Mykolaitiene, V. et al. (2010) 'Peculiarities of Tangible Fixed Assets Accounting', INZINERINE EKONOMIKA-ENGINEERING ECONOMICS, 21(2), pp.142-p150
- Nair, A. (2007) 'Internet Content Regulation: Is a Global Community Standard a Fallacy or the Only Way Out?', *International Review of Law, Computers & Technology*, March 2007, 21(1), pp.15-25, DOI: 10.1080/13600860701281655
- Nettleton, E., &Obhi, H. (2004) 'Legal protection for databases in Europe: The vexed question of whether US businesses can benefit', *Journal of Database Marketing & Customer Strategy Management*, April 2004, 11(3), pp.268-273.
- NIST (2002) Risk Management Guide for Information Technology Systems, Special Publication 800-30, U.S. Department of Commerce, Information Technology Laboratory - National Institute of Standards and Technology, Gaithersburg, MD, USA. [Online] Available at: http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf
- Nord, G., McCubbins, T., & Nord, J. (2006) 'E-Monitoring in the Workplace: PRIVACY, LEGISLATION, AND SURVEILLANCE SOFTWARE', *Communications of the ACM*, August 2006, 49(8), pp.73-77

- O'Connor, R. (2011) 'Risk issues of software and IT outsourcing and its impact on business continuity', *London: Henry Stewart Talks*, streaming video file (33 min.): digital, mono. SWF file, Dublin City University, Ireland.
- Oficina de Planeación y Desarrollo (2010) Mapa de Procesos, Sistema de Gestión de Calidad Universidad de Nariño, [Online] Available at: http://akane.udenar.edu.co/siweb/calidadudenar/wp-content/uploads/2010/06/Mapa.jpg, Retrieved: September 10th, 2011
- Oficina de Planeación y Desarrollo (2010) PRESUPUESTO 2011, Universidad de Nariño, Pasto, Colombia, [Online] Available at: http://www.udenar.edu.co/planeacion/PresupuestoAprobadoUDENAR_Vigencia2011.pdf, Retrieved September 10th, 2011
- Oswald, I., &Reller, A. (2011) 'E-Waste: A Story of Trashing, Trading, and Valuable Resources', *GAIA: Ecological Perspectives for Science & Society*, March 2011, 20(1), pp.41-47
- Paraschiv, N., Melinte, T., &Pricop, E. (2009) 'Considerations about RFID Systems Vulnerabilities', Petroleum - Gas University of Ploiesti Bulletin, Technical Series, 61(3), pp.87-92,
- Parnas, D. (2010) 'Inside Risks of Undisciplined Development', *Communications of the ACM*, October 2010, 53(10), pp.25-27, DOI: 10.1145/1831407.1831419
- Pattern (2005) , Model_view_controller, http://www.phpwact.org/pattern/model_view_controller, Accessed October 2006
- PHP (2011) 'PHP Cookies', PHP Manual, [Online] Available at: http://www.php.net/manual/en/features.cookies.php, (Retrieved: November 9th, 2011)
- PHP Manual, (2011) Chapter 19. Classes and Objects, http://au2.php.net/manual/en/language.oop5.php

- PHPDeveloper (2006), Tutorial: An Introduction to OOP in PHP, http://www.phpdeveloper.org/news/5719, Accessed Oct 2006
- Remolina-Angarita, N. (2010) 'Does Colombia have an adequate level of Personal Data Protection in light of the European Standards?' *Revista Colombiana de Derecho Internacional*, July-December 2010, 17, pp.489-523
- Roger, A., et. al. (2010) 'Business Process Requirement Engineering', *IJCSE - International Journal on Computer Science and Engineering*, 02(09), pp.2890-2899
- Rosen, M. et al (2008) Applied SOA: Service-Oriented Architectures and Design Strategies, Wiley Publishing Inc., Indianapolis (IN), USA
- Samuelson, P. (2010) 'Legally Speaking Why Do Software Startups Patent (or Not)?', *Communications of the ACM*, November 2010, 53(11), pp.30-32, DOI: 10.1145/1839676.1839687
- Sánchez, Y. (2011) Generation Y – My Profile, [Online] Available at: http://www.desdecuba.com/generationy/?page_id=108, Retrieved September 2nd, 2011
- Savage, N. (2010) 'Building Babbage's Analytical Engine'. *Communications of the ACM*, December 2010, 53(12), pp.18-18
- Schartum, D. (2010) 'Designing and Formulating Data Protection Laws', *International Journal of Law & Information Technology*, Oxford University Press, 18(1), pp.1-27, DOI: 10.1093/ijlit/ean013
- Schryen, G. (2011) 'Is Open Source Security a Myth?',*Communications of the ACM*, May 2011, 54(5), pp.130-140, DOI: 10.1145/1941487.1941516.

- Singh, R., & Singh, K. (2010) 'A Descriptive Classification of Causes of Data Quality Problems in Data Warehousing', *International Journal of Computer Science Issues (IJCSI)*, May 2010, 7(3), pp.41-50.
- Sixth Congress of the Communist Party of Cuba (2011) Resolution on the Guidelines of the Economic and Social Policy of the Party and the Revolution: "Year 53rd of the Revolution", [Online] Available at: http://www.cuba.cu/gobierno/documentos/2011/ing/l160711i.html, Retrieved September 2nd, 2011
- Solum, L. (2008) Models of Internet Governance, Illinois Public Law Research Paper No. 07-25, Social Science Research Network, [Online] Available at: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1136825, Retrieved September 2nd, 2011
- Stallman, R. (2011) 'Common Misunderstandings of "Free Software" and "Open Source"', Why Open Source misses the point of Free Software - Free Software Foundation, [Online] Available at: http://www.gnu.org/philosophy/open-source-misses-the-point.html, (Retrieved: November 4th, 2011).
- Stein, J., & Harrell, E. (2011) 'Your Data, Yourself', *Time*, March 2011, 177(11), pp.40-46.
- Taft, D.K. (2006) 'AJAX, SOA TO MERGE', *eWeek*, October 2006, 23(39), p18-18
- Taylor, F. W. (1911) The Principles of Scientific Management, Harper and Brothers Publishers, New York and London, USA and UK.
- Taylor, R. (2007) 'Ethical dimensions of electronic monitoring of employees: a cross-cultural perspective', *ASBBS E-Journal*, 3(1), pp.140-145, [Online] Available at: http://www.asbbs.org/proceedings/Taylor%20ASBBS%20EJ2007.pdf
- The European Parliament and the Council of the European Union (1996) DIRECTIVE 96/9/EC, Official Journal L 077, 27/03/1996 P. 0020 – 0028, Accessed at August 20th 2011,

[Online] Available at: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31996L0009:EN:HTML

- UN General Assembly (1948) Universal Declaration of Human Rights, United Nations, New York, USA, [Online] Available at: http://www.un.org/en/documents/udhr/ (Accessed August 28th, 2011)
- Unidad de Informática y Telecomunicaciones (n.d.) VISIÓN DE LA UNIVERSIDAD DE NARIÑO, [Online] Available at: http://www.udenar.edu.co/?page_id=9, Retrieved: September 10th, 2011.
- United Nations (1949) The Universal Declaration of Human Rights, [Online] Available at: http://www.un.org/en/documents/udhr/index.shtml, Retrieved at September 1st, 2011.
- USA Today (2010) Electromagnetic pulse impact far and wide, Posted October 26 2010, Accessed August 20th, 2011, [Online] Available at: http://www.usatoday.com/tech/science/2010-10-26-emp_N.htm
- W3C (2004) Web Service Glossary, [Online] Available at: http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/, (Retrieved: November 26th, 2011)
- W3C (2004) World Wide Web Consortium 10th Anniversary, W3C, [Online] Available at: http://www.w3.org/2004/Talks/Styles/w3c10/images/timeline.pdf, (Retrieved October 8th, 2011)
- W3C (2005) Document Object Model - DOM, World Wide Web Consortium, [Online] Available at: http://www.w3.org/DOM/ (Retrieved: October 14th, 2011)
- W3C (2010) HTML & CSS, Web Design and Applications, [Online] Available at: http://www.w3.org/standards/webdesign/htmlcss, (Retrieved October 8th, 2011)

- W3C (2011) HTML5 - W3C Working Draft 25 May 2011, [Online] Available at: http://www.w3.org/TR/html5/, (Retrieved: October 8th, 2011).
- W3C (2011) XForms implementations, World Wide Web Recommendation, [Online] Available at: http://www.w3.org/MarkUp/Forms/wiki/XForms_Implementations, (Retrieved: October 15th, 2011)
- W3C (n.d.) Forms in HTML Documents, World Wide Web Recommendation, [Online] Available at: http://www.w3.org/TR/html401/interact/forms.html, (Retrieved: October 15th, 2011)
- Wachs, J. et al. (2011) 'Vision-Based Hand-Gesture Applications', *Communications of the ACM*, February 2011, 54(2), pp60-71, DOI: 10.1145/1897816.1897838
- Wadlow, T., &Gorelik, V. (2009) 'Security in the Browser', *Communications of the ACM*, May 2009, 52(5), pp.40-45, DOI:10.1145/1506409.1506422.
- Waller, V., & Johnston, R. (2009) 'Making Ubiquitous Computing Available', *Communications of the ACM*, October 2009, 52(10), pp.127-130, DOI: 10.1145/1562764.1562796
- White, P. (2011) 'Is our future in the "cloud"?' *IBE: International Broadcast Engineer*, Jul/Aug 2011, pp.11-11
- Wiederhold, G. (2011) 'Follow the Intellectual Property', *Communications of the ACM*, January 2011, 54(1), pp.66-74, DOI: 10.1145/1866739.1866756
- Wolter, R. (2001) 'An overview of the value of XML Web services for developers, with introductions to SOAP, WSDL, and UDDI', XML Web Services Basics – MSDN (Microsoft Developer Network), Microsoft.

- Wong, W. (2007) 'What's New in Programming Languages?',*Electronic Design*, 55(3), pp.57-57.
- Wooldridge, M. (2000) Reasoning about Rational Agents, 1st Edition, MIT Press, USA.
- Wright, D., De Hert, P., &Gutwirth, S. (2011) 'Are the OECD Guidelines at 30 Showing Their Age?' *Communications of the ACM*, February 2011, 54(2), pp.119-127, DOI: 10.1145/1897816.1897848
- Zend (2004), Tutorial: Writing Classes in PHP, http://www.php-editors.com/articles/simple_php_classes.php, Accessed September 2006
- Zheng, Y. (2009) 'Different spaces for e-development: What can we learn from the capability approach?' *Information Technology for Development*, 15(2), pp.66-82, DOI: 10.1002/itdj.20115
- Zhu. H., Madnick, S., & Siegel, M. (2008) 'An Economic Analysis of Policies for the Protection and Reuse of No copyrightable Database Contents'. *Journal of Management InformationSystems*, Summer 2008, 25(1), pp.199-232. DOI 10.2753/MIS0742-1222250108
- Zipkin, P. (2006) 'The Best Things in Life Were Free: On the Technology of Transactions', Manufacturing & Service Operations Management, Fall 2006, 8(4), pp.321-329, DOI: 10.1287/msom.1060.0113