

# Computer Science's Digest

## Volume 1

Jesús Insuasti. Jairo Guerrero García. Juan Carlos Castillo



© 2015 Editorial Universitaria - University of Nariño

Authors: Jesús Insuasti. Jairo Guerrero García. Juan Carlos Castillo

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

Computer Science's Digest. Volume 1

ISBN (volume): 978-958-8958-17-0

Edition number: 1

Cover design: Galeras.NET Research Group

Proof reading: Katrin Riddle

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

Printed and made in Pasto, Colombia.

Total or partial reproduction, by any means or any purpose, without the written permission of the authors is prohibited.

Insuasti, Jesús

Computer Science's Digest / Jesús Insuasti, Jairo Guerrero García y Juan Carlos Castillo/ San Juan de Pasto: 1ª. Ed.- Editorial Universitaria. - Universidad de Nariño, Grupo de investigación Galeras.NET.- Departamento de Sistemas, 2015.

216 pág.  
v.1

ISBN: 978-958-8958-17-0

1. Informática – Avances Tecnológicos. 2. Enseñanza de la Informática. 3. Programación de computadores 4. Lenguaje de Programación.

005 I599 - SCDD- 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.

### **Jairo Guerrero, MSc.**



Jairo Guerrero was the first full-time professor in the Systems Department; in fact, he was its founder. He is a Systems Engineer from the Piloto University in Bogota (Colombia). He has a Master's Degree in Models on Problem-based Teaching (Colombia). Jairo Guerrero has served as Chair of the Systems Department, Dean of the Faculty of Engineering and the Director of the Planning Office in the University of Nariño.

### **Juan Carlos Castillo, MSc.**



Juan Carlos Castillo is a Systems Engineer from the University of Nariño, a Specialist in Networks and Telematics Services and has a Master's of Science in Electronics and Telecommunications Engineering from the University of Cauca. He is currently a full-time researcher and professor in the area of Telematics Systems at the University of Nariño.

---

## **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 the Introduction to Computer Science, Computer Communications, Networking and Web Applications.

---

## **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. INTRODUCTION TO COMPUTER SCIENCE .....	9
1.1. REFLECTIONS .....	9
1.1.1 A Near Future (Computer Education & Software Development - 2021).....	9
1.1.2. An Essay about Data Storage Technologies.....	11
1.1.3. Some Experiences with Data Storage.....	13
1.1.4. Concerning some OS environments .....	14
1.1.5. Network Environments and the Web .....	16
1.1.6. Analysis of Algorithms: A case study of numbers divisible by 3 .....	19
1.1.7. As the saying goes: “Every man to his own trade” .....	21
1.1.8. Computing History: A Personal Perspective on Software Development and Storage Technologies .....	23
1.1.9. How will the existence of the Internet and database technology affect the existence of libraries themselves? .....	27
1.1.10. Machine Independence.....	28
1.1.11. Types of Media: A Look at the BBC and Open Source Technology .....	30
1.1.12. A Note on Processes in an OS.....	32
1.1.13. New Trends in Software Intelligent Agents .....	34
1.1.14. Pros and Cons of Interpreted Code.....	35
1.1.15. Solvable and Unsolvable Problems .....	37
1.1.16. Considerations on Distributed Databases .....	39
1.1.17. Some Impacts of Artificial Intelligence .....	42
1.1.18. Threats against Network Security .....	43
1.1.19. Waterfall Development vs. XP Approach .....	45
1.2.1. Workshop 1 .....	48
1.2.2. Workshop 2 .....	52
1.2.3. Workshop 3 .....	56
1.2.4. Workshop 4 .....	60
1.2.5. Workshop 5 .....	64
2. COMPUTER COMMUNICATIONS AND NETWORKING.....	68
2.1. REFLECTIONS .....	68
2.1.1. A Bad Day for ARPANET .....	68

---

2.1.2.	Audio Streaming.....	70
2.1.3.	American Registry for Internet Numbers: Whois Service.....	72
2.1.4.	Are 3-DES and RSA secure algorithms?.....	74
2.1.5.	ATM and Gigabit Ethernet.....	75
2.1.6.	Connections within a Web Browser .....	78
2.1.7.	DNS: Untangling the Web for End Users .....	80
2.1.8.	DNS Hacking .....	82
2.1.9.	Exploring Network Management .....	83
2.1.10.	Functions of Network Adapters and Nodes.....	85
2.1.11	NAT and its impact on IPSec .....	88
2.1.12.	P2P Networking .....	90
2.1.13.	SIP basics .....	92
2.1.14.	TCP Hijacking.....	94
2.1.15.	Wi-Fi Networking: A case at home.....	96
2.2.	WORKSHOPS .....	98
2.2.1.	Hand-In Assignment 1.....	98
2.2.2.	Hand-In Assignment 2.....	102
2.2.3.	Hand-In Assignment 3.....	108
2.2.4.	Hand-In Assignment 4.....	113
2.2.5.	Hand-In Assignment 5.....	117
2.2.6.	Hand-In Assignment 6.....	124
2.2.7.	Hand-In Assignment 7.....	128
2.2.8.	Hand-In Assignment 8.....	132
3.	WEB APPLICATIONS .....	135
3.1.	REFLECTIONS .....	135
3.1.1.	Semantic Web.....	135
3.1.2.	DTD and XML Schema.....	140
3.1.3.	Formatting data in XML technologies .....	141
3.1.4.	Giving hierarchical structure to data with XML .....	143
3.1.5.	Securing SOAP.....	144
3.1.6.	SOAP in a small public university .....	146
3.1.7.	Some considerations on native XML databases .....	148
3.2.	THE MAKING OF AN XML SCHEMA .....	150
3.2.1.	UDDI's positioning.....	153

---

3.2.2.	WSDL in the SOAP's World .....	155
3.2.3.	XML table-based mapping .....	157
3.2.4.	XML technologies and open standards .....	159
3.2.5.	XML Web Services and Cloud Computing.....	160
3.2.6.	XML Web Services within a public university .....	162
3.3.	WORKSHOPS .....	164
3.3.1.	Hand-In Assignment 1.....	164
3.3.2.	Hand-In Assignment 2.....	171
3.3.3.	Hand-In Assignment 3.....	175
3.3.4.	Hand-In Assignment 4.....	181
3.3.5.	Hand-In Assignment 5.....	190
3.3.6.	Hand-In Assignment 6.....	195

---



# **1. INTRODUCTION TO COMPUTER SCIENCE**

## **1.1. REFLECTIONS**

### **1.1.1 A Near Future (Computer Education & Software Development - 2021)**

Since computer science made its appearance in the field of formal study in the 20th century, technological developments have advanced rapidly. Over the past 50 years the world has made greater progress in communications than it had during the previous 500.

Computer Science impacts every aspect of human life. As it is a science, the theoretical foundation is best and most effectively learnt in educational environments.

However, even though technological advances in this discipline are constant, it seems the teaching of computer science will continue with its traditional structure, merely using new technology. Traditional structure meaning the prevalence of computer curricula, syllabi, labs, etc. Nonetheless, teaching itself (didactics) should change over the course of time.

Nowadays, our students are taught to program computers, and there are two things which are fairly certain looking at the next 10 years:

First of all, our job will still exist in 2021. Education is a natural feature of human beings, so it will exist as long as humankind does. Furthermore, teaching-learning processes will continue down the same path, but the means and technology involved will change. That is to say, at the very least we will continue to see people learning from people (or machines, why not?).

In the second place, software development will almost certainly see some drastic changes. Professionals in computer science of the authors' generation have witnessed enormous advances in the field of computer programming in the last 30 years. All programming paradigms have their implementations in programming languages, and these languages have been evolving towards more suitable programming platforms.

In the beginning, programming had to be done line by line, instruction by instruction. Now, frameworks and object-oriented components are used which allow us to build software based on solid foundations. This makes writing code much easier than in the past. This situation is touched on in an interesting paper entitled, "Learning and Teaching Programming: A Review and Discussion" (Robins, Rountree J., and Rountree N., 2003). They outline an educational context which oscillates between a procedural, and object-oriented approach to programming, where one of the benefits of using the latter approach is that it shows a way to simplify code.

Therefore, a near future could be envisaged in which software development will be a trivial task in terms of programming-related tasks. In contrast, software design will be the main labor. It seems evident that the programmers of the future will build their software through design patterns, using natural language and holographic interfaces to enable motion capture and gesture recognition. Therefore, professionals in computer education will need new skills, in order to teach software development to future generations.

### **1.1.2. An Essay about Data Storage Technologies**

Since the beginning of Computing as a science, data storage has been one of the most important areas in the computing industry. The ability to manage data appropriately is as essential as the computation capability. Therefore, data storage technologies are as old as computing itself.

Over time, data storage technologies have evolved in parallel with developments in the field of physics and electronics. Due to this accelerated process, some technologies are obsolete these days, like punched cards.

Punched cards were used before the computer era, in the textile industry during the 18<sup>th</sup> century. At that time, punched cards served as design patterns for textile machines. It was not until Hollerith in the following century, that punched cards were used to store data for computing purposes (ACM, 1968).

This kind of technology was quite simple. Punched cards used holes to describe something, such as a code instruction, an address, or data; however, the machines which read these cards were much more complex. The biggest problem in dealing with punched cards was the preservation of data, due to their physical size (an 80-column punched card had 187.325 by 82.55 mm). This caused several inconveniences, such as the ones detailed below:

1. When storing data in punched cards, their order was very important. Having the correct sequence of the punched cards was essential. So, if an accident happened and the sequence was changed, information was corrupted.

2. If there was a large quantity of information, the information was not easily transportable.

Moving the information that was stored in punched cards was laborious, because people had to move all the boxes in which it was contained.

In fact, in the 1970's, the administrative section of the electric power plant in our hometown did not have the machines to process data; engineers only had punching devices to store information by making holes in the cards. When the plant needed to process the data, such as the payroll, workers had to pack their punched cards into boxes, then get hired transport to take everything to the nearest city where data processing was possible.

These are the main reasons why this kind of technology worked for so short a time. People need comfortable and convenient technology.

At the other end of the spectrum are the cutting-edge products which are at the forefront of the technology wave. New technological proposals on data storage are being developed by improving the data storage features. A novel and interesting proposal is holographic data storage.

Mr. Graham-Rowe (2007: 199) stated that Holographic Data Storage is a kind of optical technology that allows the storage of multiple images in the same place (a disk) by changing the light beam's angles; therefore, this approach could store from 200 GB up to 1Tb. This innovative technology has given rise to the concept of HVD (Holographic Versatile Disc) and right behind this product are several prestigious companies such as Sony, Apple, Mitsubishi, Fuji, Konica and Optware, among others. This group of Hi-Tech enterprises make up the Holography System Development Forum.

Production of this technology is already underway. Many high, international expectations surround this data storage system, especially from standard-making organization ECMA. Labriola and Mangis (2007) state, “The Ecma International standards body has begun defining 200 GB HVD-R and 100 GB HVD-ROM specifications that could eventually transform this data-archiving technology into a DVD-like content-delivery platform”.

It seems that HVD technology will be the next step in data storage, due to its proven preliminary tests and its price. Perhaps, in a very near future, it will be possible to access these huge storage capacities in the same way that we are currently using DVDs and DVD recorders. Furthermore, documentation shows that the cost of production of this technology would not be as expensive as magnetic-based data storage. (Labriola, 2006: 82)

### **1.1.3. Some Experiences with Data Storage**

Take as an example, the case of a personal laptop with a 300Mb hard disk distributed in 2 partitions: the first one with 200 GB and the second one with 100 GB. Right now, if there is only 27 GB in free space in the first partition; that means there is 173 GB of used space. If data from the laptop needs to be backed up, the choice is between CDs and DVDs, and obviously DVD technology is preferable due to its capacity i.e. it takes less DVDs than CDs to save all the information.

However, this is a hypothetical situation. In real life, Tape-based Technology would be the best choice because it has more capacity than either CDs or DVDs, and it was designed for backup purposes. Finally, it is simpler and cheaper than optical solutions, when there is a lot of information to be backed up.

However, there is another resource available: cloud-based backup. DELL laptops, offer an online backup and recovery service for the 1<sup>st</sup> year. Having tested their service, the authors came to the conclusion that this method is easy to use, and it is also very efficient. However, the user needs a very fast Internet connection to create a full backup and a full recovery in the cloud. In our case, creating a backup took about 1 hour on our Internet connection of 17.77Mb and 2.23 Mb for download and upload respectively according to SpeedTest.Net site (n.d.).

This latter kind of data storage does not make distinctions between images, text, music and videos. All data, in the form of raw information, is represented as bit streams making it very useful nowadays.

#### **1.1.4. Concerning some OS environments**

In an academic context, several platforms need to be used in which concepts concerning the teaching of software development can be applied. Those platforms are able to install Internet solutions through server infrastructure. We used to work with Oracle Solaris 11 Express, Microsoft Windows Server 2008 R2, and some Linux distributions.

However, now it is preferable to work with a type of technology with which we have had more experience and have been certified, Microsoft Technology. For our application developments, Microsoft Windows Server R2 Enterprise with Hyper-V (Service Pack 1) is used, mounted on a small server with the features below:

- OS Name: Microsoft Windows Server 2008 R2 Enterprise
- Version: 6.1.7601 Service Pack 1 Build 7601



- OS Manufacturer: Microsoft Corporation
- System Manufacturer: Hewlett-Packard
- System Model: HP Compaq dc7900 Small Form Factor
- System Type:x64-based Architecture
- Processor: 2 processors Intel(R) Core(TM) 2 Quad CPU Q8400 @ 2.66 GHz, 2667 MHz.
- RAM: 16 GB.
- Hard Disk: 2 disks 1 Tb each one.

Using this machine, our web applications and services were generated. All that is needed is a web server (Microsoft Internet Information Services 7.5) and a development/deployment platform (Microsoft .NET Framework 4 SP1). Our applications and services were also built on our laptops, which have Microsoft Windows 7 SP1 Enterprise (64-bit).

This particular server-based OS was chosen for the following 3 reasons:

1. Hyper-V is a complete tool for virtualizing servers, and its applications are efficient performers. That enables the use several OS platforms, running at the same time (Microsoft, n.d.).
2. This OS allows the user to build on their own private cloud. This means students can be taught how to build cloud-based applications (Microsoft, n.d.).
3. Integrated experience with Windows 7 (Microsoft, n.d.). The client machines belonging to students can be linked in as part of the private network, allowing us to build applications and share resources.

Furthermore, Microsoft Windows Server 2008 R2 SP1 with Hyper-V was evaluated by Network World (2010) and was shown to have significant advances in speed, security and management. This OS has a redesigned Terminal Services and it allows for an easy integration with Microsoft Windows 7 clients.

By setting up the network environments, administrators can customize the configuration profiles easily. This OS offers more advantages than its previous versions. Unfortunately, Hyper-V technology only works under a 64-bit architecture with specific features in hardware. As such, some older machines cannot reap the benefits of virtualization through this OS.

#### **1.1.5. Network Environments and the Web**

Some users connected to the global network do not distinguish between the Internet and the World Wide Web (WWW). In fact, they talk about the same idea using those terms indistinctly. The main reason could be that people are using only www-based applications and services, and as such, experience an incomplete view of global network technology.

The Internet and the WWW are two different concepts. The former is the global infrastructure which supports several communication services, it is not considered a network, but a huge collection of networks connected to each other. In contrast, the latter is just one service supported by the Internet.

According to the above definitions, the Internet represents the highway, a set of technologies related to global communication networking based on computing. The Internet serves

as a deployment platform for several services such as WWW, FTP, Telnet, IRC, RSS, e-mail and gopher, among others. WWW is just one Internet service.

In the complex world of the Web (or WWW service), two aspects are essential in understanding its functionality: servers and clients. A server (and specifically a Web Server) in the WWW environment is an entity that publishes web content, and sets up web applications and services. A server is just one point in the huge cloud of linked points which make up the Web (in fact, this term was first conceived from the biological concept of spider webs). A client is an endpoint which consumes a published web application or service, a machine (or software according to the context) with the ability to gain access to the Internet, and which is able to use WWW resources. Therefore, a client could be a desktop, a laptop, a tablet, another server, a PDA, a phone, a TV, a game console, etc.

Communications between servers and clients are regulated by protocols. The client/server base protocol for the WWW is HTTP (Hypertext Transfer Protocol). However, over HTTP other run protocols, such as SOAP (Simple Object Access Protocol).

In spite of the fact that SOAP follows the distributed computing paradigm, this protocol uses the client-server model which is implemented over HTTP in order to enable the transfer of complex XML-based data. Jorgensen (2002: 137) states that, “SOAP somewhat contrary to its name, is fundamentally just a protocol that lets two systems-client and a server-exchange data”. SOAP allows a high transportability of information, this means that SOAP is able to deploy XML Web Services by following the Service-Oriented Architecture approach, so that it can be used by practically all mobile and non-mobile devices (Yilmaz, 2006).

SOAP is used almost every day in our classes, and it is a pleasure to develop and set out XML Web Services using SOAP. The reason for this is that this feature transforms our computing solutions enabling scalability, security, portability and massive usage. Therefore, under the classification of networks by their functional relationship, this type of development can create computing solutions for both client-server networks and peer-to-peer networks.

In discussing the implementation of SOAP, the following must be taken into account: when a computing solution is built based on SOAP, it has to be developed for two locations: the server and client. (Yilmaz, 2006: 26). On the server, the developed XML Web Service must be published (we use Microsoft .NET technology). Then, on the client, desktop-based application that consumes the XML Web Service is installed. If data needs to be processed, such as new financial calculations in a credit account, a database is designed, and the access logic is written in it (through stored procedures, functions and triggers). Then, the XML Web Service will deal with connectivity, authorization and authentication tasks and the business logic (account transactions such as deposits and withdrawals), so all the information can be shown in XML envelopes. Finally, the client application just serves as an interface between customers and the corporation. The client application grants to customers a friendly user interface which allows them to handle data easily. This is an example of the kind of client-server activities that are developed in a network environment.

In addition, a web application can also incorporate client-server scripts which are executed in separate locations. For instance, web applications were built using Microsoft ASP.NET technology, which execute processes such as in server user authentications. For this to happen, code had to be written in Microsoft C# in order to validate the login data which comes before a database (that is the server code, and this code is private: clients cannot get access to it). On the

other hand, the same web application uses AJAX technology to enrich its user interface. Visual effects are made with JavaScript and that code will run in client, so the code is open to everyone.

#### **1.1.6. Analysis of Algorithms: A case study of numbers divisible by 3**

In order to build an algorithm to get a string of numbers, and identify all of the substrings that form numbers that are divisible by 3, a pseudo-code has been used showing a logic sequence.

In the pseudo-code below, some of the concepts have been underlined. These indicate the use of pre-defined functions, methods or operations which are implemented in almost all programming languages. For example some of the functions used here were:

Length: This refers to the number of characters in a string.

The Substring: This refers to a substring starting at an index position (an integer number starting with 0), and counts the number of characters which follow.

The Integer Conversion: This is an operation which converts from string to integer.

The Module: This is the remainder of an integer division.

Show: This is an operation which shows information to the user.

```

procedure AllDivisibleBy3(input_number)
  i <- 0;
  while (i < The Length of input_number) do
  (
    j <- The Length of input_number;
    while (j > i) do
    (
      substring <- The Substring of input_number started by i using j - i characters;
      if (The Module between The Integer Conversion of substring and 3 = 0)
        then (Show substring " is divisible by 3" );
      j <- j - 1;
    ) end while
    i <- i + 1;
  ) end while

```

Figure 1.1 Pseudo-code about divisibility by 3

This algorithm has two nested loops. The first one goes from the beginning of the input string (input\_number) till the end. The second one, which is within the first one, takes in the whole string (input\_number) and finishes at the first loop's control variable.

As such, the sequence of the algorithm's behavior can be explained with the tracking below:

**Input\_number:** 375

Iteration	substring	Comparison	Result
1 (i = 0; j = 3)	375	The module between 375 and 3 is 0? True	375 is divisible by 3
2 (i = 0; j = 2)	37	The module between 37 and 3 is 0? False	
3 (i = 0; j = 1)	3	The module between 3 and 3 is 0? True	3 is divisible by 3
4 (i = 1; j = 2)	75	The module between 75 and 3 is 0? True	75 is divisible by 3
5 (i = 1; j = 1)	7	The module between 7 and 3 is 0? False	
6 (i = 2; j = 1)	5	The module between 7 and 3 is 0? False	

Table 1.1 Sample of comparisons and results



As a comparison, colleague Tony Kehoe (2011) published another similar algorithm which is interesting for the following reasons:

Firstly, his procedure can work with any kind of divisibility application. That is to say, Kehoe put the divisibility criteria as a parameter. In contrast, the previous solution only works for divisibility by 3.

Secondly, he also designed two loops, but using contrary logic. In other words, his loops have a different logic: the former has positive logic with initial comparison (while ... do ...), and the latter has negative logic with final comparison (repeat... until...). In contrast again, the former solution employs the same logic for both loops.

In conclusion, both Kehoe's algorithm and the authors' are equally complex, taking into account when they were created. Perhaps the authors' presents fewer instructions, but the logic is practically identical, without a difference in performance. Both algorithms were tested using Microsoft Visual C# 2010 and both work perfectly.

Based on the authors' experience, the first algorithm meets the stated requirements. However, the expansion of its functionality through Kehoe's use of parameterization of the divisibility criteria does add to it.

#### **1.1.7. As the saying goes: "Every man to his own trade"**

Several centuries ago, scholars were able to follow a "Humanistic" approach, and study as many diverse subjects as they wanted. Mathematics, Physics, Chemistry, Poetry, Philosophy and so on, were all studied simultaneously.

With the expansion of human knowledge, each subject area has become more and more complex. So, as time has gone on, academics have not been able to cover all these aspects of knowledge in detail.

Nowadays, we live in a world where the catalogue of information available is practically infinite. It stands to reason that nobody can know everything. Technology is no exception, its infinite complexity is evident and it is not feasible that an individual would know the internal details of a machine, unless he or she is a professional in that area.

People have the freedom to learn as much as they want. However, we all face the same, epoch defining dilemma, lack of time. People do not have enough time to learn all about everything.

It can be asserted with some confidence that a lot of people drive their cars without knowing their basic mechanics. A lot of people buy pizzas without knowing how to cook. Almost everybody takes flights without knowing how to fly. These are some examples which validate the saying: “Every man to his own trade”, or as people say in Colombia “*Zapatero a tus zapatos!*”<sup>1</sup>

Let us apply this to an open-source vs. proprietary software debate. It has been stated that it is not necessary to “take control” of a system at a profound level, through manipulation of its open sources. Whereas others claim that a comprehensive education in computer science can only be reached through an understanding of open source philosophy.

Both these statements have are valid, people should take the appropriate course to achieve their personal goals. Every project begins with a purpose, if the way to achieve that purpose points

---

<sup>1</sup> ‘Shoemaker! To your shoes’. Author Note. (Translation from Spanish)

to the use a particular method, then whichever method is employed (open-source vs. proprietary software) to reach said goals is irrelevant; so long as the method fits the context. In conclusion, it is fair to say that a detailed knowledge of a machine is not necessary for an individual, unless it is specifically relevant to their context. This issued was raised by Glen Brookshear. (2009: 120).

### **1.1.8. Computing History: A Personal Perspective on Software Development and Storage Technologies**

Reflecting on the long history of computers, which have been present for more than a century of human existence, it stands to reason that these machines have had an impact on several aspects of society. Nowadays, computers are used in different capacities, such as communications, health care, transportation, security, surveillance, education, art, entertainment etc. Most work places use computers to manage information properly. Now, we have computers everywhere and we interact with them all the time.

Huge developments have been accomplished in a few years. In fact, some of the technological changes related to computing in the field of software development and storage techniques have been witnessed by the authors. Therefore, let us focus on some of our personal insights into these fields by way of a retrospective of some of the authors' early experiences in computing, compared to their current jobs.

First of all, one of the authors remembers their first contact with computer programming in 1987, in elementary school. In those days, some “new” Atari 600XL were delivered to the school, and his generation started to learn BASIC language to program those computers.

In 1987, the first programs were written using BASIC language on those machines. Atari consoles had “enough” memory to load and run simple programs; in fact, the amount of memory that they had was considered huge in those days. The Atari 600XL had full sized typewriter keyboard and it had BASIC in ROM. It had 16Kb in RAM, and it was able to handle peripheral devices such as floppy drives and joysticks (DuHamel, 1987). In fact, games which used to come in cartridges used to be played on those machines.

BASIC is a programming language inside the Imperative Paradigm, because written programs follow a sequential top-down design (Malcolm, 2001). Each statement of our BASIC programs had to be individually labeled, in order to differentiate between them. This feature allowed the sequence to keep its momentum while other programs ran. As a form of control, we were able to execute some “jumps” into our programs using sentences such as GOTO <label>.

BASIC was a great language for learning purposes because it was easy and fun to use. The language allowed us to explore our creativity, while still making us write several code lines to reach our goals. This feature however, was a serious problem when trying to debug a program or to add more modules (scalability). Another difficult task was the maintenance labor of programs, due to their long code lines.

However, a man who worked for a data center in the middle of 1980’s guided us though how to save information on disks using other systems based on mainframe machines. Floppy disks were used as a storage medium and the physical size of these disks was surprising, they were 8-inch floppy disks, as was the size of the floppy drive, it was huge, heavy and noisy.

At that time, while learning the basis of IBM mainframe systems, the complex process was saving information on to floppy disks using a set of statements in a command-line prompt. In fact,

the meaning of each sentence was not understood by us in those days, an 8-inch floppy disk was just inserted into its drive, the disk was locked in with a kind of weight, the sentences were typed, and the machine did the rest.

Fortunately, technology changed, and the range of software development and storage techniques improved. Today, the authors work for a small public university in southern Colombia. They teach software development and have noticed huge changes. Furthermore, storage technology also has been transformed in a meaningful way.

Nowadays, Microsoft .NET Framework is used as a development platform to build complex software. This software development technology is easier to use than previous approaches. Programming is done with a Framework-based language called Microsoft Visual Basic .NET inside the Object-Oriented Paradigm.

Some people used to think that Microsoft Visual Basic .NET was only a development of the old BASIC language. However, based on experience, Microsoft Visual Basic .NET is a brand new language whose syntax is similar to the old BASIC, but has a totally different approach.

There are no labels to identify each line of code in Microsoft Visual Basic .NET. Furthermore, code is organized in classes with attributes, properties, methods and events. Those programming concepts did not exist in the old BASIC language due to its procedural vision. In the classic BASIC, developers just had to follow Niklaus Wirth's formula on programming: algorithms + data structures = programs (Wirth, 1976). In contrast, developers in Microsoft Visual Basic .NET have to deal first with a software engineering process, because this kind of language was made for building complex systems instead of simple programs. In this vein, initial designs are required to face a computing problem.

Professional developers in Microsoft Visual Basic .NET used to design their systems before the development phase. This first phase used to contain logic diagrams (such as case-use diagrams, interaction, activities, etc.), class diagrams, and user interface designs, so on. The term “Visual” is oriented to environments which facilitate the creation of user interfaces. “Visual” technology was a great leap towards the creation of cutting-edge software, whereas classic BASIC lacked this feature.

Obviously, the more complex the software, the bigger the software will be. Storage technology evolved in order to handle higher capacity, more efficient and faster software. The 8-inch floppy disk’s capacity is 256Kb in MS-DOS FAT file system. (Tripathi, 2010: 1449).

At that time, those enormous disks had enough space to store several programs, documents, data and even small games. 1.2Mb was a large amount of space in the middle of 1980’s. However things are different now, individuals talk in terms of gigabytes and terabytes, while enterprises and large businesses talk about Petabytes and Exabytes.

Now, people use USB devices as removable storage units. These kinds of devices are able to store from hundreds of megabytes (the oldest ones) to terabytes (the newest ones).

The differences between floppy disk technology and USB technology are obvious; the former had a low storage-capacity, it used large devices, it was difficult to use and it was not as easy to transport. The latter has a higher storage-capacity, it uses small devices and it has the “plug-and-play” approach, which is easier to use. Furthermore, because of its size, USB technology is highly portable.



Technology will carry on evolving and in particular we hope to see more innovations in the field of software development and storage technologies. Perhaps in the future people will not need to type codes, instead they just will talk to development software, and physical storage devices will not be needed because of the capacity of cloud computing platforms.

#### **1.1.9. How will the existence of the Internet and database technology affect the existence of libraries themselves?**

Reading habits are changing. Brookshear (2009: 462) raised the social issue of how the existence of the Internet and database technology impacts the existence of libraries themselves. This is closely related to the way that people are choosing to read.

Nicholas Carr (2008) stated that the human brain biologically adapts, due to its plasticity. This is the scientific explanation behind why our brains tend to work like computers in the software age.

It has been suggested that Internet reading “style” may compliment biological facets of the brain. According to Rebecca Rosen (2009), the scanning-based style of Internet reading uses the seeking system in the brain, a natural biological evolutionary feature.

There is scientific evidence that proves that Internet reading is more a scanning activity than a reading one. Rebecca Rosen, (2009) showed that eyes follow scanning patterns when they read online, and based on these findings researchers tracked the eye movements of Web surfers, and found that more than 80% of people scan the text in front of them and keep browsing, rather than reading it thoroughly. This curious way of reading is related to natural activities of the brain.

The seeking system of the brain is based on certain biological adaptations in the human brain. Dr. Jaak Panksepp defined this system as a biological mechanism related to anticipation, curiosity, excitement, and the search for new stimuli (1998: 162). The seeking system has a direct connection with chemical reactions in the human brain, as it uses hormones to activate behavioral reactions. For instance, Rosen (2009) argued that Dopamine, which is naturally activated in the seeking system in response to evolutionary processes, drives people to get out the bed every day and to check emails every minute.

It stands to reason that the scanning-based style of Internet reading could fulfil a natural evolutionary adaptation. Consequently, the human brain, when is reading on the Internet, tends to scan texts, as a natural response to the biological adaptations looking for new stimuli.

In conclusion, we consider that the traditional libraries are at risk of extinction caused by the Internet and database technology, and furthermore due to people's new reading patterns.

#### **1.1.10. Machine Independence**

Since its inception, Computer Science has established the foundations for solving problems from a computing perspective. One of the many activities related to computing is programming, which constructs solutions to computing problems. Fortunately or unfortunately (depending on the outlook) programming has a close relationship with, if not a dependency on, hardware. In other words, software cannot “live” without the base that supports it, and this base is hardware technology which will be referred to as “machine” from this point.

When people develop software, the final product, which is called a "program", consumes hardware resources so that it can be used by the machine. Those resources include memory allocation, space in secondary storage units and processor usage time. Within the processor, programs are executed instruction by instruction, and each instruction must be written in its simplest form: machine code.

In general terms, machine code is expressed in numbers (in binary notation) which represent an identifier of the instruction set of the processor, a register, an interruption pointer, a memory address and data, among others. This is the first instance in which a hardware dependency can be observed, because each processor has its own set of instructions which may or may not be compatible with other machines. In the case of incompatibility with a set of instructions, programs simply will not run.

This raises the issue of lack of portability. It is the operating system which manages the machine's resources, sustaining the program's execution. These programs can be run only on the specific platforms for which they were built; in different contexts, programs will need to be rebuilt (recompiled) so as to guarantee their success.

The evolution of programming languages has led us to deal with the problem of portability through frameworks. In computer programming, a framework "is an abstraction in which software providing generic functionality can be selectively changed by user code, thus providing application specific software. It is a collection of software libraries providing a defined application programming interface (API)" (Wikipedia, 2011).

Many trends are being developed in order to enable High-Availability (HA) frameworks. Wong (2006: 45) states, that "HA frameworks and operating systems typically provide this by

insulating the developer from the underlying hardware. This provides a level of portability for developers as long as the HA framework supports the desired hardware and software platform”. Accordingly, the framework-based approach is one of the best ways to confront the issue of portability, because a framework deals with executing code, instead the operating system. With this software abstraction layer, portability is easily achieved, in a professional manner.

In this feature of computer programming, “there are a number of reasons why the analysis portion of a compiler is normally separated into lexical analysis and parsing (syntax analysis) phases” (Aho et al, 2006: 110). Aho proposes that portability is enhanced due to some specific peculiarities which can be restricted to the lexical analyzer, instead of the parsing process. This feature is suitable for framework-based environments.

The authors prefer developing software using the framework approach with Oracle Java and Microsoft .NET technologies. This is a professional way to take advantage of software potential to combat the toughest and most common problems. Furthermore, it is a good option for bypassing the portability problem.

#### **1.1.11. Types of Media: A Look at the BBC and Open Source Technology**

The 2008 Olympics was fascinating for many reasons, one of which was the technology involved, especially the incredibly high quality TV coverage from Beijing (China). One of the broadcast enterprises that covered the Olympics was the BBC.

In 2008, the BBC introduced a brand new technology to handle high-definition (HD) video signal for its broadcasting, using the Internet. This technology was called DIRAC, and it is defined

as “a general-purpose video compression family suitable for everything from internet streaming to HDTV and electronic cinema” (BBC R&D, n.d.).

The first examples of BBC’s video transmission from Beijing in 2008 were completely stunning, due to the quality of the signal via the Internet. This technology is able to compress byte streams and then encode them. According to its specification (Dirac Video Compression, 2008), its video format supports Ultra HDTV, because Dirac supports resolutions up to and over 1920x1080 and demonstrates meaningful improvements to MPEG-2 Part 2 and MPEG-4 Part 2. In addition, the authors state that Dirac has a higher performance than the most recent generations such as H.264/MPEG-4 AVC and VC-1.

The creators of this technology used lossless compression mode in order to meet the quality standards for video streaming via the Internet (Borer, T., Davies, T., Wilson, P., 2006). Furthermore, this technology was born of an open-source philosophy, therefore Dirac is designed for extensive use in diverse environments, for example, academic, research, entertainment, communication and media. Borer said of Dirac, that “it’s not just about being free in terms of money; we’re giving people the ability to customize it.” (Singleton, 2008: 50).

One of the surprising things about this technology is that it allows Ultra HDTV handling via the Internet; this feature is key for applications dealing with on-live video streaming services. Moreover, it seems that Dirac will be a great measure of quality when it comes to internet-TV series, based on its current performance and development support. Dirac is cataloged as VC-2 and based on the above the conclusion can be drawn that H.264/MPEG-4 AVC and VC-1 technologies will be overrun by VC-2.

### 1.1.12. A Note on Processes in an OS

It has been stated that a process is a program in execution (Silberschatz, Galvin, Gagne, 2005: 82). In general terms, a process has a process stack for temporary data, and a data section for global variables. A heap is a portion of memory which is allocated while the process is running. Finally, the set of instructions in a program is called a text section. These elements are represented in the table below:

Stack
↑ ↓
Heap
Data
Text

Table 1.2 Memory allocation structure

This is how a process is located in memory. All operating systems have to manage processes in order for machines to function. When an operating system is running, a lot of processes are running at the same time in front of the user's view. The order of execution of those processes is given by a diagram of process state (Silberschatz, Galvin, Gagne, 2005: 83), shown below:

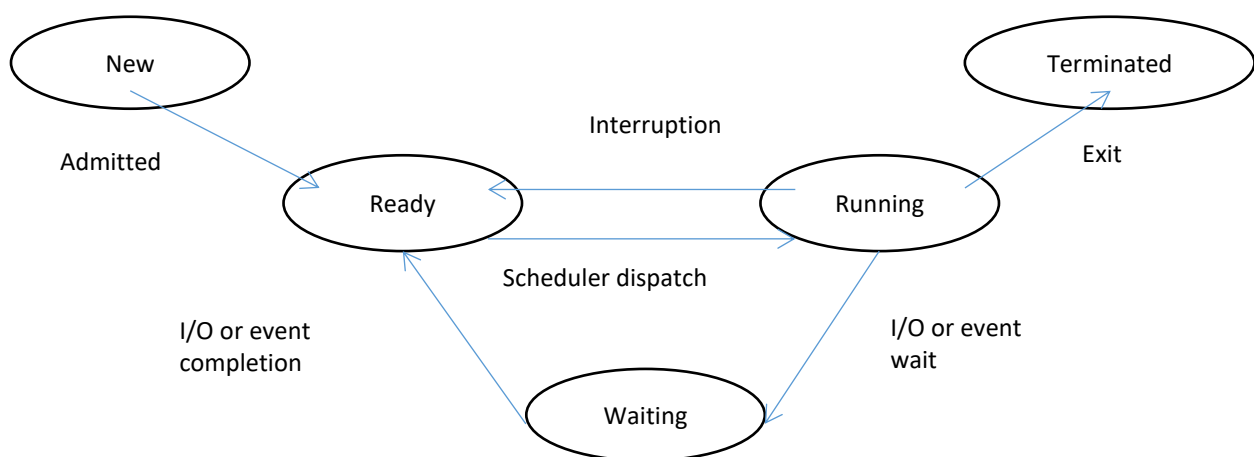


Figure 1.2 Machine processes in memory

A PCB (Process Control Block) is the structure that represents a task in an operating system. It is built by components such as: process state (new, ready, running, waiting, terminated), process number (an identifier of the process), program counter (an indicator of the next instruction of the program), CPU registers (data of the current value of the processor registers), memory limits (memory usage, addresses), and I/O status (a list of open files, devices in use, etc.).

An interpretation of the processes execution in an operating system is addressed in the following general steps:

- 1. A multiple queue structure is created via a linked list.**
- 2. If a new process arrives, it is queued in one of the listed queues.**
- 3. When a process reaches the front of the queue, CPU attends to it.**
- 4. If a process is terminated, it is released from the queue.**
- 5. When CPU attends to a process, the process state is updated.**

Table 1.3 General steps in processes execution

Several criteria can be applied to the previous steps. There are multiple algorithms which schedule processes in operating systems such as: FIFO, Shortest Job First, Priority-based, Round-robin, and multilevel queue among others.

The application of any algorithm depends on the operating system's inner structure, and its main purpose; that is to say, multiuser model, multitask features, etc. Taking into account these principles, queues and lists are used to schedule processes. It is evident that the usage of the heap and the stack, in a machine's memory, is very efficient, employing as it does a solid logic supported by a strong and professional design.

### 1.1.13. New Trends in Software Intelligent Agents

#### About Intelligent Agents

Several researchers and scientists have defined the term “agent” in different ways, which means there are several diverse definitions of what an agent is. The following definition of an “Intelligent Agent” was provided by Wooldridge, a professor of Computer Science and head of the Agents Research Group at the University of Liverpool; he stated, that “agents are capable of *independent, autonomous action* in order to meet their design objectives...they are capable of deciding for *themselves* what to do in any given situation” (Wooldridge, 2000: 1). Another definition of an “Intelligent Agent” was given by Brookshear (2009: 504), in which it is associated with a “device”, and responds to environmental stimuli.

Both these definitions imply a context in which intelligent agents perform actions which are influenced by the variables of the environment. Variables such as: Accessibility vs. Inaccessibility, Deterministic vs. Nondeterministic, Episodic vs. Non-episodic, Static vs. Dynamic, and Discrete vs. Continuous.

#### Applicability and Advantages

The need to build complex applications which are composed of several subsystems interacting with each other, has led to the creation of intelligent distribution frameworks, through agents. Agents do not need human will to operate, their autonomy allows them to perform activities in their area of expertise automatically. This is the main advantage they have over “traditional” software, and their domain takes in practically everything, more specifically: industry, process control, production, information management, monitoring and training, among others.



## **Limitations**

As previously mentioned, the extent of an agent's intelligence has some limitations, that is, the meaning of the acquired knowledge. Evaluation of the information processed, which can only be completed by the "human factor" is the limitation of these systems. Knowledge is not only data (mathematical or statistical), but also involves the appropriation of concepts through valuation of affective and emotional aspects that is characteristic of human beings (so far).

## **Current Trends**

Technological adaptations in biological organisms are one of the current trends in Intelligent Agents. For instance, the Rat Brain Robot is a brand new proposal in which artificial intelligence interacts with biological entities. This pitch was developed by Kevin Warwick and some students from the University of Reading (Upson, 2010).

### **1.1.14. Pros and Cons of Interpreted Code**

One of the important foundations of Computer Science is computer programming. Software development has existed since the middle of 20<sup>th</sup> century, and it has undergone significant changes throughout its evolution.

Following the timeline of programming languages, generations have been defined using a criteria of paradigms. In the case of the first generation of programming languages, people used to program in machine language using number-based instructions; the second one used a more "human" approach, employing mnemonics (assembler language) but programming tasks were still

tedious. Finally, the third and fourth generations allowed a close relationship with the natural languages as a “new” way to express computing solutions.

However, in each generation of programming languages there is a special concept related to the way in which these languages run their code: Interpreted or Compiled approaches.

A Compiled approach consists of an entire translation from source code to machine code. In designing a compiler, a language translation must be done before running a program, and that program “first must be translated into a form in which it can be executed by a computer. The software systems that do this translation are called compilers” (Aho et al, 2006: 1).

Although a compiled approach offers high performance in the execution of computer programs, this feature has problems with portability. When machine code is generated, programs depend on the platform on which it is running. In order to move computer programs to different kinds of machines, these programs have to be compiled again in those new contexts.

Which brings us to the Interpretative Approach, which opens up new possibilities in computer programming. By definition, “An interpreter is another common kind of language processor. Instead of producing a target program as a translation, an interpreter appears to directly execute the operations specified in the source program on inputs supplied by the user” (Aho et al, 2006: 2).

In accordance with the above definition, interpreted programming languages allow relative ease of programming, which means that the interpreter can execute each instruction once they have been written, in the source code. Another benefit of interpreted code is its high level of portability;

in modern programming environments, virtual machines and frameworks such as Oracle Java and Microsoft .NET technologies manage both compiled and interpreted paradigms.

With this technology, the most important thing in the interpreted scenario is that essentially the same code can run on different platforms. Software designed with this kind of technology can run easily via the Internet, in spite of its heterogeneous platforms.

However, the benefits of using interpreted code come at a price: decreasing performance. Within the context of frameworks and virtual machines, there is a software layer which interprets and runs code. Operating systems are not the main authority responsible for running code directly, the frameworks are. Nonetheless, nowadays issues of performance are questionable. People can access hardware technology with powerful features which can guarantee high-performance solutions. In these cases, problems related to decreasing performance are practically imperceptible.

#### **1.1.15. Solvable and Unsolvable Problems**

Using a computing approach, problems should be solved by algorithms. One of the features of algorithms is their finite definition according with Nicklaus Wirth (1985). This means that the problem solving process using algorithms must start at one point and finish at another one. With this in mind, unfortunately not all problems have a computing solution. In fact, computational complexity theory has an interesting list of unsolved problems. This text and complementary readings discuss this conundrum.

It is important to take into account that the origins of computing are linked to mathematics. Therefore, several problems which occur in mathematics are likewise reflected in computing.

Sometimes these problems have a computer-based solution, and sometimes not. For instance, the ancient problem of squaring a circle using only a compass and straightedge, proposed by classical geometers, now has a computer-based solution (Furst, C., Volk, M., Makeschin, F., 2010). In addition, Poincare's Conjecture now has a computer-based solution too (Wikipedia, n.d.). However, there are still many problems without mathematical solutions which are being studied by computer scientists.

The authors would like to put forward their personal perspective on solvable and unsolvable problems. A problem only has a solution in computing when the existing algorithms and processes satisfy all the requirements of the problem in question. An unsolvable problem occurs when no algorithm or process has yet been successfully designed that fulfils all the requirements of a given problem.

Consequently, some problems require a computation capacity that exceeds the current infrastructure. This infrastructure deals with memory, storage and processing capability. For example, according to Vandelli (2009), CERN is working with the LHC, which generates huge volumes of information for each fraction of a second of its usage. Therefore, almost all of the technology involved in the LHC experiments had to be redesigned and rebuilt. CERN faced an unsolved problem at the outset, because there was no computing technology that was able to handle the LHC during the preliminary years when the experiments began. However, now that "unsolvable" problem has been solved.

It can be drawn as a conclusion therefore, that all computing-related problems can be solved as long as there is a scientific and technological background that can support them.

### **1.1.16. Considerations on Distributed Databases**

Distribution as a concept relates to the capabilities of technology in the context of the decentralization of computing environments. This concept has been developed after networking-based technology; in this sense, telematics is the main facilitator of distributed computing, and its history spans over 40 years.

Ozsu and Valduriez define a distributed database as, “a collection of multiple, logically interrelated databases distributed over a computer network. A Distributed DBMS is the software system that permits the management of the distributed database and makes the distribution transparent to the users” (2011: 3). Following this definition, it stands to reason that the mere distribution of data containers is not enough to create a distributed database system. Ergo, it is necessary to implement control mechanisms whose purpose is to manage those data containers. That is why a DDBMS (Distributed Database Management System) is required.

In this vein, the main design benefits of distributed databases are geared toward large business environments. However, what is reason for distributing databases? “The classical answer to this question indicates that distributed processing better corresponds to the organizational structure of today’s widely distributed enterprises, and that such a system is more reliable and more responsible” (Ozsu, Valduriez, 2011: 3).

However, database distribution raises three main concerns: Firstly, that data may be replicated in a distributed environment. Secondly, that the communication infrastructure can fail. Thirdly, that the synchronization of transactions must be transparent for users. These issues are the current basis for study in several ongoing researches and new technology launches. For instance,

new algorithms are being developed in the field of synchronous or asynchronous communications; new hardware support is used to fulfil all the needs of distributed computing, and new techniques are being used to reduce the backup cost of distributed databases via redundancy criteria with optimal performance (Poordavoodi, Khayyambashi, Hamin, 2010: 27).

This raises some questions around the creation of a distributed database system, which can be outlined as follows:

### **When would you choose to implement distributed databases rather than local databases?**

A distributed database system would be implemented when the size and requirements of the organization requesting the system so warrant. More specifically there are two variables which justify the creation of a distributed database system:

1. Organization size: A distributed database system would be necessary for a large enterprise with international or national coverage. If this organization has several branches and each one of them manages a lot of data, a distributed database better suits their context.
2. Organization requirements: If an enterprise is dealing with mobile computing, which needs to interact with a lot of data, a distributed database system would grant a suitable solution for mobile and personal devices.

**Would this design work if all the computers are in one building, or only if they are spread out over the country or worldwide?**

In accordance with the previous answer, a computing solution based on distributed databases would only be made if the size and the requirements of the organization are oriented to this approach. Furthermore, a distributed database scenario would not be necessary for a small or local infrastructure, that is to say a network which consists of one building.

**When should mirror databases, or databases tailored for the local environment, be used?**

“A distributed database might contain fragmented and/or replicated data” (Brookshear, 2009: 422). When almost all information must be available at each point of the system, it is convenient to have some degree of redundancy (replicated data) over the network. In following this idea, mirroring techniques may be required in the creation of that system. In contrast, tailored databases for local environments are suitable when redundancy is not a necessary resource.

Other interesting points, are the criteria on performance and backup/recovery tasks. In this context, the usage of tailored databases could offer higher performance than mirrored databases. Furthermore, tasks associated with backup and recovery could be easier for local environments than if distributed database were used.

In conclusion, the authors’ preferred method is to use centralized databases based on web services (or even cloud computing) to create computing solutions by facing real and complex problems. The design and development of a distributed database would only be considered if the specific conditions explained above were met.

### **1.1.17. Some Impacts of Artificial Intelligence**

Since their arrival, computers have changed our lifestyles. Computing innovations and technology adaptations have diffused into almost every area of our lives; for instance, computers are present in education, entertainment, medicine and health care, environmental monitoring, communications, media, transportation, finance and marketing, to name but a few.

These days, computer-based solutions are known for offering high-speed information processing, reliability, portability and security. However, some of them employ what we might call “intelligent behavior” to face specific problems. Artificial Intelligence (AI) is the key player in the quest for finding rational solutions in the same way humans do, or better.

Professor Winston outlined a perspective, conceived at MIT, on Artificial Intelligence. He wrote, that “the primary goal of Artificial Intelligence is to make machines smarter. The secondary goals of Artificial Intelligence are to understand what intelligence is (the Nobel laureate purpose) and to make machines more useful (the entrepreneurial purpose)” (Winston, 1984: 1). Defining Intelligence, as mentioned in the above statement, is in and of itself a difficult task which spans more than one discipline (Brookshear, 2009: 504).

Many studies involving AI have high hopes of solving the world’s toughest problems. In the contexts of Biology and Neuroscience in particular, AI could realize its full potential. Robotics and Intelligent Control are areas which receive much attention and innovation in developing AI; in fact, currently the sector which is benefiting the most is industry.



However, there are some widely held concerns about the impact of AI on our society. One of the most common issues related to AI's implementation in a social context is evaluation from a human perspective.

Evaluation from a human perspective is a feature of our species which allows us to make judgments of value about someone or something. Human beings do not only handle and process information, but also we associate emotions and feelings with this information; this feature gives meaning to the information we process in the real world. A personal experience has meaning for an individual because he or she has associated values with that learning situation (both bad and good). Finally, evaluation from a human perspective is part of the field of affective computing in which emotions play an essential role in rational thoughts (Damasio, 1994).

Following this, one of the dangers of AI would be to allow machines to make decisions based on data (mathematics and statistics) without taking into account aspects of human nature (values, emotions, feelings and beliefs). A memorable example in pop culture comes from the movie *I, Robot* based on the book of the same name, written by Issac Azimov. In a car accident, two cars were submerged under water and the main protagonist asked a robot to save a little girl, but the robot made the decision to save the protagonist based on its calculation of their chances of survival.

#### **1.1.18. Threats against Network Security**

Currently, millions of people are connected to the Internet; the most of them use the network system for both work and personal use. Ordinary people sometimes experience problems

regarding the use of computer technology such as viruses, worms, malware and spyware among others.

These are common, everyday problems, but ones for which there are several computing solutions. The computer industry has developed a wide range of applications and system software to protect against these threats, such as firewalls, anti-virus software, etc.

To take a metaphorical stance, we could say that we are living in a technological world ruled by two forces: the good and the evil. It could be added that there is a kind of balance for computer users to find between these forces.

However, on a less domestic scale, there is a kind of threat to network security which could be regarded as one of the most dangerous threats to any enterprise, institution, or even government. That threat is hacking.

Following a study of the hacker's mind, the concept of the "hacker" has taken on negative connotations, of computer "intruders" who carry out destructive acts (Young, Lixuan, Prybutok, 2007: 281). Destructive acts constitute a crime which must be penalized. However, there is a sector of the computing community who state that "hacking" is a quite normal attitude which comes from a person who wants to know as much as possible about a subject, and for any of the following reasons: terrorism, sabotage, spying, revenge, a salaried job, vanity, egotism, self-esteem, or even plain curiosity. However, a "hacker" is obviously a person with a deeper knowledge of computer-based systems than your average internet surfer.

The fact is that hacking is illegal worldwide. However, the results of the aforementioned study show, that "hackers justify their behavior, perceive a low level of social sanction, a high

level of severity punishment from the court but a low likelihood of getting caught and receiving the punishment. Hackers also perceive a relatively high utility value resulting from engaging in illegal hacking activities.” (Young, Lixuan, Prybutok, 2007: 286).

These results are alarming because they show evidence of criminal conduct which is simultaneously addictive and contagious. Private information is at risk and these people could get access to it.

There are two suggestions for facing this problem: prevention and counterattack. Prevention implies minimizing hacking behavior. Mr. Stone (1999) argues that hacking potential begins at high school, and the way to minimize this behavior is through proper education and computer technology (software) which filters Internet-based contents in order to reduce the individual’s access to hacker resources. Additionally, parental supervision at home is also key in preventing hacking behavior.

The fight is being carried out by different government agencies with the aid of the software industry. This issue affects everything, so government agencies are dealing with this problem in first instance. The responsibility of governments is to ensure the welfare of their societies.

#### **1.1.19. Waterfall Development vs. XP Approach**

Over time, software development has evolved in parallel with achievements in hardware technology. The creation of software is one of the most common activities in the field of computing; in addition, global demand for software development has increased in the last few

decades. Consequently, the way software is created has undergone drastic changes in order to face new challenges.

Since its beginning, Software Development has been considered both an “art”, and a “formal engineering process”. Some programmers have established their own strategies for the creation of computer-based solutions which combine these two elements. However, in a professional arena, the development of complex software systems demands the usage of collaborative processes which allow coordinating software development activities involving several people. This is the point at which software development becomes a formal process.

One of the first approaches for constructing software was Waterfall development. The Waterfall model is a sequential design process, which was so popular in the 70's and the first half of 80's, due to its being easy to use with the imperative development paradigm. Its author states that everything related to software creation has to complete two main stages: analysis and coding (Royce, 1970: 328). With these two principles, Dr. Royce established sequential phases such as: Requirements specification, Design, Construction, Integration, Testing and Debugging, Installation, and Maintenance.

According to Dr. Royce's vision, large software systems were considered big enough for the technology at the time. Nowadays, current requirements dictated by globalization have created the need to develop software of huge proportions. The complexity, along with new challenges in computing, require cutting-edge features for the new software systems involving more scalability, more modularity, more robustness and more portability.

Today's development tools are more powerful than the previous ones. Furthermore, new theoretical approaches in software development feature in new founding paradigms. This is the

case of the object-oriented approach. By using the object-oriented approach, the software engineering processes become easier to implement (through the proper foundation of the paradigm, of course). In this order of ideas, professionals who are involved in a software construction process can coordinate their activities in order to achieve high quality programming. This is true in the case of the eXtreme Programming (XP) Approach.

Mr. Holcombe stated that XP has many benefits which have led to its success in software development, one of them being its high level of user interaction. According to Holcombe “Software development is a human activity, and we must ensure that the human dimension is at the center of our thoughts when we discuss ways to make software more effectively” (2008: 19). Holcombe further outlines five essential values: Communication, Feedback, Simplicity, Courage and Respect.

There are many cases in which XP has been used with successful results. XP gives designers and programmers a level of “freedom” which is not available in other strict methodologies. In contrast, the Waterfall approach demonstrates a rigid view of the software construction process. For example, documentation is highly valued in the Waterfall model; whereas in XP, documentation is just a means of support for the development processes, and often this documentation does not follow an international standard, only the team’s own set standard.

Finally, an important aspect to take in account is the maintenance capability of the computer-based final solutions. In the waterfall model, a “trial-and-error” approach is all but avoided (Brookshear, 2009: 334); however, this technique compliments agile methodologies, and especially XP.

## 1.2 WORKSHOPS

### 1.2.1. Workshop 1

#### *Ambiguity in a proposed algorithm and an ambiguity in the representation of an algorithm*

Ambiguity in a proposed algorithm could present itself when the idea behind the solution has one or more steps which lack definition. This kind of ambiguity is related to lack of clarity in one or more steps of the algorithm. The most important thing to note, is that a proposed algorithm is just an idea, and this idea could have more than one interpretation depending on the reader.

In the same way, ambiguity in the representation of an algorithm could surface when one or more primitives, have more than one interpretation. A primitive should have only one interpretation with the purpose of executing a pre-defined task step by step.

In addition, both in terms of abstractions of thinking and primitive definitions, the interpretation depends on the level of knowledge of the individual who is “reading the idea”, because there are some terms that could be understood in different ways in a single step. Furthermore, the amount of detail provided at each step of the algorithm is a key factor which affects ambiguity.

#### *How the use of primitives helps remove ambiguities in an algorithm’s representation*

Primitives are the basic elements in the structure of a language. In Computer Science, those primitives applied to programming languages allow new definitions of rules; they establish the

structure of the language and provide a logic behind their usage. Therefore, programmers must use those predefined primitives to write computer programs.

In algorithms, a primitive represents one, and only one, specific instruction in the same way as in programming languages, but with the distinction that in algorithms, primitives are one step of an idea. This feature helps to avoid ambiguity, due to the strictness of the primitive, and this means that a primitive has been defined to “process” one specific function in the algorithm, and it connects to the sequence of the algorithm itself as only one step. The more detailed the primitive, the less likely it is that there will be any ambiguity.

### ***What is the difference between a formal programming language and a pseudo-code?***

In Computer Science, pseudo-code is a tool that is used to express algorithm logic independently of any formal programming language. Pseudo-code was conceived to facilitate the comprehension of a design logic as an algorithm; it is very similar to human language in the sense that it enables understanding and diffusion. In general, a pseudo-code expresses an idea, but no real solution in terms of technology implementation.

In contrast, a formal programming language is a tool which allows a solution to be implemented as a computer program. In this vein, programming language is able to create a computer-based platform in which a real solution can run; the end users can interact with that platform, so they can see real results according to their individual requirements. Formal programming languages have their own syntax, semantics and grammar; therefore, a computer-based solution must be written following the strict regulations of the chosen language.

### ***What is the difference between syntax and semantics?***

Syntax and Semantics are two different concepts in formal languages.

Syntax is the feature related to formal languages, which governs the way their components are used, such as lexicon, expressions, symbols and operators among others. Syntax determines how those elements are used in terms of order and composition, and it validates statements as legitimate according to specific rules. In other words, in Computer Science, syntax deals with how statements in a language must be constructed.

Semantics is the feature related to formal languages, which governs the meaning of their statements. Semantics provides meaningful expression in language, which means that semantics identifies a clear purpose behind a set of instructions which expresses logic and rationality. In short, in Computer Science, semantics deals with the meaning of the statements in a programming language.

Both concepts are the basis of the creation of grammar for compiler design. Syntax takes tokens from the lexicon to create a representation according to the grammatical structure, and Semantics checks whether those structures are consistent with the language definition in terms of behavior, data types and meaning of statements (Aho et al, 2006: 8).



The following is a problem which can be solved using an algorithm: Four prospectors with only one lantern must walk through a mineshaft. At most, two prospectors can travel together and any prospector in the shaft must be with the lantern. The prospectors, named Andrews, Blake, Johnson, and Kelly, can walk through the shaft in one minute, two minutes, four minutes, and eight minutes, respectively. When two walk together they travel at the speed of the slower prospector. How can the prospectors get through the mineshaft in only 15 minutes? After you have solved this problem, explain how you got your foot in the door.

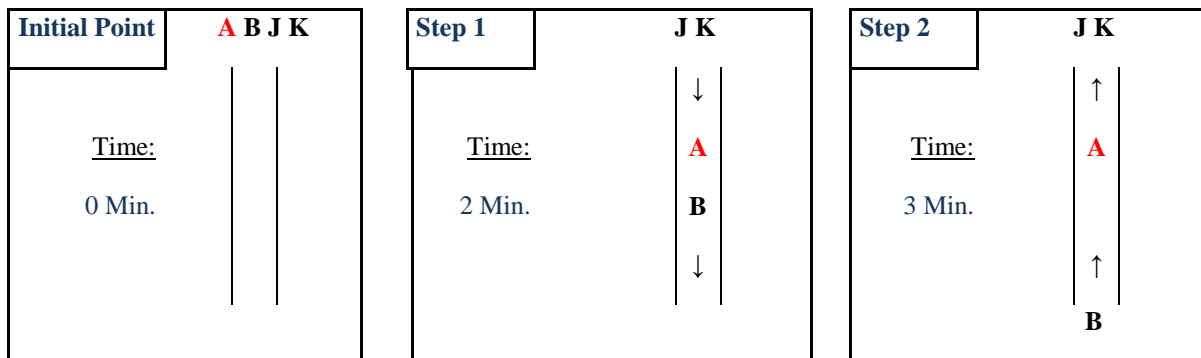
The algorithm that solves the problem in only 15 minutes, using just 5 steps is described below:

1. Andrews (with the lantern) and Blake go first through the mineshaft.
2. Blake waits and Andrews (with the lantern) goes back delivering the lantern to Johnson.
3. Johnson (with the lantern) and Kelly go through the mineshaft, while Andrews waits.
4. Johnson delivers the lantern to Blake who will return for Andrews, Johnson and Kelly wait.
5. Finally, Andrews (with the lantern) and Blake go through the mineshaft.

Table 1.4 Algorithm for the solution

The same algorithm is expressed in a graphic form:

Red Character brings the LANTERN



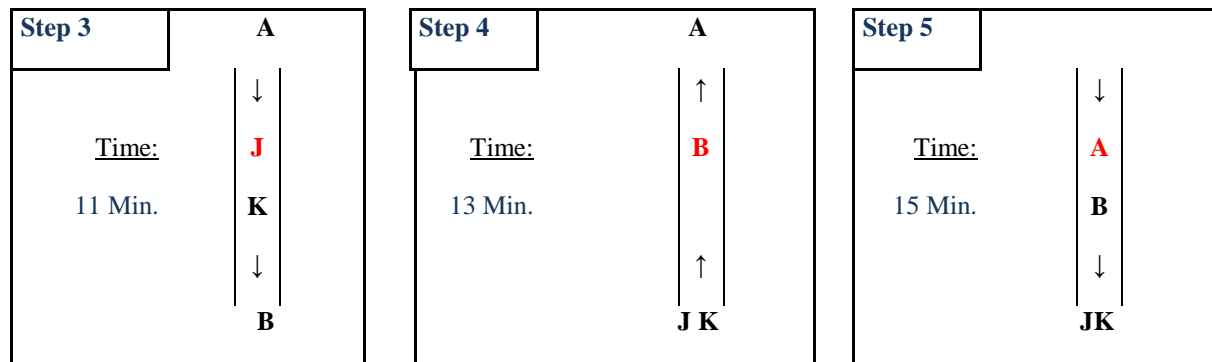


Figure 1.3 Graphic solution of the applied algorithm

The main criteria taken into count were:

In order to reach the goal of 15 minutes, the fastest people must start and finish the travel (criterion taken from athletics). That is why Andrews and Blake went in the first and the last step.

In order to reach the goal of 15 minutes, the slowest people must travel together with the purpose of spending the lowest number of possible steps. That is why Johnson and Kelly traveled together in the middle of the process.

### 1.2.2. Workshop 2

*What does it mean to say that a programming language is machine independent?*

A programming language is considered machine independent when its final products (such as an executable code or an object/intermediate code) can run on different platforms (hardware and operating system) with no changes to the source code.

This duality can be reached by means of software frameworks and programming virtual machines. The most relevant examples of this kind of approach are Microsoft .NET Technology and Oracle Java Technology.

Both technologies were created for the construction of huge scalable software systems. One of the multiple fields of development is the Web/Cloud environment. In the aforementioned context, programmers can spread their creations using compiled code and interpreted code (specifically in server and client respectively). In this example, interpreted code for a web-based solution (JavaScript, JScript, ECMAScript, among others) is the best for achieving machine independency.

#### ***Four different programming paradigms***

Imperative Paradigm: This paradigm follows the sequential logic of steps (one step behind another). Its execution is ruled by control structures such as conditionals and loops. Its logic points toward the classic computing model of inputs, processes and outputs. Some examples of this paradigm are FORTRAN, Pascal and C. (Gabbrielli, Martini, 2010)

Functional Paradigm: This is based on mathematical foundations, specifically the Lambda Calculus. Its statements follow the logic of the theory of functions in mathematics; in consequence, everything is handled as functions, with their parameters being input values, output values, and input-output values. This paradigm exploits computational concepts such as iterations and recursion. An example of this paradigm is Scheme, F#. (Gabbrielli, Martini, 2010)

Logic Paradigm: It is based on knowledge extraction from basic facts and their relationships, and also on axioms declarations, queries and inference principles. This kind of paradigm is highly suitable for developing computing solutions in the field of artificial intelligence. An example of this paradigm is Prolog. (Gabbrielli, Martini, 2010)

Object-Oriented Paradigm: This paradigm tries to represent reality in the form of classes and instances (objects). It is based on the principle of encapsulation which is an organized way to build information packages. Those packages can handle their own logic and behaviors through the implementation of methods and events. In addition, inheritance and polymorphism are relevant concepts related to extensibility and functionality. Relations between classes allow the modelling of complex information systems using professional techniques such as re-usability and scalability. Due to its popularity, there are several programming languages which belong to this paradigm; for example: C++, Java, Visual Basic .NET, C#, etc. (Gabbrielli, Martini, 2010)

### ***What is structured programming?***

Structured programming is an approach to programming languages which allows us to take control of the program's execution sequence through conditionals, loop statements, and code-division, via procedures and functions (previously known as subroutines). This approach strictly follows the principles of algorithms on sequences, finite steps, unambiguity and precision.

Before this paradigm, non-structured languages were based on labels (tags as identifiers for each statement or group of statements) where instructions like "jump" or "go to" were used in order to change the program's flow sequence. Debugging and maintenance tasks were a complete nightmare in non-structured languages.

Nowadays, practically all programming languages (even in different paradigms) preserve the structured programming approach in one way or another.

***What does it mean to say that a programming language is strongly typed?***

Variable/object declarations allow a relationship to be established between identifiers and data types. In some programming languages, a variable (or object) which was created with an initial data type, can later store other kinds of data types without encountering any problems. For instance, in JavaScript, programmers can declare a string of variables at the beginning; then in later statements, the same variable can store integers, floats, dates, Boolean values, etc.

In the above case, the language is loosely typed because there are no problems with the indiscriminate usage of values for a variable. Programmers assume the responsibility of controlling their data.

Conversely to loosely typed languages, strongly typed languages *do not* allow the storage of a value if it has a different data type from the variable/object definition. In this case, a strongly typed language needs to carry out conversion tasks. This means that a type-mismatch situation must be overruled using data type conversion (casting in some platforms) with the risk of failure at some point in the process. (Aho et al, 2006).

### 1.2.3. Workshop 3

#### *An example of how efforts in the development of software can pay dividends later in software maintenance*

In a study about maintenance costs, it was found that nowadays software maintenance accounts for over 90% of a project's total cost (Koskinen, 2010). This is an alarming situation that causes concern to many software designers and developers. Furthermore, Seacord, Plakosh and Lewis stated that "In each ensuing increment, the portion of the legacy system that has not yet been replaced, as well as code already modernized, needs to be maintained. Unlike development costs, which occur only once, maintenance costs are recurring" (2003: 121).

With this in mind, in order to improve the dividends in the maintenance phase, a modular approach may turn out to be one of the most valuable design techniques for large systems.

An example can be seen in the creation of a brand new ERP system. The designers should break it down into integrated modules such as Inventory, Purchase, Production Planning and Control, Sales and Marketing, Exercise, Finance, HR and Payroll. Every module has its own cost of development and maintenance; then in this vein, a well-conceived modular design could give them the opportunity to increase dividends in the maintenance phase, because each section of the organization is used to request more features along the deployment/testing phase for its modules. This growth factor could be used as an estimation tool to determine the size of each new requirement (increment) (Seacord, Plakosh, & Lewis, 2003: 121), and this growth factor could be a good initiative to earn money postproduction.

On the other hand, the authors personally prefer to construct computer-based solutions in a cloud environment, where we may charge for the rental of our applications with a maintenance fee.

### ***How the lack of metrics for measuring certain software properties affects the software engineering discipline***

Sir William Thompson (aka Lord Kelvin), a famous mathematical physicist and engineer, said: “If you cannot measure it, you cannot improve it”. In terms of quality improvement, this legacy is highly relevant when applied to engineering, and the field of software production is no exception.

Some features of a software development process are feasibly measured, such as the number of code lines, performance and response times, among others. However, there are some elements of software production which cannot be measured, such as certain features pertaining to the “human” factor. For instance, work schedules, the level of expertise of programmers and social relationships in workplaces, which can all influence the entire software engineering process. Kan stated that, “In general, software quality metrics are more closely associated with process and product metrics than with project metrics. Nonetheless, the project parameters such as the number of developers and their skill levels, the schedule, the size, and the organization structure certainly affect the quality of the product” (2002: 65).

In the opinion of the authors, all features strictly related to software are feasibly measured, this includes size, number of code lines, usability, portability, fault tolerance and even user’s satisfaction levels, among others. In conclusion, the human factor has a significant impact on

software engineering as a discipline. In some contexts it is possible to measure the programmer's level of skill; nonetheless, some aspects such as their behavior in the workplace and their relationships with the professional staff are a subjective, and therefore unquantifiable matter.

***What complications arise if one tries to implement a dynamic list using a traditional homogeneous array?***

Dynamic structures are designed to manage large volumes of data in an efficient way. Dr. Kalicharan (2008: 53) stated: "One question which arises is how many nodes will there be in the list? This, of course, depends on how many numbers are supplied. One disadvantage of using an array for storing a linear list is that the size of the array must be specified beforehand".

The allocation of memory "as needed" is called dynamic memory allocation; the opposite being memory stored in array-style structures, which is called static storage. Performance can be affected by the static storage approach. In addition, operations such as insertion or deletion of items are less efficient in static structures (arrays) due to the following:

1. The insertion of a new item in a traditional homogeneous array implies its growth in elements, but these static structures have an initial pre-defined size (number of elements) from their inception; therefore, the array must be redefined again. In a dynamic context, a dynamic structure will grow with no problems (depending of course on the amount of memory they have).
2. The deletion of an item in a traditional homogeneous array implies the creation of empty spaces in the static structure. In terms of memory, an item in a traditional homogeneous array cannot be deleted, instead, an item can be re-initialized (with 0 in the case of numbers or an empty



string for alphanumeric data types) but the item stays in the memory. So, the concept of freeing up memory space pertains only to dynamic environments.

### ***How an array could be used to implement a queue in a high-level language***

The best way to describe the usage of an array as a queue is using the following algorithms:

Start by assuming a *MAX* constant which represents the maximum number of items in the queue (the array), and the global instance of the *current position* variable which starts with 0;

<pre> <b>procedure</b> ShowQueue   i ← 0;   <b>while</b> (i &lt; MAX) <b>do</b>   (     <u>print</u> Queue[i];   ) <b>end while</b> </pre>	<pre> <b>procedure</b>NewItem   <b>if</b> (currentposition = MAX)   <b>then</b> (<u>print</u> "More items are not allowed..."; )   <b>else</b>   (     Queue[currentposition] ← <u>read</u> "a new item: ";     currentposition ← currentposition + 1;   ) </pre>
<pre> <b>procedure</b> DeleteItem   <b>if</b> (currentposition = 0)   <b>then</b> (<u>print</u> "The queue is already empty..."; )   <b>else</b>   (     i ← 1;     <b>while</b> (i &lt; MAX) <b>do</b>     (       Queue[i - 1] ← Queue[i];       i ← i + 1;     ) <b>end while</b>     Queue[MAX - 1] ← 0;     currentposition ← currentposition - 1;     <u>print</u> "An item was deleted...";   ) </pre>	

Figure 1.4. Pseudocode

#### **1.2.4. Workshop 4**

***What is meant by data independence? Explain your answer.***

Rob and Coronel state, that “data independence exists when it is possible to make changes in the data storage characteristics without affecting the application program’s ability to access the data” (2007: 15). According to this concept, data independence is a feature of database systems related to the isolated (but conceptually associated) application layer and database management system. Following this, applications should be able to manage the user interaction of information through user interfaces; these applications do not need to be re-programmed when the structure of the data is changed at the level of the database management system.

It is important to note that data independence eases the development of upgrades; because, by following this approach, modifications in the data structure can be implemented at a user application level. Within a software engineering context, some models could benefit from the implementation of data independence when managing increments, new versions, and new functional prototypes, as they would not need to rebuild everything from scratch.

***Two benefits of separating application software from the database management system***

Software architecture patterns can define elements and their relationships with a view to solving specific problems. Architectural patterns are a key concept in software engineering which support documentation decisions, communication with stakeholders and provide relevant information on both the problem definition and its solution (Avgeriou, Zdun, 2005: 4).

There are some architecture patterns which focus their approaches on the separation of application software and the database management system. One of these is called State-Logic-Display Pattern (a.k.a. Three-Tier Pattern). According to Taylor, Medvidovic, and Dashofy (2009: 94), a State-Logic-Display (SLD) pattern has a well-defined software deposit structure; it has a linear relationship between the 3 units: Presentation (Display or User Interface), Business Logic (Main behavior of an organization, algorithms), and Data (State or persistent data access). An example of such a scheme is detailed below:

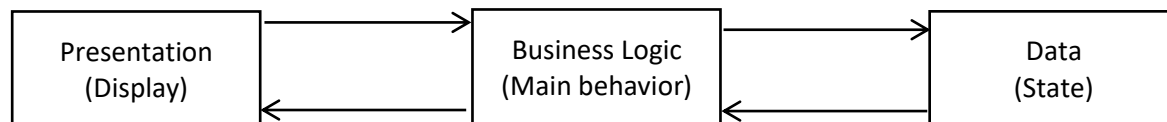


Figure 1.5 State-Logic-Display Pattern. Based on (Taylor, Medvidovic, Dashofy, 2009: 95)

An important factor here, is the level of independence of each of these tiers. Each tier has its own mechanisms for communicating information to its neighbor; however, each tier also has its own functions. It therefore stands to reason that application software is separate from the DBMS. Two benefits of this approach are outlined as follows:

- Maintenance and Upgrade purposes: With this approach, a computing solution driven by this model can be supported for long time due to its modular design. In addition, this kind of separation through software units allows for a high level of scalability (the ability to grow).
- Security and Performance: The usage of stored procedures, which are embedded in the database, is a strategy which increases performance in large queries and transactions.

Furthermore, in this scenario, all or part of the transactional functions would be written on the database. Therefore, an unauthorized user would encounter more constraints in trying to get access to the source code, because he or she would have to overcome the OS restrictions, Firewall rules and finally the security levels of the Database engine.

***In what ways does an operating system's file directory differ from a traditional index?***

The relationship between the two has been expressed in the following way: “a prominent example is the hierarchical directory...directories or folders play the role of indexes...From this perspective, the entire file system is merely one large indexed file” (Brookshear, 2009: 447). The recursive usage of the hierarchical structure of directories can be compared to the functionality of an indexed file. That is, when an indexed file's in-memory is a “dynamic tree”.

Nevertheless, there are some differences between these two concepts. First off, the terminology surrounding file systems adds to their complexity. For instance, in a file system structure (no matter what type of file system or its platform) there are some terms (such as disk, block, partition, volume, metadata, i-node, extends, among others) which are relevant to its construction (Giampaolo, 1998: 8). In other words, the inner structure of a file system is more complex than that of an indexed file.

Furthermore, the mapping of an item's position is more complex in a file system than in an indexed file. The former needs to deal with disk blocks, and also needs to calculate sufficient space to locate a file within the main structure and its relationship with the physical storage device (Giampaolo, 1998: 15). In contrast, an indexed file just has to deal with pre-established units of the same size when locating items in a file of specific contents.

- **Advantages that a sequential file has over an indexed or a hash file.**

A sequential file is easier to implement than an indexed file or a hash file. In a sequential file, developers just implement an ordered and consistent structure, with a view to storing data in a linear manner. In an indexed file, developers need to include all its content in the form of a dynamic structure in-memory. In addition, in a hash file, developers need to modify items' positions using searching and sorting algorithms.

- **Advantages that an indexed file has over a sequential file**

Efficient searching is the key feature available in an indexed file. When searching for information in a sequential file, the linear process takes longer. However, an indexed file is represented in memory as a dynamic structure and therefore searching is faster.

- **Advantages that an indexed file has over a hash file**

An indexed file performs better than a hash file under certain circumstances. The main function of an indexed file is to support the location and searching processes for the information contained in a structured file. These tasks are achieved through dynamic representation in memory using data management algorithms. In contrast, a hash file has these tasks incorporated, and some more besides. For instance, a hash file uses inner functions (hash functions) to establish the location of individual items. The use of hash functions could impact the performance in data access when range queries are required. Furthermore, indexed files do not need to deal with the problems of clustering and overflows.

- **Advantages that a hash file has over a sequential file or over an indexed file**

A hash file has more security than the others, due to its data verification. In data transportation scenarios, a hash file is more reliable because of its verification algorithms, so information can always be confirmed.

### **1.2.5. Workshop 5**

#### ***Different meanings of “Do you know what time it is?”***

Situation #1: Two people waiting in a bus stop. A person asks to the other one: “Do you know what time it is?”

Meaning #1: One person looking for the exact time at that moment.

Situation #2: A wife waiting at home for her husband. The man comes late and he is a little bit drunk. The woman says: “Do you know what time it is?”

Meaning #2: A reprimand.

Situation #3: An instructor in a museum raises a question to the audience (a group of kids from an elementary school): “Do you know what time it is?” while the instructor points with a finger toward a picture with an Egyptian Solar clock.

Meaning #3: A clue about the nature of an ancient instrument.

#### ***Some examples of declarative knowledge and procedural knowledge***

Generally speaking, intelligence is a broad field of study that takes in several disciplines such as biology, neuroscience, psychology, and computer science among others. Most commonly

definitions of intelligence are taken from cognition studies, and that is perhaps why Adams made the following association between neuroscience and types of knowledge. He stated that, “there is one type of knowledge- that involved in what neuroscientists call declarative knowledge and there is another type of knowledge- that involved in what neuroscientists call procedural knowledge” (Adams, 2009: 109).

According to the above idea, declarative knowledge is related to things, concepts, definitions and structured ideas, which are stored in a knowledge repository; in other words, declarative knowledge deals with information itself and its representation as knowledge. In contrast, procedural knowledge is related to actions, processes, methods and ways to obtain knowledge; that is to say, procedural knowledge deals with how to get knowledge in a learning context through practical resources (Bitan, Karni, 2004).

In a labor context, an example of procedural knowledge can be applied to construction workers. They are used to doing tasks which are based on repetition; therefore, construction workers exercise their knowledge in the performance of tasks such as building brick walls, columns of concrete, etc. Therefore, experience gives them knowledge they need to do their jobs.

In an academic context, an example of declarative knowledge can be seen in the first arithmetic lessons at elementary level education. Association processes between objects and symbols are the base for the construction of declarative knowledge about Numbers and Quantities. For instance, an early lesson may be developed using boxes and apples; the teachers prepare an exercise by putting a certain number of apples into different boxes, and using this, exemplify the number of apples in each box with its symbol (the mathematical number). Children can “store” the information as declarative knowledge.

### ***Difference between state graphs and search trees***

According to Brookshear, a state graph is a representation of a production system that solves a specific problem. This concept was inherited from the directed graph in mathematics (2009: 517). In general terms, a state graph involves sets of states and sets of productions. In contrast, the same author defines a search tree as a searching strategy whose main aim is to reach a goal more efficiently. This technique is represented as a tree structure that makes up part of a state graph (2009: 519).

The main difference between state graphs and search trees are:

- Size. If a search tree is just a part of a state graph, then the graphic representation of a search tree must be smaller than a state graph.
- Notation. A state graph is a directed-edge and node-labelled graph, this means that the visualization of a state graph can spread in multiple directions. A search tree has a hierarchical structure with a top-down approach.
- Functionality. A state graph puts an idea into the form of a structure. A search tree can be built according to search algorithms; for instance, Depth First Search, Breadth First Search, Best First Search.

### ***“Traveling from one city to another” – A production system’s view***

In the case of traveling from one city to another, a scenario is created with a set of states, a collection of productions and a control system. In general terms, these steps are followed:



- Locating the cities and the midpoints between them
- Locating the routes which join midpoints and cities
- Setting the rules for those routes when a passenger is at a midpoint or in a city (lengths, trip time, costs, schedules)
- Finally, fire up the goal of that system.

A fairly simple example has been outlined in the form of a graph as follows:

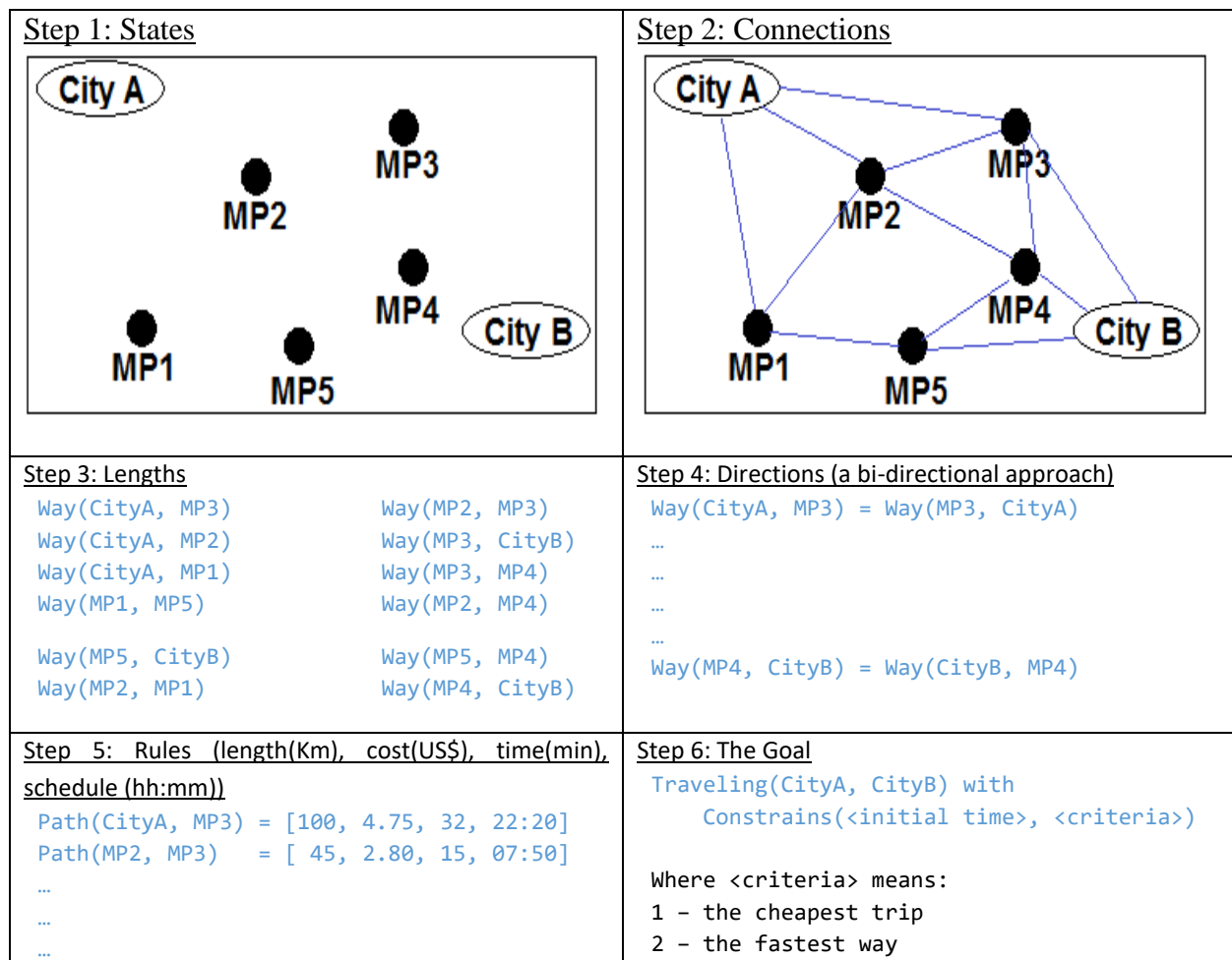


Figure 1.6 Graphic interpretation of the algorithm

## **2. COMPUTER COMMUNICATIONS AND NETWORKING**

### **2.1. REFLECTIONS**

#### **2.1.1. A Bad Day for ARPANET**

On October 27<sup>th</sup> 1980, the ARPANET crashed for several hours because some high priority processes exhausted the system's resources. In this case, an out-of-control process provoked the generation of faulty sequence of control packets, directly affecting the IMP processes; the consumption of resources was excessive, causing the breakdown of the ARPANET (Rosen, 1981: 1).

The main feature affected in that situation was Internet Message Processor (IMP). An IMP was a final protocol for communicating with IMP-based machines to send messages on the network, in this case: ARPANET. The entire IMP-based solution was composed of hardware and software. From a hardware side, a Honeywell DDP-516 was selected as the IMP processor; the software was created for processing packets. However, the resources of those machines and their software were limited to 6000 words of memory. (Edmondson-Yurkanan, n.d.)

That is why an out-of-control process consumed the all the available memory on those machines. In the modern context, such a problem could be solved by network administration tools which are in charge of monitoring resource consumption on the network.

Following the IMP devices of the ARPANET era, current devices that allow routing processes can deal with network resource consumption. This is true in the case of a software-based solution called Embedded Resource Manager (ERM) produced by Cisco.

The architecture of Cisco's ERM is shown below. In this case, the ERM framework establishes the parameters of managing common resources, including Buffers, CPUs and Memory. This software works in parallel with Cisco's IOS deployed along the network – especially in routers – in order to detect and report failures.

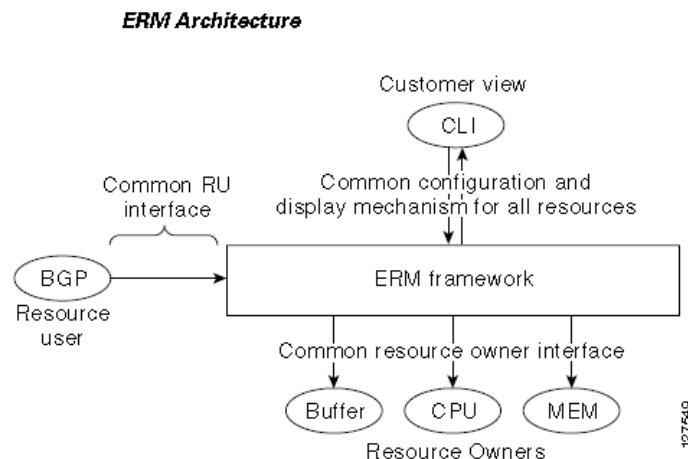


Figure 2.1 ERM Architecture

Source: (Cisco Systems, 2008: 3)

It is important to note that this is just one solution of several available on the market. In the specific case of Cisco's solution, ERM is 100% fully-compliant with world-wide standards such as MIB. Indeed, Cisco Systems provides a full platform of tools called Cisco IOS MIB Tools. The managed objects (MIB), or variables, can be set or read to provide information on the network devices and interfaces (Kurose & Ross, 2010: 778).

### **2.1.2. Audio Streaming**

Streaming technology is related to the consumption of media such as video and audio via the Internet; streaming services can be used in two ways: streaming stored media and streaming live media. The former is about publishing stored video and audio on the server; this means that audio and video are previously stored as independent files on the server's file system, and they are played on demand by clients through the network (Kurose & Ross, 2010: 599).

There are several corporations in the industry that offer stored streaming services, two examples are Real Networks Incorporated and Microsoft Corporation. In the following paragraphs, their main features will be analyzed.

Real Networks is a company focusing on the management of digital media, it produced the first audio streaming solution on the Internet in 1995. Through RealPlayer, its main product, it is possible to listen to music and watch videos via streaming (Real Networks Inc., 2012). Today, many companies offer streaming media services using Real Networks technology.

Real Networks technology in terms of streaming services is made of two parts: client application (The RealPlayer) and server platform (Media Entertainment Platform). In order to deploy streaming services, a configuration tool provided for Real Networks is used within the Real Media Entertainment Platform. On this a platform, which belongs to the Software-as-a-Service (SaaS) paradigm, the configuration tool for the media content's deployment establishes the Network Protocol for default as RTSP, and this network protocol runs over UDP. Although UDP is the default (and recommended) option, it is possible to deploy streaming services via TCP while

taking the risk of incurring more network overheads due to the reliable style of the protocol (Real Networks Inc., 2008: 15).

On the other hand, Microsoft Corporation also offers a streaming solution through its products and solutions. In particular, Microsoft has a long tradition in the development and deployment of web-based solutions via Microsoft .NET technology and Microsoft Internet Information Services as a default web server. The latter platform provides the resources to publish and deploy web content; in the case of rich media, Internet Information Services are extended through IIS Media Services.

Microsoft IIS Media Services is a platform that works in conjunction with Microsoft's Web Server, known as Internet Information Services. IIS Media Services can carry out HD live video and on-demand streaming media. In the particular case of on-demand streaming media, IIS Media Services uses a special protocol called Smooth Streaming Transport Protocol.

According to the official specification of Smooth Streaming Transport Protocol, "SSTP provides a means of delivering Media from Servers to Clients in a way that can be Cached by standard HTTP Cache Proxies in the communication chain"(Microsoft Corp. 2009: 7,8). This means that Microsoft's Smooth Streaming Transport Protocol uses HTTP resources to set out media content; ergo, the transport protocol is TCP. The advantages in this regard are related to the reliability of the offered service; however, the performance could be affected by that same factor, reliability.

The choice of which transport protocol to use is very important; global experience in media streaming demonstrates that UPD is more adequate. This notwithstanding, the authors believe that

the physical infrastructure (communication channels) and software/hardware technology in both client and server sides are key factors in terms of the performance of a streaming-based solution.

### **2.1.3. American Registry for Internet Numbers: Whois Service**

According to the RFC 3912, Whois is a protocol that provides information services about sites to end users. Whois works over TCP-oriented connections and this protocol is implemented by certain organizations in order to provide information about those websites which belong to the registry (Network Working Group, 2004: 4).

Historically, Whois has passed through four stages (Mueller & Chango, 2008: 307-308):

1. The NICKNAME/Whois directory service (early 1980's)
2. Whois service global directory (early 1990's)
3. ICANN's Whois (mid 1990's)
4. "The Endless Contention" (from 2001 until now)

One of the main issues related to the usage of Whois-based services is data protection and privacy. Privacy concerns about the usage of the global Whois Database were (and still are) a key point of attention for people who rule the Internet-based addressing (Mueller & Chango, 2008: 315). Under the U.S. Congress regulations on the matter of security and privacy, Whois service of the ARIN (American Registry for Internet Numbers) still continues to function. The current ARIN's Whois metadata is composed of Name, Handle, Street, City, State/Province, Postal Code, Country, Registration Date, Updates and RESTful Links, among others.

In a practical experiment, three universities were found by ARIN's Whois Service: San Jose State University, University of California at Riverside, and Indiana University/Pardue

University at Indianapolis. The Information of each organization is expressed in the following tables:

Name	San Jose State University
Handle	SJSU
Street	One Washington Square
City	San Jose
State/Province	CA
Postal Code	95192-0209
Country	US
Registration Date	
Last Updated	2011-09-24
RESTful Link	<a href="http://whois.arin.net/rest/org/SJSU">http://whois.arin.net/rest/org/SJSU</a>

Name	University of California, Riverside
Handle	UNIVER-211
Street	University of California, Riverside Computing and Communications
City	Riverside
State/Province	CA
Postal Code	92521
Country	US
Registration Date	1987-07-08
Last Updated	2010-05-13
RESTful Link	<a href="http://whois.arin.net/rest/org/UNIVER-211">http://whois.arin.net/rest/org/UNIVER-211</a>

Number	10680
Name	IUPUI
Handle	AS10680
Organization	Indiana University-Purdue University at Indianapolis (IUUAI)
Registration Date	1997-10-22
Last Updated	1997-10-22
RESTful Link	<a href="http://whois.arin.net/rest/asn/AS10680">http://whois.arin.net/rest/asn/AS10680</a>

Table 2.1 Tables with Whois Information, generated using ARIN's Whois Service (ARIN, 2012)

Some organizations offer the IP tracking services using the Whois' IP blocks. This tracking service extracts the Whois Service's metadata related to geographical location (in terms of Postal Code, City, State/Province and Country). With this information, a company like IPAddress.org (<http://www.ip-address.org>) offers a complementary service of IP tracking with a georeference-based system. In the United States, the location on the map of a server (generally an ISP) based on the Whois IP block is very accurate.

#### 2.1.4. Are 3-DES and RSA secure algorithms?

In terms of security, encryption algorithms play a key role in protecting information. Using a top-view perspective in the field of cryptography, the general schema shows a hierarchical structure in such a field as demonstrated in the following figure:

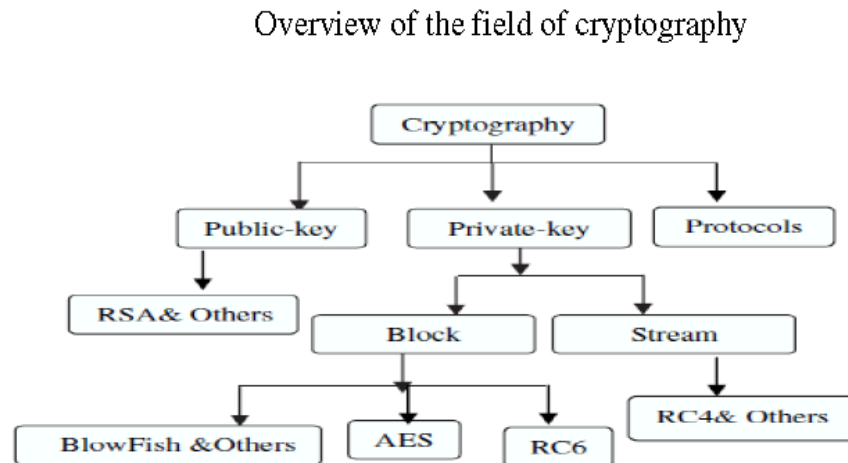


Figure 2.2 Overview of the field of cryptography

Source: (Genesan & Selvakumar, 2011: 178).

In accordance with the above figure, some encryption algorithms use public keys, as in the case of RSA, and others use private keys based on block cyphers, as in the case of 3DES. As an example, the following table summarizes the main features of 3DES and RSA:

Criteria	3DES	RSA
Classification	Private-key cryptography	Public-key cryptography
Key length	168 bits (3 times 56-bit DES key)	It depends on the needs (1024-bit is recommended)
Cypher type	Symmetric block cypher	Asymmetric system
Developed	1978	1978

Table 2.2 Comparison between 3DES and RSA

Source: (Hamdan et al, 2010: 155) and (EMC Corp. 2012).



Since their inception, both algorithms have been widely used in cryptography through software. However, in spite of their strengths, both algorithms are not completely indestructible, but they are still in use due to their enhancements. For instance, RSA enhancement is immune to attacks under the following situations: Known plain-text attack, Small public exponent, Johan Hasted and Don Coppersmith attack, Common Modulus Attack, Timing Attack and Adaptive chosen cypher text attacks (Kakish, 2011:245). In the same way, experimental tests have shown that 3DES can be improved by using key-updating with each packet (Genesan & Selvakumar, 2011: 193).

The development of new enhanced algorithms in matters of security is an ongoing quest. Fortunately, there are many interested parties trying to reach this goal, how to secure information against criminals. It follows then, that cryptography is a passionately researched field where security is tested in authentic environments using valuable information. This shows that we still trust in those algorithms with their upgrades. In the end, the final responsibility lies in our own hands, in maintaining good practices to avoid security failures.

#### **2.1.5. ATM and Gigabit Ethernet**

Initially known as cell relay, the Asynchronous Transfer Mode (ATM) was considered an international standard for high-speed connections, especially in telephony (Goralski, 2009: 439). This technology used time-division multiplexing through fixed-size cells over virtual circuits on the network (that is why it was named cell relay at its inception).

Within its functionality, ATM offers services between 155Mbps and 622 Mbps and higher; it is based on the approach of Broadband Integrated Service Digital Network (B-ISDN). Since its

creation, ATM was conceived as a communication technology for integrating end-to-end transport of voice, video and data (Kurose & Ross, 2010: 280).

The ATM architecture is formed by 3 layers in end systems and 2 layers in ATM switches. The following figure shows the ATM layer stack in a simple end-to-end connection.

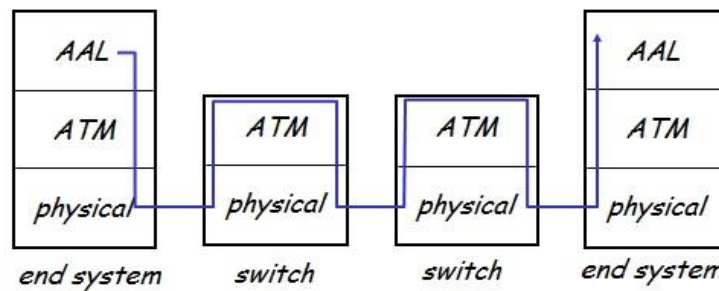


Figure 2.3 ATM layer stack

Source: (Kurose & Ross, 2010: Digital material on slides – Chapter 5)

Briefly, AAL is the adaptation layer that works with data segmentation and reassembly in a similar way to the Internet transport layer; this level has the task of adapting IP or native ATM applications to fit with lower layers. The ATM layer is the network layer that works with cell switching and routing tasks. Finally the physical layer represents the hardware devices with the implementation of the ATM standard. In spite of the fact that ATM was born as a stand-alone network standard, it is used today as switched link layer, by connecting IP-based networks in backbone environments.

ATM has several services in AAL such as AAL1 (with Constant Bit Rate for circuit emulation), AAL2 (with Variable Bit Rate for video-based applications), and AAL5 (with Available Bit Rate for data related with IP datagrams). The advantages of the ATM approach are related to “QoS performance guarantee for connection mapped to VC (bandwidth, delay, delay

jitter). It is notorious for its high performance in the matter of media deployment, and perhaps its major disadvantage is in terms of the implementation costs” (Kurose & Ross, 2010: Digital material on slides – Chapter 5).

On the other hand, Gigabit Ethernet is a proposal that works with high-speed connections via optical platforms. Its predecessor model, the Ethernet, was extremely popular in the networks assembly for several reasons: Ethernet was the first implementation for High-Speed LANs and it was cheaper than token ring, FDDI and ATM. In spite of the fact that the Ethernet has its origins in mid 1970’s, today it is still in use (Kurose & Ross, 2010: 476).

Gigabit Ethernet inherited some aspects of the Ethernet experience, such as its access method (CSMA/CD). In this vein, Gigabit Ethernet is 100% fully-compliant with interconnection of “traditional” Ethernet/Fast Ethernet platforms ((Kurose & Ross, 2010: 484). Gigabit Ethernet is defined by the standard IEEE 802.3z and one of the most important advantages is its scalability power with Ethernet-based infrastructure. The architectural model of Gigabit Ethernet is represented in the figure bellow:

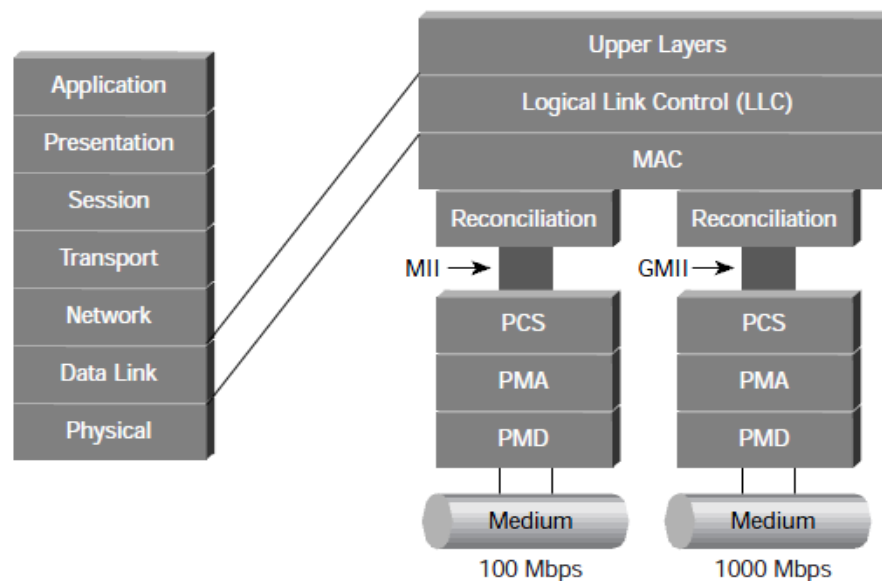


Figure 2.4 Source: (Cisco Systems, 2000: 3)

As an extension of the Ethernet proposal, Gigabit Ethernet offers high performance in media deployment via fiber channels. In conclusion, both systems (ATM and Gigabit Ethernet) enable high-speed environments; the performance in terms of media deployment (in special video) is practically the same. However, Gigabit Ethernet offers better conditions for upgrading to current systems which are based on Ethernet/Fast Ethernet platforms. In fact, it seems to be less expensive than ATM in the end.

#### **2.1.6. Connections within a Web Browser**

In client, web browsers are applications that show the content of computing solutions based on the World Wide Web service. In the first draft of HTML, the creators established its main function of “interpreting” web content within a specific application called a browser, and as such, the application has to establish a connection with the web server to get access to HTML documents (Berners-Lee & Connolly, 1993).

This primary concept in web browsing establishes the base for communication between clients and servers. The communication process is carried out by use of the Hypertext Transfer Protocol (HTTP). On the Web, HTTP is the application-layer protocol that works with messages within a generic technology environment; this means that communications are established regardless of hardware platforms, operating systems and web browsers.

A key point is that HTTP uses TCP as its transport protocol. HTTP initiates a TCP connection with the web server, and then when the TCP connection is established, the dialog between the processes of both client and server is possible, using HTTP messages (Kurose & Ross, 2010: 101). The number of TCP connections is a key factor when it comes to performance after

the web content has loaded, and web browsers have default values which manage these connections.

In technical terms, it is possible to work out the number of TCP connections for web browsers. For example, Windows Internet Explorer 9 uses Windows' Registry values in run-time. The way to figure out the maximum number of connections for the aforementioned web browser is by modifying the entry below using the *RegEdit* program:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\InternetExplorer\MAIN\FeatureControl\F  
EATURE_MAXCONNECTIONSPERSERVER
```

The key above establishes the value of the maximum number of connections per server within the environment of Microsoft's Internet Explorer. However the value is given according to the Request for Comments 2616 for the specification HTTP/1.1. As such, it states that, "A single-user client SHOULD NOT maintain more than 2 connections with any server or proxy. A proxy SHOULD use up to  $2*N$  connections to another server or proxy, where  $N$  is the number of simultaneously active users". This statement is given so as to avoid congestion on the Web (Network Working Group, 1997).

There are two key features when it comes to altering the maximum connections in a web browser: Performance and Standard Compliant. Performance can be increased by altering the TCP connection parameters in contexts with nonpersistent connections. However, it is important to note that the parameters were established in accordance with standards, so the manipulation of such values could be risky for the web server's work.

### **2.1.7. DNS: Untangling the Web for End Users**

Hosts on the Internet are identified by numbers; specifically through IP addresses of 4 bytes (or 6 in their latest specifications). Taking into account that the World Wide Web is a service that uses the Internet as a platform, web servers are reached by following the regulations of the Hypertext Transfer Protocol (HTTP). HTTP requests messages that contain a special line called a “host” which is provided in a user-friendly format (Kurose & Ross, 2010: 105).

End users type URLs which are easy to remember; these addresses have the form “hostname-directory-resource”, where the “hostname” part is mandatory. A hostname is composed of mnemonics in order to facilitate people’s usage of it; however, those easy-to-read strings must finally be translated into IP addresses. This is the work of the DNS.

The Internet’s Domain Name System (DNS) is the translation mechanism which changes hostnames into IP addresses and vice versa. In technological terms, DNS works within the level of application protocols along with HTTP, FTP and SMTP, among others. DNS technology is based on a hierarchical structure of servers with distributed databases working much like a huge directory service (Kurose & Ross, 2010: 133).

Without DNS, people would have to deal with IP addresses directly, and this would be both tedious and difficult work. Fortunately, DNS is globally regulated by the Internet Corporation for Assigned Names and Numbers. The corporation, in its affirmation of commitments, stated that DNS security is a priority in sustaining the system over the years (ICANN, 2011), but as with all technology-based systems, DNS also has its vulnerabilities.

There are several forms of attacks to DNS servers, but perhaps the most common is Distributed Denial of Service (DDoS). An extreme attack created by a massive botnet could queue up the requests on a DNS server, causing saturation and ultimately, failure of the service. If the DNS service fails due to a DDoS attack, websites are not visible, IP resolution is not possible, e-mail systems face undeliverable messages, etc., in short: chaos on the Web.

As with all services, DNS has its own configuration parameters. With this in place, a DNS Server should be protected against possible attacks. Protection is achieved by implementing the actions listed below (Ruiz et al., 2009: 106):

1. Isolate resources: Dedicate a server purely to DNS, avoid remote access to this server, and minimize software functioning on it.
2. Redundancy: Use a secondary (slave) server with a different ISP.
3. System Update: Use the latest updates in order to secure the server, in particular the DNS service.
4. Access Control: Restrict the zone transferences, use signed transactions and avoid recursive queries.
5. Authentication: Strong policies for authentication and authorization.
6. DNS secure configuration: Never show its version or good practices in monitoring logs, etc.

### 2.1.8. DNS Hacking

DNS is a service in the application-layer responsible for the proper translation of hostnames, which are easily understood by people, into IP addresses, which are managed by machines, and vice versa (Kurose & Ross, 2010: 132).

As this is an enormous responsibility, the normal “order” that people experience using the Internet, when it comes to the names of sites, is affected if the DNS-based infrastructure collapses. There are several ways to attack DNS; for example by using a masquerade or by “spoofing” the address of a legitimate host. In this scenario, if an attacker is able to alter the DNS registers in a host, this person can also redirect all of the clients’ requests to a customized private server where phishing and identity theft can be committed, leaving the way open for several Denial-of-Service or Man-in-the-Middle attacks (Piscitello, 2006: 44).

With this scenario in mind, if DNS were to be hacked, 3 negative consequences would ensue:

1. **Pharming:** If a DNS cache is poisoned, it is possible that a domain would be hijacked during a period of time (Radcliff, 2005: 39). One of the most devastating consequences of a domain hijack is the reputation of the company which is attacked; its image in the eyes of the world can take a negative turn, and in the business world it could result in high economic losses.
2. **Denial-of-Service:** One of the major cases involving Denial-of-Service happened in China in 2009, when a massive attack to China’s registrar DNS server shut down Internet access



in several parts of the country. This attack fits into the parameters of Denial-of-Service because it unleashed a massive consumption of online video websites beyond control (CWHK, 2009: 76). In this DNS attack, compromised servers stopped working because of the overload of simultaneous clients' requests, the systems just collapsed.

3. Identity Theft: As a consequence of Pharming, a domain can be hijacked; and in that case, the attacker's authentication interfaces look like the real ones. An unwary user may give his/her credentials (username and password) without being aware of the risk and their identity could be stolen.

Finally, these problems regarding information exchange have enormous impact on our lives. However, the outlook is not so bleak. In fact, there are efforts being made to avoid DNS hacking. For example DNSSEC and TRUE (Trusted Response and Universal Enforcement) continually confront the aforementioned hacks, and with good results. Particularly the latter which even has the infrastructure to globally report malicious sites as a scalable blacklist, working in complement with DNSSEC (Marsan, 2009: 21).

### **2.1.9. Exploring Network Management**

Network Management is a complex job due to the sheer number of elements which are present in a network, including hardware, network adapters, software, communication infrastructure such as cables (if applicable), connectors, access points, patch cords, and so on. Every single element could fail at any given moment, either due to physical incidents (electric

overload, corrosive substances, humidity, and extreme heat, among others), misconfiguration, security vulnerabilities, and even poor utilization (Kurose & Ross: 2010: 772).

Due to its complexity, the most common tasks have already been identified, such as: detecting failure of a NIC, host monitoring, traffic monitoring, intrusion detection, etc. However, in essence, practically everything related to network usage, account management, and security is part of Network Management.

Perhaps another field that could come under the cadre of Network Management is surveillance. Within the ISO's network management model, there is a special area related to Accounting Management. In this area, it is possible to control the access and usage of people who work within a network environment. Typically, such a control is done using logs and session management, according to specific roles and privileges (Kurose & Ross, 2010: 775).

In addition to Accounting Management, real-time surveillance in workplaces can be a part of network management, in spite of the privacy issues this entails. Video and audio monitoring in workplaces could ensure a high level of security against theft. An important implication here however, is that of the "big brother" phenomenon, this surveillance could be considered a violation of privacy.

On the other hand, with the development of wireless sensors in control systems, surveillance activity is becoming more of a monitoring one. That is to say, sensors working alongside audio/video in real-time, provide full scope for capturing data in a given context. Such sensors and devices are part of a network in which management is required in order to avoid failures and prevent accidents.

Wireless Sensor Networks can be incorporated as part of an Internet-based network in order to access wider areas. Previously, WSN worked only in small areas, but today such networks can be integrated into the Internet. In this vein, there are several “new” elements, beyond hosts, servers and routers. This integration can be achieved using a Web-based REST approach. Web services are the key in combining WSN with web-based applications, to give a fast response in real-time. This is carried out using virtual links between physical nodes using the Internet Protocol Stack (Coulson et al, 2012).

As always, security is a key point of attention for developers. Wireless Sensor Networks integrated into Internet-based networks face their own issues in this regard. Wireless Sensor Networks are victim to some Internet-like attacks, for instance: eavesdropping, Denial-of-Service, and message tampering, among others (Sadeghi et al, 2012). In such a scenarios, Network Management has the responsibility to confront these threats to ensure secure data transportation. This is yet another task for network administrators.

#### **2.1.10. Functions of Network Adapters and Nodes**

Within the Internet protocol stack, the majority of the link layer is made up of network adapters (Kurose & Ross, 2010: 446). The logic that pertains to network adapters is written at an electronic level. A network adapter has two main parts: the core interface and the network interface. The former part is related to the usage of software mechanisms which communicate with the upper layers (this part is known as the controller); and the latter part is related to physical media, which means that the network interface within the hardware adapter is responsible for

sending and receiving physical signals (Ahmed et al, 2011: 265). The distribution of the Internet protocol stack within a host is shown in the figure below.

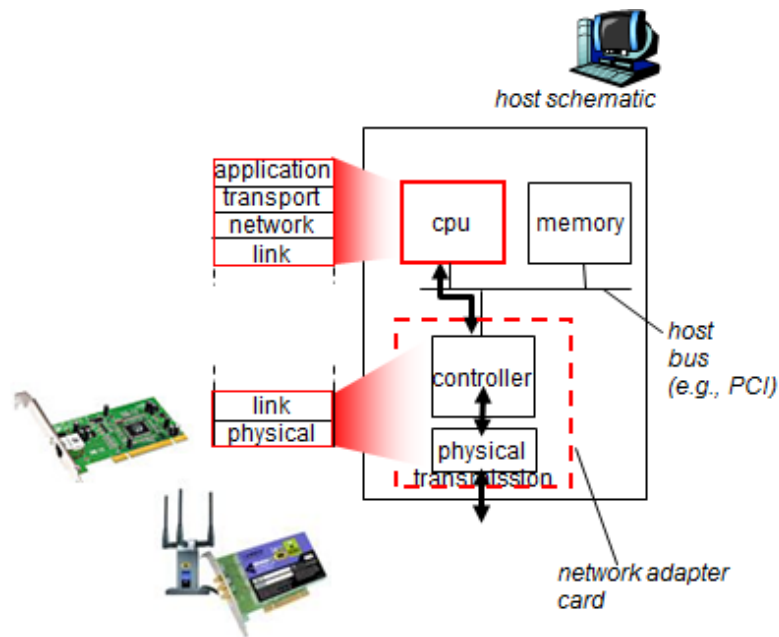


Figure 2.5 Distribution of the Internet protocol stack

Source: (Kurose & Ross, 2010: 446)

According to the previous figure, the element responsible for transmitting/receiving information through computer networks is the network adapter card. As such, a set of functions related to the physical transmission are written at an electronic level in firmware. Obviously each network adapter has its own firmware depending on the manufacturer; however, some adapters can be configured by altering the value of parameters available in EPROMs.

Functions related to the network adapter include framing, link access, flow control, error detection, and error correction. On this device, the functions related to I/O tasks are performed by the adapter's circuitry; however, the controller assembles the datagram that comes from the network layer into link-layer addressing information (Kurose and Ross, 2010: 447).

The main advantage of having implemented functions within network adapters is the fast response time. The actions performed by network adapters in terms of transmission and reception of physical signals do not depend on the node's CPU. In fact, they are directly controlled by the adapter's circuitry. Nonetheless, taking into account that network adapters are important elements in implementing the Internet protocol stack, these hardware devices interact with the upper layers via software interfaces called controllers.

In the first instance, operating systems must manage the correct resources by use of the network adapter. In computer architecture, ports are reserved for facilitating communication with I/O devices, and interrupts are the low-level mechanisms which send/receive messages in real time (within the scale of milliseconds). In the second instance, common software applications (the application layer) in the host, use the network adapter in an indirect way; that is to say, applications create a socket-based scenario where APIs establish communication with the operating system in order to command a hardware adapter. Evidently more resources are needed when information is passing through other levels such as the network and transport layers.

Nowadays, network adapters work in this way, with the implementation of the link layer in the hardware device, and some parts in the host's software. The heavy work is done physically (by the network adapter), not only in terms of transmission/reception, but also in terms of error detection/correction. Taking all this into account, the role of the host's CPU is delegated to that of communication with the upper layers in the Internet protocol stack. A possible disadvantage could arguably be related to the dependency of the context into which network adapters are installed, the differences between operating systems and application development models according to those operating systems.

### **2.1.11 NAT and its impact on IPSec**

Network Address Translation (NAT) is a computing mechanism which converts IP addresses and ports, in order to communicate real world-wide hosts with dynamically-assigned hosts within a private local area network. This is two-way communication. Understandably, an IP address translator is required, and the translation process is done within a NAT router (Kurose & Ross, 2010: 359-360).

Over time, NAT has been used in different contexts such as corporate LANs, home networks, Internet cafes, video game rooms, etc. These different settings all have one thing in common: dynamic identification of local hosts by DHCP. In other words, local machines within a private LAN have access to their resources according to the local network's restrictions; outside resources on the Internet are available via an entry point (NAT router) with translation capabilities in the matter of IP addressing.

The usage of NAT is accepted in RFC1631 with some limitations according to the IPv4 capabilities; in fact, NAT was considered a short-term solution in overcoming range-based limitations in IPv4 (Phifer, 2000). As a matter of fact, NAT is not 'popular in some radical circles in IETF, due to issues related to some NAT-sensitive protocols. This is true in the case of IPSec.

IPSec is a protocol for secure IP-based packing of information. IPSec interacts with the network layer, and also establishes direct contact with the application layer (Tiller, 2001: 38). The following figure illustrates the location of IPSec in front of the OSI stack:

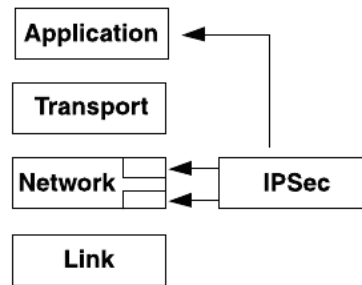


Figure 2.6 IPsec in the TCP Stack (Tiller, 2001: 38)

To work properly, IPsec needs to preserve the IP addresses contained within the packets. A problem presents itself when NAT changes the IP addresses in its translations; in those moments, “When end-to-end IPsec authentication is used, the IPsec integrity check will always fail” (Tiller, 2001: 98).

Over the passage of the time, alternatives which also bypass the range-based limitations of IPv4 have been developed, such as IPv6. IPv6 can clearly handle the imminent exhaustion of the IPv4 addressing space, due to its inner structure. IPv4 in general terms has a defined life expectancy due to its addressing space, and as such, NAT helped IPv4 to extend its life in the arena of private networks. IPv6 is a sustainable option because it offers plentiful addresses, routing scalability and easier end-to-end support (Loshin, 2004: 23).

With a 128-bit format for addresses, IPv6 has the ability to define  $2^{128}$  (approximately  $3.4 \times 10^{38}$ ) addresses. With this in mind, NAT-based mechanisms could soon be superfluous to requirements because IPv6 can deal with real public IP addresses at all levels at which a device requires a globally unique identification. Unfortunately, nowadays the most popular technology is still based on IPv4. The development of IPv6 has not been slow, but its implementation has been delayed due to changes in IT infrastructure. In short, it is only a matter of time.

### 2.1.12. P2P Networking

The main purpose of P2P technology is to share resources, such as content, storage, and processing capability; in this vein, P2P technology is able to transmit portions of files, between two points (peers), and lastly reconstructs the file at its final destination (Kurose & Ross, 2010: 147).

File sharing is the activity in which this technology, that of P2P, has most frequently been applied. Content distribution applications transform personal computers into active elements within a distributed storage medium in coordination with hundreds, or even thousands of others (Androutsellis-Theotokis & Spinellis, 2004: 335). P2P has been popular since its inception in the 90s, in spite of the fact that the idea which spawned P2P is older than its implementation. (Moody, 2002).

Several computing systems are based on P2P technology allowing them to share files. Five examples are detailed below:

1. **The Ocean Store Project:** This project was conceived by Professor John D. Kubiawicz with the cooperation of graduate and undergraduate students, and other contributors from the University of California at Berkeley. This project is an entirely academic-research proposal, so the content associated with the usage of Ocean Store could be anything, stored by even  $10^{10}$  users, each with at least 10,000 files (Kubiawicz, 2011).
2. **KaZaA:** KaZaA is a well-known company in the world of music downloads. This company provides several services, but the most frequently used one, is related to the P2P file sharing service through FastTrack protocol. Today, KaZaA works under a license, with official paid subscriptions, and is only available in United States. This company uses application



in client machines as a Media Desktop. The file types are music, in mp3 format (KaZaA, 2010).

3. **Voddler Group AB:** This Company offers a streaming solution for renting movies online. In addition, Voddler offers users the option of renting movies to download and watch offline. The latter service works via P2P technology. The file types, when it comes to renting movies for download include mp4 format, among others, and a paid subscription is required. These downloads can reach up to Gigabyte size (Voddler Group AB, 2012).
4. **Oversi Networks Ltd:** This company has created a computing solution which relates to Internet Videos. Oversi uses Over Cache P2P on the delivery platform of media content, by using peer-to-peer architectures. This technology has an important feature which is temporary storage in the network's backbones. In this way, the traffic on the network is reduced (Oversi Networks Ltd., 2010). File-sharing is then established for general use.
5. **Ignite Technologies:** Its mission is stated as the identification and use of the most efficient ways to deliver content on demand. One product from its catalog is called Live Streaming, created to deploy media content via streaming which is based on peer-to-peer technology (Ignite Technologies, 2012).

In conclusion, these computing solutions were created with the main purpose of deploying and delivering content, especially media content. The main concern around the use of peer-to-peer architectures is related to negative experiences in the matter of infringements of copyright laws. Unfortunately, P2P technology has been associated with piracy due to several cases of violation of copyright terms.

### 2.1.13. SIP basics

SIP stands for Session Initiation Protocol, which is defined in RFC 3261 and RFC 5411 (Kurose and Ross, 2010: 640). In spite of the fact that the implementation of SIP has taken place fairly recently, its development as a project began in 1996 and it was first released in 1997, by IETF Multi-party Multimedia Session Control Working Group – MMUSIC (Johnston, 2009: 20). At that time, voice communication was based on public switched telephone networks, using protocols such as SS7 and H.323.

An interesting point about SIP's development is that this protocol is not an entirely new one, it actually reuses common elements which were already available in previous technologies. For instance: SIP reuses status codes from HTTP, SRV records from DNS, and formats in MIME extensions via SMTP (Cisco Systems, 2009). In short, SIP is similar to AJAX technology in the sense that neither are brand new proposals. In the case of AJAX, the framework was created based on existing JavaScript and XML principals.

In 2001, SIP was regulated by RFC 3261. In technical terms, SIP is considered an application-layer protocol; SIP works over UDP, within the Internet Multimedia Protocol Stack and it also functions in other layers (Johnston, 2009: 3). The following figure shows SIP's location within the stack.

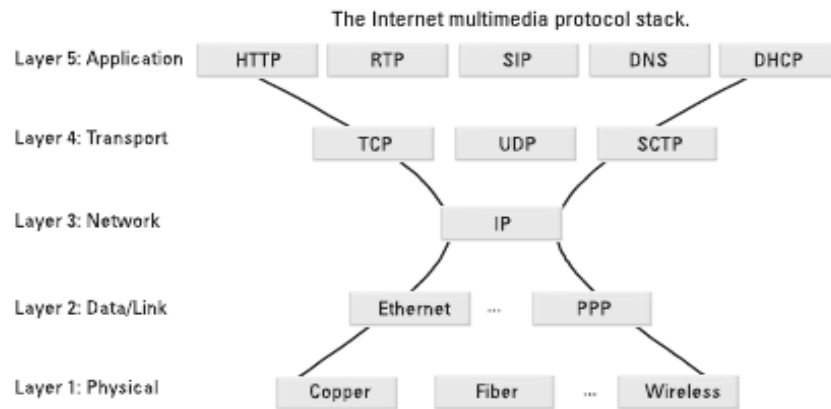


Figure 2.7 SIP's location on the stack

Source: (Johnston, 2009: 3)

In essence, SIP has four basic tasks: to locate users and to translate addresses (SIP to IP and vice versa), to negotiate capabilities with the session's participants, to maintain even session parameters when a call is live, and to control the beginning and the end of a session (Cisco Systems, 2009). In order to understand how SIP works, it is important to take into account three elements of local domain calls: user agents (cellphones, IP phones, computers, etc.), registrar servers (databases where the information of agents' location within a domain is stored), and proxy servers (for authentication and the initial maintenance of the call's channel, then the call can be held by the users directly). For other scenarios outside a local domain call (the Internet for instance), redirect servers are required in order to reach the final destination via WANs. The following is a diagram illustrating this technology:

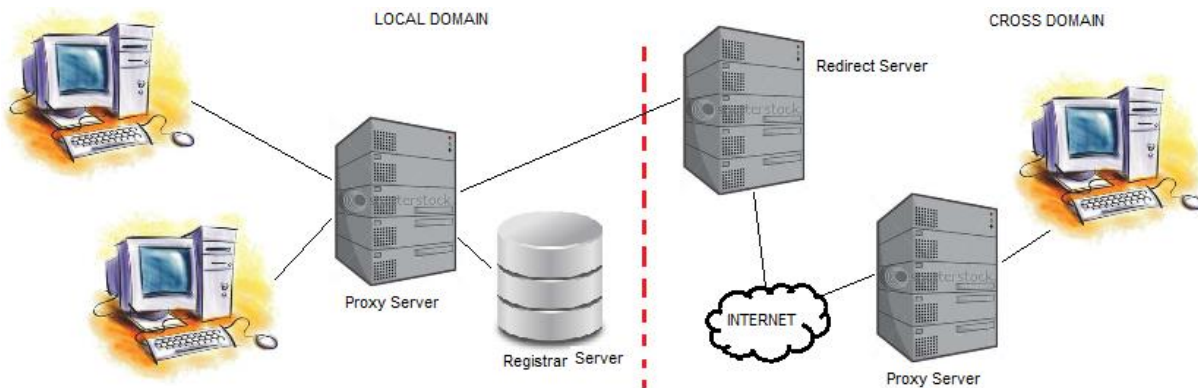


Figure 2.8 WAN sample

#### 2.1.14. TCP Hijacking

The Transmission Control Protocol (TCP) is a transport-layer, connection-oriented protocol that allows the transfer of reliable data through computer networks. This protocol is provided with congestion control mechanisms, retransmission policies and error detection, among other features (Kurose & Ross, 2010: 242). In its basic form, TCP provides reliable methods for data transport; however, TCP is not entirely secure, and as such security must be provided by protocol extensions such as HTTPS over Secure Socket Layers.

In spite of its achievements in terms of security, TCP connections can still be hacked under certain circumstances. Evidently this represents a security threat for Internet-based information systems. Today, there are several ways to attack information systems on the Internet, and in such attacks, information integrity can be compromised by partially (or totally) altering its inner data. One of these techniques is called TCP Hijacking.

According to RFC 793, TCP functional environment is commanded by flags according to the TCP header format; this means that a TCP connection can change its status by using flags such as SYN, ACK, RST and FIN (DARPA, 1981: 21).

The altering of flags in messages can be an indicator that there is a problem in a TCP connection, such as in the case of a Denial-of-Service (DoS) attacks (Berghel, 2002: 23). For instance:

- A packet with a RST flag will drop the TCP connection. The connection is simply lost between client and server.
- A packet with a SYN flag will reset the TCP connection. As in the previous case, the connection is lost when the host responds to such a packet.

On the other hand, the data within TCP packets can also be altered, in order to corrupt the end user's message, or even to inject malicious commands into a sequence. This injection of characters or data in packets is an example of a *Man-in-the-middle* attack, using techniques known as Nmap, Amap and Scanrand port scanning in order to "poison" ARP registers and TCP segments. (Gregg & Watkins, 2006: 165).

Unfortunately, the TCP hijacking technique is quite difficult to detect if it is used within a DoS attack. In order to avoid such a threat, some countermeasures can be implemented as detailed in the list below:

- Implement static ARP tables.
- Use strong encryption on protocols and applications for end users.
- Configure smaller windows and random source ports.
- Use proper filters for ingress and egress.

Unfortunately, not everybody can do these things. These require a profound knowledge of operating systems and communications, and possibly significant experience in IT and networking, especially in matters of security.

#### **2.1.15. Wi-Fi Networking: A case at home**

These days, having more than one computer per house is common, and so when different people want to access the Internet from different computers, a home network is required. Within a house it is possible to create a wire network, if the computers are desktops. However, when there are mobile devices which are part of the computer inventory, such as laptops, tablets, PDAs and smart phones to name but a few, wireless networks come on the scene.

The standard that defines wireless technology is IEEE 802.11. This standard provides all the specifications on the implementation of communication solutions based on wireless technology. One of the main issues with Wireless technology is related to the coverage of physical areas. As the standard says: “well-defined coverage areas simply do not exist; propagation characteristics are dynamic and unpredictable” (IEEE, 2007: 27). The location of the access point is a fairly important factor when trying to give the entire house access to Internet signal.

Within a wireless network, the architecture establishes a fundamental element, called a base service set (BSS) which contains the clients (end devices), and a central base station known as the access point (Kurose & Ross, 2010: 537). In this structure, end devices get Internet access through wireless network adapters (wireless cards); the signal is emitted from the access point, from which a switch or a router connected to an ISP or another network has access to the Internet.

At home, access to the Internet is provided by a specialized company (in most cases, such companies belong to the communications and telephony sectors). When a provider has been chosen, the next step is to have the access points (at least one) in a strategic place that will give internet access to the entire building.

To make setting up a wireless network at home easier, the following infrastructure has been suggested:

Item	Quantity	Unit Price (US\$)	Total (US\$)
<b>ADSL Wi-Fi Router/Modem</b> (Ambit uBEE U10C037 Wireless Doc 2.0 Cable Modem + Router)	1	119	119
<b>Wi-Fi Network Adapters</b> (Linksys Refurbished WUSB54GC Compact Wireless-G USB Adapter)	3	17	34
<b>Total</b>			153

Table 2.3 WiFi Comparison

With the aforementioned technology it is necessary to contract an ISP to get Internet access. Normally a monthly payment plan is required, and the prices vary according to the bandwidth. Taking into account that it is a home connection, 4Mb is sufficient in most cases. The contract may range from US\$10 to US\$30 monthly.

## **2.2. WORKSHOPS**

### **2.2.1. Hand-In Assignment 1**

#### **1. Packet Switching and Statistical Multiplexing**

Packet switching is a common technique used in the transmission of information via the Internet. Instead of creating a fixed circuit between two (or more) end devices, packet switching distributes messages between nodes along the path to the final destination (Kurose & Ross, 2010: 30).

Packet switching manages link use according to demand; this means that the traffic at routing points has to be distributed by following rules in order to avoid chaos, multiplexing techniques are required. More specifically, the best way is through statistical multiplexing.

In tech domains, statistical multiplexing has been an important element in packet switching since the 1960s; however, it was taken up in a major way in the 1990s, when physical media allowed for transmission levels of around 10Gbps at the core of the Internet. Since its inception, statistical multiplexing has been designed to divide the bandwidth into a number of channels (Chandra, 2003: 2420).

In contrast, TDM (Time-Division Multiplexing) is a technique in which time is divided into slots of fixed length, in a number of channels. Within Circuit-Switched Networks, the number of channels established in a fixed pre-designed path, is constant (Ojesanmi, 2009: 74).



## 2. Botnet and DDoS

There are several types of attack on the Internet, such as Increased Access, Disclosure of Information, Corruption of Information, Theft of Resources, and Denial of Service, among others. In particular, in the case of Denial of Service Attacks, there is an intentional degradation of the normal service of a specific company (the victim) through the abuse of requests by legitimate users.

Denial of Service is commonly associated with the client-server model. In contrast, DDoS (Distributed Denial-of-Service) has the same *modus operandi* as Denial of Service Attacks, but is perhaps more effective and powerful, due to the use of large and complex botnets working within a distributed environment (Arora, Kumar, & Sachdeva, 2011: 878).

In order to define a botnet, it is first necessary to outline a general picture of a bot (short form of robot). A bot is a malware installed in an end user's device; this malware can "spy" and discover sensitive information on the client's machine, such as SSN, keystrokes, passwords and so on. The main objective of a bot is to serve a central point (sometimes called a botmaster) without the end user realizing what is happening. In a global arena, thousands, or even millions, of bots can be managed remotely by the attacker, and a victim (a website) is inundated with legitimate requests made by the group of clients in which the bots are active. (Greengard, 2012).

## 3. Wireshark Lab

Using this tool, the IP address of the following was extracted, <http://www.uol.ohecampus.com>. The steps that were required to do this are outlined below:

- Enter the packet sniffer.

- Start the capture mode with the specific options which have been previously edited.
- Go to the default web browser, and type the URL.
- Stop the capture.

In the textbox for filters, it is possible to show all packets, or specific packets according to regular expressions. We used *http.response == 1* and *http.request == 1* in order to get the specific information. All screenshots are provided:

No.	Time	Source	Destination	Protocol	Length	Info
4	0.208763	10.0.2.15	74.116.156.44	HTTP	271	GET / HTTP/1.1
31	0.626624	10.0.2.15	74.116.156.44	HTTP	355	GET /templates/skins,
46	0.639154	10.0.2.15	74.116.156.44	HTTP	350	GET /templates/skins,
48	0.639594	10.0.2.15	74.116.156.44	HTTP	432	GET /templates/skins,
49	0.639818	10.0.2.15	74.116.156.44	HTTP	404	GET /templates/skins,
51	0.640217	10.0.2.15	74.116.156.44	HTTP	395	GET /templates/skins,
56	0.642203	10.0.2.15	74.116.156.44	HTTP	408	GET /templates/skins,
60	0.721471	10.0.2.15	74.116.156.44	HTTP	409	GET /templates/skins,
62	0.732250	74.116.156.44	10.0.2.15	HTTP	1308	HTTP/1.1 200 OK (te)
76	0.737646	74.116.156.44	10.0.2.15	HTTP	1273	HTTP/1.1 200 OK (app)
78	0.738750	74.116.156.44	10.0.2.15	HTTP	785	HTTP/1.1 200 OK (app)
85	0.749302	10.0.2.15	74.116.156.44	HTTP	393	GET /templates/skins,
87	0.750362	10.0.2.15	74.116.156.44	HTTP	354	GET /templates/skins,
88	0.750612	10.0.2.15	74.116.156.44	HTTP	413	GET /templates/skins,
98	0.834820	74.116.156.44	10.0.2.15	HTTP	60	HTTP/1.1 200 OK (te)
103	0.849335	74.116.156.44	10.0.2.15	HTTP	1153	HTTP/1.1 200 OK (app)

GET / HTTP/1.1\r\n
Accept: */*\r\n
Accept-Language: en-us\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; windows NT 5.1; Trident/4.0)\r\n
Accept-Encoding: gzip, deflate\r\n
Host: www.uol.ohcampus.com\r\n
Connection: Keep-Alive\r\n
\r\n

Figure 2.9 HTTP analysis

According to this figure, the IP Address of [www.uol.ohcampus](http://www.uol.ohcampus.com) is **74.116.156.44**

The next figures illustrate how to filter the information, the former is for HTTP GET packets and the latter is for HTTP OK packets.

AMD PCNET Family Ethernet Adapter (Microsoft's Packet Scheduler) [Wireshark 1.6.5 (SVN Rev 40429 from /trunk-1.6)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: `http.request == 1` Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
4	0.208763	10.0.2.15	74.116.156.44	HTTP	271	GET / HTTP/1.1
31	0.626624	10.0.2.15	74.116.156.44	HTTP	355	GET /templates/skins/Aque-DropDown/d
46	0.639154	10.0.2.15	74.116.156.44	HTTP	350	GET /templates/skins/Aque-DropDown/d
48	0.639594	10.0.2.15	74.116.156.44	HTTP	432	GET /templates/skins/Aque-DropDown/f
49	0.639818	10.0.2.15	74.116.156.44	HTTP	404	GET /templates/skins/Aque-DropDown/f
51	0.640217	10.0.2.15	74.116.156.44	HTTP	395	GET /templates/skins/splash09/writes
56	0.642203	10.0.2.15	74.116.156.44	HTTP	408	GET /templates/skins/Aque-DropDown/f
60	0.721471	10.0.2.15	74.116.156.44	HTTP	409	GET /templates/skins/Aque-DropDown/f
85	0.749302	10.0.2.15	74.116.156.44	HTTP	393	GET /templates/skins/splash09/webco
87	0.750362	10.0.2.15	74.116.156.44	HTTP	354	GET /templates/skins/Aque-DropDown/d
88	0.750612	10.0.2.15	74.116.156.44	HTTP	413	GET /templates/skins/Aque-DropDown/f
251	1.673640	10.0.2.15	74.116.156.44	HTTP	363	GET /templates/skins/Aque-DropDown/f
252	1.673873	10.0.2.15	74.116.156.44	HTTP	364	GET /templates/skins/Aque-DropDown/f
256	1.675526	10.0.2.15	74.116.156.44	HTTP	402	GET /templates/skins/Aque-DropDown/f
258	1.676347	10.0.2.15	74.116.156.44	HTTP	404	GET /templates/skins/Aque-DropDown/f
260	1.677521	10.0.2.15	74.116.156.44	HTTP	419	GET /templates/skins/Aque-DropDown-d

Frame 60: 409 bytes on wire (3272 bits), 409 bytes captured (3272 bits)

Ethernet II, Src: CadmusCo\_66:24:58 (08:00:27:66:24:58), Dst: RealtekU\_12:35:02 (52:54:00:12:35:02)

Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 74.116.156.44 (74.116.156.44)

Transmission Control Protocol, Src Port: tgp (1223), Dst Port: http (80), Seq: 1, Ack: 1, Len: 355

Hypertext Transfer Protocol

Figure 2.10 GET packages

AMD PCNET Family Ethernet Adapter (Microsoft's Packet Scheduler) [Wireshark 1.6.5 (SVN Rev 40429 from /trunk-1.6)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: `http.response == 1` Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
62	0.732250	74.116.156.44	10.0.2.15	HTTP	1308	HTTP/1.1 200 OK (text/css)
76	0.737646	74.116.156.44	10.0.2.15	HTTP	1273	HTTP/1.1 200 OK (application/x-javascript)
78	0.738750	74.116.156.44	10.0.2.15	HTTP	785	HTTP/1.1 200 OK (application/x-javascript)
98	0.834820	74.116.156.44	10.0.2.15	HTTP	60	HTTP/1.1 200 OK (text/html)
103	0.849335	74.116.156.44	10.0.2.15	HTTP	1153	HTTP/1.1 200 OK (application/x-javascript)
139	0.970187	74.116.156.44	10.0.2.15	HTTP	812	HTTP/1.1 200 OK (text/css)
167	1.082500	74.116.156.44	10.0.2.15	HTTP	457	HTTP/1.1 200 OK (application/x-javascript)
181	1.115356	74.116.156.44	10.0.2.15	HTTP	242	HTTP/1.1 200 OK (text/css)
201	1.192183	74.116.156.44	10.0.2.15	HTTP	530	HTTP/1.1 200 OK (text/css)
204	1.217878	74.116.156.44	10.0.2.15	HTTP	1373	HTTP/1.1 200 OK (application/x-javascript)
249	1.573971	74.116.156.44	10.0.2.15	HTTP	998	HTTP/1.1 200 OK (application/x-javascript)
274	1.783602	74.116.156.44	10.0.2.15	HTTP	683	HTTP/1.1 200 OK (GIF89a)
278	1.789860	74.116.156.44	10.0.2.15	HTTP	1131	HTTP/1.1 200 OK (GIF89a)
283	1.790408	74.116.156.44	10.0.2.15	HTTP	1167	HTTP/1.1 200 OK (application/x-javascript)
286	1.793383	74.116.156.44	10.0.2.15	HTTP	443	HTTP/1.1 200 OK (GIF89a)
323	1.897298	74.116.156.44	10.0.2.15	HTTP	717	HTTP/1.1 200 OK (GIF89a)

Frame 62: 1308 bytes on wire (10464 bits), 1308 bytes captured (10464 bits)

Ethernet II, Src: RealtekU\_12:35:02 (52:54:00:12:35:02), Dst: CadmusCo\_66:24:58 (08:00:27:66:24:58)

Internet Protocol Version 4, Src: 74.116.156.44 (74.116.156.44), Dst: 10.0.2.15 (10.0.2.15)

Transmission Control Protocol, Src Port: http (80), Dst Port: aeroflight-ads (1218), Seq: 1, Ack: 302, Len: 1254

Hypertext Transfer Protocol

Line-based text data: text/css

Figure 2.11 HTTP packages

### 2.2.2. Hand-In Assignment 2

## 4. Services in a transport protocol

Transport services provide the necessary elements to establish communications through computer networks; four individual services support several different aspects, such as fidelity, performance and security, among others. The following services establish parameters depending on their main purpose:

- Reliable Data Transfer: Some packets can get lost in a computer network for different reasons (for instance: physical disconnection, excessive timeouts/delays, saturation/buffers' overflow, etc.). For some applications, all information must be saved so that no part of it can be lost; this is where reliable data transfer comes in.
- Throughput: Within a connection established in a computer network, the rate of transferred bits through the communication channel is called "throughput". However, the number can vary because it depends on the connection's bandwidth and usage, which changes according to the number of concurrent clients. As with the previous service, some applications cannot tolerate strong variations in bit rate, so the throughput service is key in bandwidth-sensitive applications.
- Timing: This service is strongly related to the previous one; the timing service establishes the parameters in terms of time, such as delays when information travels through an active communication channel. Whilst throughput is relevant to bandwidth, timing could be affected by the physical infrastructure of a computer network, including the distances between nodes.

- Security: Without doubt security is one of the greatest concerns in environments where information must travel. Privacy is a right, and the unauthorized reading (or manipulation) of private information is considered a violation of that right. It is important to note that security not only deals with confidentiality issues, but also data integrity.

Taking into account the definitions above, the main transfer protocols used in the Internet (TCP and UDP) use transport services. A brief summary of the relationship between Internet protocols and transport services is shown in the following table:

	<b>TCP</b>	<b>UDP</b>
<b>Reliable Data Transfer</b>	Provided, TCP uses a connection-oriented approach and reliable data transfer.	Not provided, UDP uses a connectionless-oriented approach.
<b>Throughput</b>	Partially provided, via a congestion-control mechanism.	Partially provided, via rate metrics.
<b>Timing</b>	Not provided.	Not provided.
<b>Security</b>	Not directly provided, SSL is necessary.	Not provided.

Table 2.4 TCP vs UDP

## 5. The Cookies' role within e-commerce

A cookie is a local text file (on client) that stores information related to the user and his/her interaction with a specific website (Lanois, 2011: 4). It is true that cookies give users a faster navigation experience; however, cookies represent a threat against privacy if they are exposed (King, 2012: 89). In spite of this, it is necessary to construct an e-commerce system where purchase records must be saved as cookies, and so the following steps are recommended:



- Firstly, the company that manages the e-commerce website should establish a clear privacy policy integrated in the portfolio of services (Lavin, 2006: 279). This declaration outlines the rules and regulations on how information will be managed.
- Secondly, when it comes to programming, web applications should be created under a client-server approach, with active content on the server-side, and dynamic behavior on the client-side. The latter function would work with the cookies being on the customer's host.
- The process is explained as follows:
  - Customer requests information from the website.
  - Website replies and it sends a cookie that would be placed on the customer's computer.
  - The user experience on the website is recorded into the local cookie.
  - The information of a local cookie is sent to the website on demand, in order to give feedback on the user's experience.
- With this in mind, it would be necessary to produce web applications with server-side active content, via scripting or compiled solutions. For such an environment, the newest versions of server-side technology could be used; this technology includes: PHP (The PHP Group), ASP.NET (Microsoft Corp.), JSP (Oracle Corp.), among others.

As an example, cookie management is simple in PHP; the code is not particularly extensive:

### Creating Cookies

Sending a signal to create a cookie with PHP is as simple as using the *setcookie()* function. This function sends all the necessary information concerning a particular cookie. All parameters are optional except for \$name.

```
<?php
    setcookie($name,$value,$expire,$path,
              $domain,$secure,$httponly);
?>
```

Table 2.5 Creating Cookies

## Reading values from cookies

There are several ways to read 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<br />";
    }
}

?>
```

Table 2.6 Reading values from cookies

## 6. The Case of a Client-Server Application

Typically, a network-based application has a server part and a client part (when the client-server approach is used). In this case, the server program must be in use before the client tries to connect to it via TCP connections. This is the “listening” phase of the communication process between client and server.

It stands to reason that a server must be online all the time; therefore, a server hardly ever shuts its services down (in some situations, the server is shutdown or restarted due to a maintenance process, reinstallation or hardware upgrade to name a few). The socket-based process of a server

program can be considered a service, and it must always be “listening” to clients’ requests. This situation is unceasing in TCP, due to its connection-oriented approach.

In contrast, a UDP connection works using a connectionless-oriented approach. In this case, the client starts its processes related to the creation of the socket address through a “hostID” and a “Port”. When this is established, the client process sends a datagram request process, looking for an available server, and when the server is reached, a request process is activated on the server side with the purpose of serving the client’s demands. This system works in connectionless environments, because the transmission of messages can start without needing a previously established connection.

## 7. Wireshark Lab

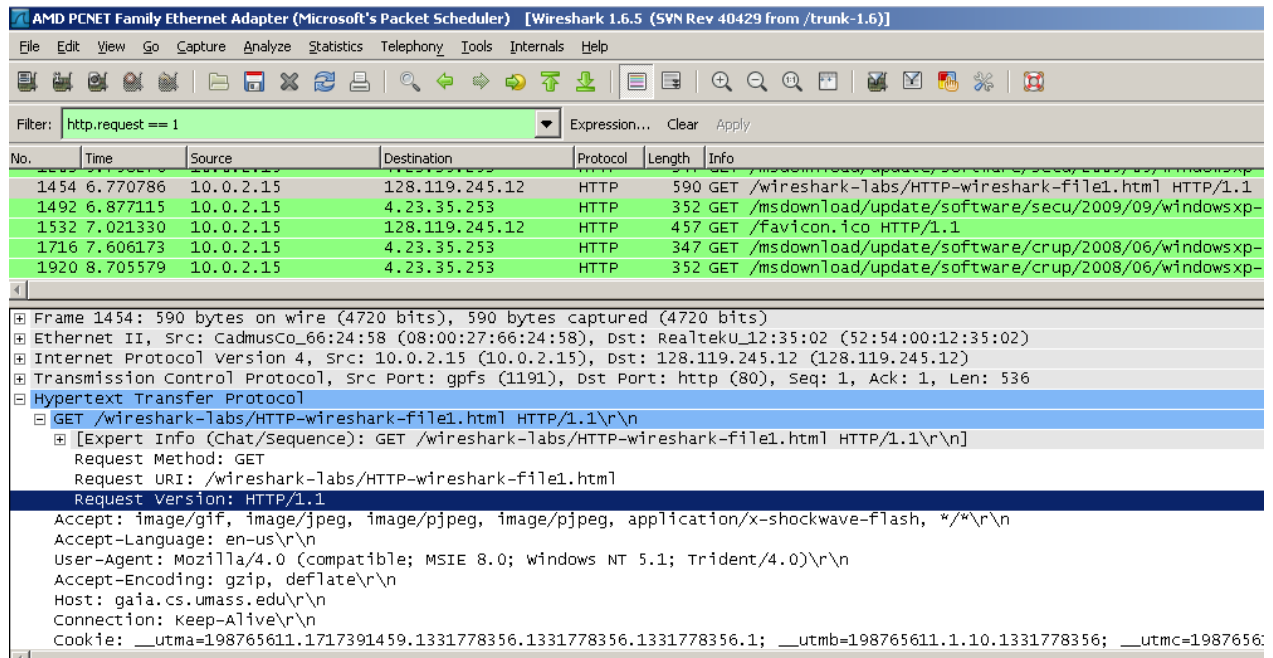
In the packet analysis of the website <http://www.umass.edu>, and DNS analysis for <http://www.premiereleague.com> (it seems to be an unofficial site about the Premiere League), the Wireshark tool obtained the following results:

HTTP server version: 1.1	IP Address (DNS): 208.87.35.100
Accept-languages: en-us	Transport Protocol for DNS: TCP
HTTP browser version: 1.1	DNS’s Name, Value, Type, TTL:
Arrival-Time: March 14, 2012 – 21:29:23 SA Pacific Standard Time	pagead2.googlesyndication.com, TypeA, Class IN, TTL=128
# of GETs sent: 13	Authoritative NS: Static-IP-1901572140.cable.net.co

Table 2.7 Wireshark results



## Analysis for [www.umass.edu](http://www.umass.edu)

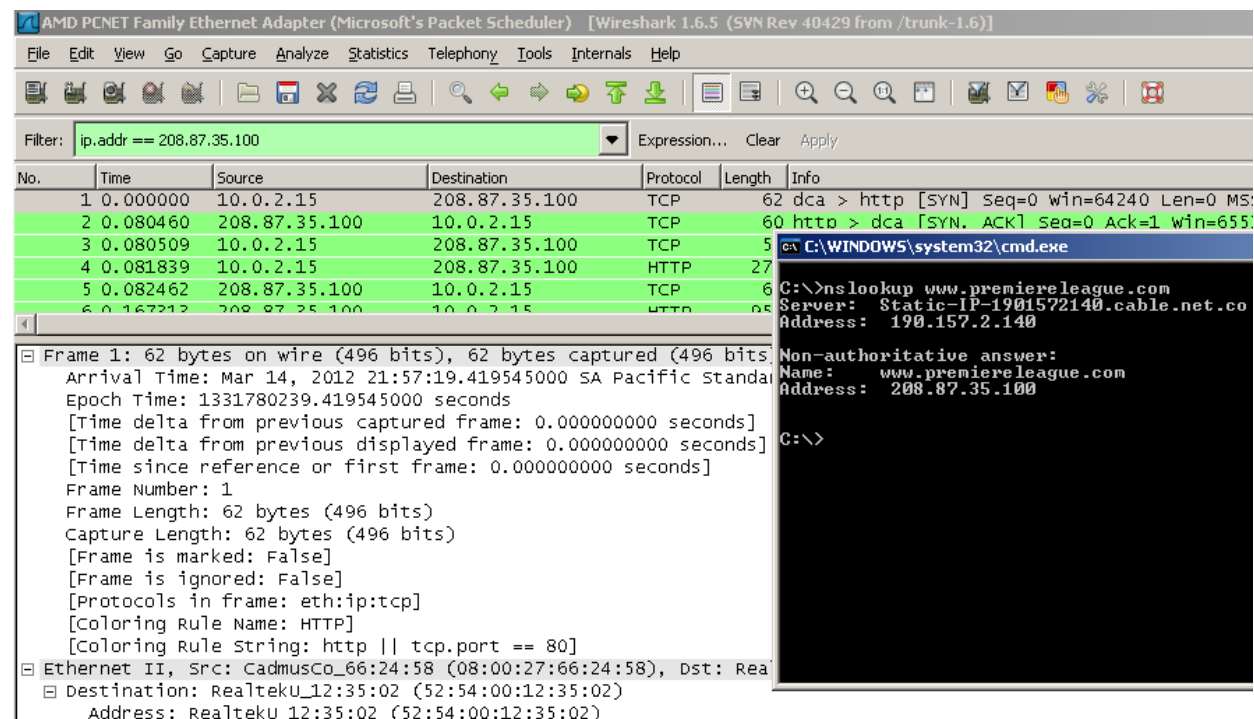


The screenshot shows a Wireshark capture of network traffic on an AMD PCNET Family Ethernet Adapter. The filter is set to 'http.request == 1'. The packet list shows several HTTP GET requests. The selected packet (No. 1454) is an HTTP GET request for '/wireshark-labs/HTTP-wireshark-file1.html' from 10.0.2.15 to 128.119.245.12. The packet details pane shows the following information:

- Frame 1454: 590 bytes on wire (4720 bits), 590 bytes captured (4720 bits)
- Ethernet II, Src: CadmusCo\_66:24:58 (08:00:27:66:24:58), Dst: RealtekU\_12:35:02 (52:54:00:12:35:02)
- Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 128.119.245.12 (128.119.245.12)
- Transmission Control Protocol, Src Port: gpfs (1191), Dst Port: http (80), Seq: 1, Ack: 1, Len: 536
- Hypertext Transfer Protocol
  - GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
  - [Expert Info (Chat/Sequence): GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n]
  - Request Method: GET
  - Request URI: /wireshark-labs/HTTP-wireshark-file1.html
  - Request Version: HTTP/1.1
  - Accept: image/gif, image/jpeg, image/pjpeg, image/pjpeg, application/x-shockwave-flash, \*/\*\r\n
  - Accept-Language: en-us\r\n
  - User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; windows NT 5.1; Trident/4.0)\r\n
  - Accept-Encoding: gzip, deflate\r\n
  - Host: gaia.cs.umass.edu\r\n
  - Connection: keep-alive\r\n
  - Cookie: \_\_utma=198765611.1717391459.1331778356.1331778356.1331778356.1; \_\_utmb=198765611.1.10.1331778356; \_\_utmc=198765611

Figure 2.12 Analysis of a website

## Analysis for [www.premiereleague.com](http://www.premiereleague.com)



The screenshot shows a Wireshark capture of network traffic on an AMD PCNET Family Ethernet Adapter. The filter is set to 'ip.addr == 208.87.35.100'. The packet list shows a TCP SYN packet (No. 1) from 10.0.2.15 to 208.87.35.100. The packet details pane shows the following information:

- Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
- Arrival Time: Mar 14, 2012 21:57:19.419545000 SA Pacific Standard Time
- Epoch Time: 1331780239.419545000 seconds
- [Time delta from previous captured frame: 0.000000000 seconds]
- [Time delta from previous displayed frame: 0.000000000 seconds]
- [Time since reference or first frame: 0.000000000 seconds]
- Frame Number: 1
- Frame Length: 62 bytes (496 bits)
- Capture Length: 62 bytes (496 bits)
- [Frame is marked: False]
- [Frame is ignored: False]
- [Protocols in frame: eth:ip:tcp]
- [Coloring Rule Name: HTTP]
- [Coloring Rule String: http || tcp.port == 80]
- Ethernet II, Src: CadmusCo\_66:24:58 (08:00:27:66:24:58), Dst: RealtekU\_12:35:02 (52:54:00:12:35:02)
- Destination: RealtekU\_12:35:02 (52:54:00:12:35:02)
- Address: RealtekU\_12:35:02 (52:54:00:12:35:02)

A command prompt window is open in the background, showing the following commands and output:

```
C:\WINDOWS\system32\cmd.exe
C:\>nslookup www.premiereleague.com
Server: Static-IP-1901572140.cable.net.co
Address: 190.157.2.140

Non-authoritative answer:
Name: www.premiereleague.com
Address: 208.87.35.100

C:\>
```

Figure 2.13 Analysis of another website, part 1

The screenshot shows a Wireshark capture on the AMD PCNET Family Ethernet Adapter. The filter is set to 'dns'. The packet list shows several DNS packets. Packet 40 is a standard query for 'pagead2.googlesyndication.com'. Packet 41 is the corresponding response. Packet 42 is another standard query for 'googleapis.l.google.com'. Packet 43 is the response. Packet 137 is a standard query for 'googleads.g.doubleclick.net'. Packet 140 is the response.

No.	Time	Source	Destination	Protocol	Length	Info
39	1.402585	10.0.2.15	190.157.2.140	DNS	79	Standard query A ajax.googleapis.com
40	1.403729	10.0.2.15	190.157.2.140	DNS	89	Standard query A pagead2.googlesyndication.com
41	1.431977	190.157.2.140	10.0.2.15	DNS	286	Standard query response CNAME pagead.l.doubleclick.net
42	1.431996	190.157.2.140	10.0.2.15	DNS	249	Standard query response CNAME googleapis.l.google.com
137	1.838683	10.0.2.15	190.157.2.140	DNS	87	Standard query A googleads.g.doubleclick.net
140	1.866762	190.157.2.140	10.0.2.15	DNS	272	Standard query response CNAME pagead.l.doubleclick.net

Frame 40: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on interface 0  
 Ethernet II, Src: CadmusCo\_66:24:58 (08:00:27:66:24:58), Dst: RealtekU\_12:35:02 (52:54:00:12:35:02)  
 Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 190.157.2.140 (190.157.2.140)  
 User Datagram Protocol, Src Port: blackjack (1025), Dst Port: domain (53)  
 Domain Name System (query)  
 [Response In: 41]  
 Transaction ID: 0x6a9f  
 Flags: 0x0100 (Standard query)  
 Questions: 1  
 Answer RRs: 0  
 Authority RRs: 0  
 Additional RRs: 0  
 Queries  
 pagead2.googlesyndication.com: type A, class IN  
 Name: pagead2.googlesyndication.com  
 Type: A (Host address)  
 Class: IN (0x0001)

Figure 2.14 Analysis of another website, part 2

### 2.2.3. Hand-In Assignment 3

#### 1. Connection between 2 point via TCP

When a connection is established between 2 hosts via TCP, bi-directional communication is established in order to send and receive data. Within the transport layer, protocols such as TCP or UDP are used when data travels; in the case of TCP, TCP segments play the role of data containers with a structure called a TCP segment.

TCP segments have header fields and data fields. In the former, a TCP segment is made up of 32-bit block, and numbers the composite ports (source and destination ports, each with 16 bits),

the sequence number, and the acknowledgement number, among others (Kurose & Ross, 2010: 245-246). The space for storing the source and destination ports is used in multiplexing/demultiplexing tasks with a view to matching segments with host processes.

In a simple example, the host *A* is connected via TCP with the host *B* with *x* and *y* ports respectively. According to the previous concepts, a segment that travels from *B* to *A* has in its composite port number the values *y* and *x* (source and destination). The following figure illustrates this:

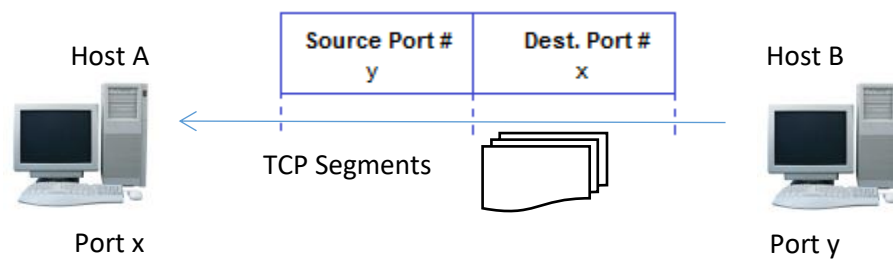


Figure 2.15 TCP segments

## 2. Reliable connections via UDP

An interesting experience, like that of a company called Javvin Technologies, shows how it is possible to reliably transfer data via UDP. Reliable UDP (or RUDP) consists of an acknowledgement mechanism to control reliable packets, windowing and congestion control, retransmission techniques on server, and overbuffering, among others (Javvin Technologies Inc., 2007: 46).

Reliable UDP was created by Bova and Krivoruchka at Cisco Systems, and the proposal was given to IETF in 1999, under the title: “Reliable UDP is based on RFCs 1151 and 908 - Reliable Data Protocol” (Network Working Group, 1999). In spite of the creators’ initiative, IETF never

adopted a standard at that time. Nonetheless, there are ways to create a RUDP-based connection by implementing specific software projects.

An example of how to implement a Reliable UDP connection can be seen in an open-source project called, “UDP-based Reliable Data Transfer Library” available at *The Code Project’s* Repository. This library, written in C++, allows the creation of UDP connections using reliable data transfer; the code is based on the socket principle, and it is fairly easy to implement in Visual C++/C# (In Microsoft Windows) or gcc (In GNU/Linux) environments. This Library provides a middle-layer called UDT between the application (socket-based) and the transport protocol (UDP). UDT has its own reliability and flow/congestion control over the UDP’s arena (The Code Project, 2006).

The authors successfully used a library in Microsoft Visual C++ .NET 2010 to these ends; two applications were created (the client one and the server one) with User Interfaces in Microsoft Foundation Classes (MFC). The project worked well, but there were some problems with the transfer of files over 512Mb.

### **3. An exception in Transmission Rates**

When two simultaneous TCP connections are trying to send packets through a bottleneck link of  $R$  bps, the congestion-control mechanism, should ensure that the bottleneck link works evenly. That is, if the link divides the total rate by the number of active connections ( $N$ ), each connection works at a rate of  $R/N$  (Kurose & Ross, 2010: 291).

Taking into account that there are two connections in this example, each connection should use a rate of  $R/2$  bps. The interesting point in this case, is how a TCP connection tries to establish

congestion control in the presence of bottleneck links; in other words, TCP is designed to enable a higher quality of service when it comes to the transportation of data via the Internet.

#### 4. Wireshark Lab

This exercise is about TCP segments in connection with <http://gaia.cs.umass.edu>. In the development of the image captured with Wireshark, the packets were caught as the figure below shows:

No.	Time	Source	Destination	Protocol	Length	Info
2	9.506964	10.0.2.15	128.119.245.12	TCP	62	avocent-proxy > http [SYN] Seq=0 w
3	9.878743	128.119.245.12	10.0.2.15	TCP	60	http > avocent-proxy [SYN, ACK] Seq
4	9.878762	10.0.2.15	128.119.245.12	TCP	54	avocent-proxy > http [ACK] Seq=1 A
5	9.879524	10.0.2.15	128.119.245.12	HTTP	360	GET / HTTP/1.1
6	9.879916	128.119.245.12	10.0.2.15	TCP	60	http > avocent-proxy [ACK] Seq=1 A
7	10.044248	128.119.245.12	10.0.2.15	TCP	1474	[TCP segment of a reassembled PDU]
8	10.044277	128.119.245.12	10.0.2.15	TCP	94	[TCP segment of a reassembled PDU]
9	10.044303	10.0.2.15	128.119.245.12	TCP	54	avocent-proxy > http [ACK] Seq=307
10	10.044920	128.119.245.12	10.0.2.15	TCP	1474	[TCP segment of a reassembled PDU]
11	10.044941	128.119.245.12	10.0.2.15	TCP	94	[TCP segment of a reassembled PDU]
12	10.044966	10.0.2.15	128.119.245.12	TCP	54	avocent-proxy > http [ACK] Seq=307
13	10.114245	10.0.2.15	128.119.245.12	TCP	62	socks > http [SYN] Seq=0 win=64240
16	10.156651	10.0.2.15	72.3.245.222	TCP	62	pvuniwien > http [SYN] Seq=0 win=6
18	10.166471	128.119.245.12	10.0.2.15	HTTP	88	HTTP/1.1 200 OK (text/html)
20	10.204110	10.0.2.15	207.46.15.254	TCP	62	amt-esd-prot > https [SYN] Seq=0 W

Frame	Time	Source	Destination	Protocol	Length	Info
2	9.506964	10.0.2.15	128.119.245.12	TCP	62	avocent-proxy > http [SYN] Seq=0 w

Frame 2: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)  
 Ethernet II, Src: CadmusCo\_25:23:b1 (08:00:27:25:23:b1), Dst: RealtekU\_12:35:02 (52:54:00:12:35:02)  
 Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 128.119.245.12 (128.119.245.12)  
 Transmission Control Protocol, Src Port: avocent-proxy (1078), Dst Port: http (80), Seq: 0, Len: 0

Figure 2.16 TCP analysis

The results can be seen in the following figure:

IP Address:	128.119.245.12
18 10.166471 128.119.245.12 10.0.2.15 HTTP 88 HTTP/1.1 200 OK (text/html)	
20 10.204110 10.0.2.15 207.46.15.254 TCP 62 amt-esd-prot > https [SYN] Seq=0 win=64	
24 40.333330 128.119.245.12 10.0.2.15	
Frame 18: 88 bytes on wire (704 bits), 88 bytes captured (704 bits)	
Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: CadmusCo_25:23:b1 (08:00:27:25:23:b1)	
Internet Protocol Version 4, Src: 128.119.245.12 (128.119.245.12), Dst: 10.0.2.15 (10.0.2.15)	
Version: 4	
Header length: 20 bytes	
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))	

Figure 2.17 Getting information based on TCP

Client IP Address and Port:	IP: 10.0.2.15 and Port: TCP 80 (http)
20 10.204110 10.0.2.15 207.46.15.254 TCP 62 amt-esd-prot > https [SYN] Seq=0 win=64240 Len=0	
24 40.333330 128.119.245.12 10.0.2.15	
Transmission Control Protocol, Src Port: http (80), Dst Port: avocent-proxy (1078), Seq: 2921, Ack: 307, Len: 34	
Source port: http (80)	
Destination port: avocent-proxy (1078)	
[Stream index: 1]	
Sequence number: 2921 (relative sequence number)	
[Next sequence number: 2955 (relative sequence number)]	

Figure 2.18 Port analysis

Sequence number on HTTP POST command:	152248
147 3.203632 10.0.2.15 128.119.245.12 HTTP 869 POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)	
148 3.203648 128.119.245.12 10.0.2.15 TCP 60 http > nt2-discover [ACK] Seq=1 Ack=153063 win=43313 Len=0	
Frame 147: 869 bytes on wire (6952 bits), 869 bytes captured (6952 bits)	
Ethernet II, Src: CadmusCo_25:23:b1 (08:00:27:25:23:b1), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)	
Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 128.119.245.12 (128.119.245.12)	
Transmission Control Protocol, Src Port: pt2-discover (1101), Dst Port: http (80), Seq: 152248, Ack: 1, Len: 815	
Source port: pt2-discover (1101)	
Destination port: http (80)	
[Stream index: 1]	
Sequence number: 152248 (relative sequence number)	
[Next sequence number: 153063 (relative sequence number)]	
Acknowledgement number: 1 (relative ack number)	
Header length: 20 bytes	
Flags: 0x18 (PSH, ACK)	
000. .... = Reserved: Not set	
0. .... = Nonce: Not set	

Figure 2.19 Getting information of a port

First six TCP segments' length:				1 <sup>st</sup> : 735 bytes, the rest: 1460 bytes			
147	3.203632	10.0.2.15	128.119.245.12	HTTP	869	POST	/ethereal-labs/lab3-1-reply.htm HTTP/1.1
148	3.203848	128.119.245.12	10.0.2.15	TCP	60	http >	pt2-discover [ACK] Seq=1 Ack=153063 win
<div> <div>Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 128.119.245.12 (128.119.245.12)</div> <div> <div>Transmission Control Protocol, Src Port: pt2-discover (1101), Dst Port: http (80), Seq: 152248, Ack: 1, Len: 815</div> <div> <div>[107 Reassembled TCP Segments (153062 bytes): #8(735), #10(1460), #11(1460), #13(1460), #14(1460), #16(1460), #17</div> <div> <div>[Frame: 8, payload: 0-734 (735 bytes)]</div> <div>[Frame: 10, payload: 735-2194 (1460 bytes)]</div> <div>[Frame: 11, payload: 2195-3654 (1460 bytes)]</div> <div>[Frame: 13, payload: 3655-5114 (1460 bytes)]</div> <div>[Frame: 14, payload: 5115-6574 (1460 bytes)]</div> <div>[Frame: 16, payload: 6575-8034 (1460 bytes)]</div> </div> </div> </div> </div>							

Figure 2.20 Segments tracking

The image is stored in a file that it is available at <http://190.254.4.13/Week3.zip>. All the screenshots are provided in this document.

## 2.2.4. Hand-In Assignment 4

### 1. IP Addresses in routers

A router has network interfaces that are used to connect hosts (or subnets). Therefore, each network interface must use an IP address so as to be identifiable in the context of the Internet. The number of IP addresses in a router depends on the number of physical interfaces the router has. A router has at least 2 links, called interfaces (Kurose & Ross, 2010: 348).

### 2. Routing Algorithms

Within the network layer, routing is one of the main functions by which data can reach its destination under certain performance criteria, for example: spent time, traffic, and computational cost among others. These criteria can be managed by implementing routing algorithms; in the case

of the Internet's routing system, there are two algorithms: the Link-State (LS) routing algorithm and the Distance-Vector (DV) routing algorithm.

The foundations of these algorithms are explained in the table below:

LSRA	DVRA
<ul style="list-style-type: none"><li>• Inter-domain protocol</li><li>• Each router has link states with its neighbors and the rest of the network</li><li>• Shortest route is computed</li></ul>	<ul style="list-style-type: none"><li>• Each router has a list of known distances and routes</li><li>• These lists are updated constantly with their neighbors</li><li>• Each router finds the best route with each neighbor, this process is continuous</li></ul>

Table 2.8 LSRA vs DVRA

A technical comparison between LSRA and DVRA stated that, “LSRP has quicker convergence speed and higher flexibility” (Feixue, Yong, & Zhijie, 2010: 22). This is because LSRA update the link's state information only when the network changes into a new topology. In conclusion, LSRA is more complex than DVRA; although, LSRA could be damaged by delays and the oscillation phenomenon.

As an example, a simulated environment of a bank network was established by Feixue, Yong, and Zhijie. In their experiment, a network with 20 nodes was simulated. The results established that LSRA came first place when it came to time cost. In spite of the fact that the first data packets were delivered with better time cost records by DVRA, LSRA was able to maintain its lead over DVRA, due to LSRA sending less information, and using less bandwidth once the routing process converged. In conclusion, LSRA has more scalability than DVRA, when networks have a tendency to grow to huge proportions (Feixue, Yong, & Zhijie, 2010: 26).



### 3. Dijkstra's Algorithm Example

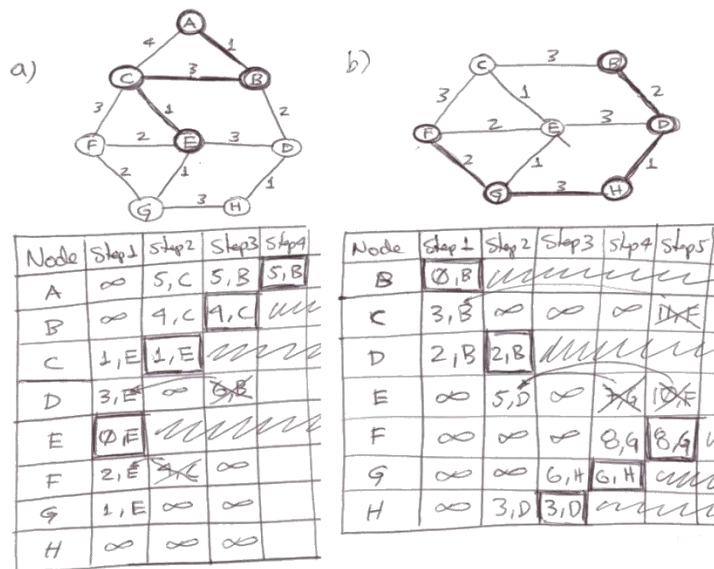


Figure 2.21 Mockups of Dijkstra's algorithm

### 4. Wireshark Lab

#### PART 1:

This figure shows Google's IP address with a *ping* command at three different times of day:

<pre>C:\Users\Administrator&gt;ping www.google.com  Pinging www.l.google.com [74.125.45.147] with 32 bytes of data: Reply from 74.125.45.147: bytes=32 time=81ms TTL=52 Reply from 74.125.45.147: bytes=32 time=82ms TTL=52 Reply from 74.125.45.147: bytes=32 time=90ms TTL=52 Reply from 74.125.45.147: bytes=32 time=91ms TTL=52  Ping statistics for 74.125.45.147:     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),     Approximate round trip times in milli-seconds:         Minimum = 81ms, Maximum = 91ms, Average = 86ms</pre>	<pre>C:\&gt;ping www.google.com  Pinging www.l.google.com [74.125.45.147] with 32 bytes of data: Reply from 74.125.45.147: bytes=32 time=91ms TTL=52 Reply from 74.125.45.147: bytes=32 time=107ms TTL=52 Reply from 74.125.45.147: bytes=32 time=93ms TTL=52 Reply from 74.125.45.147: bytes=32 time=108ms TTL=52  Ping statistics for 74.125.45.147:     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),     Approximate round trip times in milli-seconds:         Minimum = 91ms, Maximum = 108ms, Average = 99ms</pre>	<pre>C:\Users\Administrator&gt;ping www.google.com  Pinging www.l.google.com [74.125.45.99] with 32 bytes of data: Reply from 74.125.45.99: bytes=32 time=90ms TTL=52 Reply from 74.125.45.99: bytes=32 time=90ms TTL=52 Reply from 74.125.45.99: bytes=32 time=91ms TTL=52 Reply from 74.125.45.99: bytes=32 time=107ms TTL=52  Ping statistics for 74.125.45.99:     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),     Approximate round trip times in milli-seconds:         Minimum = 90ms, Maximum = 107ms, Average = 94ms</pre>
9 am (86ms round trip delay)	1 pm (99ms round trip delay)	7 pm (94ms round trip delay)

Figure 2.22 Ping latency

In Colombia the time is GMT-5. Google's IP address ([www.google.com](http://www.google.com)) is 74.125.45.147, and the local IP Address is 192.168.0.10. Colombia is behind the ISP's server, its public IP, which is 186.144.16.180.

The information gathered using the *tracert* command is shown in the following figure:

<pre> C:\Users\Administrator&gt;tracert www.google.com Tracing route to www.l.google.com [74.125.45.147] over a maximum of 30 hops:   0  &lt;1 ms  &lt;1 ms  &lt;1 ms  router.udenar.edu.co [200.21.87.161]   1  1 ms  4 ms  1 ms  10.7.2.31   2  31 ms  31 ms  31 ms  10.32.0.186   3  66 ms  66 ms  66 ms  Xe6-0-1-grtbaqt2.red.telefonica-wholesale.net   4  66 ms  66 ms  66 ms  Xef-0-2-0-grtniahr3.red.telefonica-wholesale.net   5  66 ms  66 ms  66 ms  Xef-0-2-0-grtniahr3.red.telefonica-wholesale.net   6  70 ms  68 ms  68 ms  Xef-0-2-0-grtniahr3.red.telefonica-wholesale.net   7  75 ms  68 ms  68 ms  GOOGLE-xe-6-1-0-0-grtniama3.red.telefonica-wholesale.net   8  68 ms  68 ms  68 ms  209.85.253.74   9  87 ms  81 ms  81 ms  209.85.254.252  10  82 ms  83 ms  82 ms  65.249.94.22  11  90 ms  81 ms  81 ms  209.85.248.29  12  84 ms  89 ms  89 ms  209.85.253.133  13  82 ms  82 ms  81 ms  gx-in-f147.1e100.net [74.125.45.147] Trace complete. </pre>	<pre> C:\&gt;tracert www.google.com Tracing route to www.l.google.com [74.125.45.105] over a maximum of 30 hops:   0  &lt;1 ms  &lt;1 ms  &lt;1 ms  10.0.2.2   1  *  *  *  Request timed out.   2  *  *  *  Request timed out.   3  *  *  *  Request timed out.   4  *  *  *  Request timed out.   5  *  *  *  Request timed out.   6  *  *  *  Request timed out.   7  *  *  *  Request timed out.   8  *  *  *  Request timed out.   9  *  *  *  Request timed out.  10  *  *  *  Request timed out.  11  *  *  *  Request timed out.  12  *  *  *  Request timed out.  13  *  *  *  Request timed out.  14  *  *  *  Request timed out.  15  *  *  *  Request timed out.  16  94 ms  95 ms  *  gx-in-f105.1e100.net [74.125.45.105]  17  95 ms  96 ms  94 ms  gx-in-f105.1e100.net [74.125.45.105] Trace complete. </pre>	<pre> C:\Users\Administrator&gt;tracert www.google.com Tracing route to www.l.google.com [74.125.65.104] over a maximum of 30 hops:   0  1 ms  &lt;1 ms  &lt;1 ms  192.168.0.1   1  *  *  *  Request timed out.   2  *  *  *  Request timed out.   3  14 ms  11 ms  11 ms  172.22.111.230   4  16 ms  32 ms  34 ms  172.22.32.237   5  20 ms  17 ms  37 ms  190.85.254.145   6  *  *  *  Request timed out.   7  60 ms  64 ms  68 ms  so-1-0-0.usa.brx-edge01.columbus-net   8  78 ms  78 ms  78 ms  63.245.43.571   9  80 ms  78 ms  79 ms  63.245.3.34  10  79 ms  79 ms  79 ms  209.85.253.120  11  93 ms  91 ms  93 ms  216.239.48.192  12  90 ms  88 ms  89 ms  209.85.254.247  13  112 ms  104 ms  93 ms  209.85.253.218  14  91 ms  90 ms  91 ms  gx-in-f104.1e100.net [74.125.65.104] Trace complete. </pre>
9 am (81ms round trip delay)	1 pm (94ms round trip delay)	7 pm (91ms round trip delay)

Figure 2.23 Tracert outputs

The difference in routing is very noticeable; at 9am there were 13 routers and 7pm there were 14 routers; however near noon, 17 routers were listed, but most with request time out. Different information has been reported concerning the “ping” command, and it is obvious from this experiment that paths change according to the time of day.

## PART 2:

By using ICMP, it can be seen that the values of the datagram that always change are related to the identification of each element, that is to say, the sequence number BE and LE.

```

Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x2648 [correct]
Identifier (BE): 1 (0x0001)
Identifier (LE): 256 (0x0100)
Sequence number (BE): 199 (0x00c7)
Sequence number (LE): 50944 (0xc700)
Data (28 bytes)
Data: 304550696e67506c6f747465725374616e64617264332e33...
[Length: 28]

```

Figure 2.24 ICMP tracking

### PART 3:

The results, which follow the previous image captured with Wireshark in the ICMP study, are shown in the table below:

	ping request packets sent by our host	ping reply packet
ICMP Type and Code	8 echo, 0 code	0 Echo replay, 0 code
Other ICMP fields	Data	User datagram protocol
Checksum bytes	0x2e39 [correct]	0xfbb0 [correct]
Sequence numbers and identifier fields	Identifier (BE/LE) Seq. Numbers (BE/LE)	Identifier (BE/LE) Seq. Numbers (BE/LE)

Table 2.9 ICMP according Wireshark

#### 2.2.5. Hand-In Assignment 5

##### 1. Link-layer Protocol's Services

According to Kurose and Ross, the possible services of a Link-Layer Protocol include Framing, Link Access, Reliable Delivery, Flow Control, Error Detection, Error Correction, and transmissions via Half-duplex and Full-duplex techniques (Kurose & Ross, 2010: 444-445). As a link-layer protocol belongs to the Internet protocol stack, its services must establish communication with adjacent layers. This is to ensure that there is close correspondence between link-layer protocol's services and the other services located in upper layers.

The transport layer and the network layer in the Internet protocol stack are mostly ruled by TCP/UDP and IP respectively and so the correspondence of a link-layer protocol's services with TCP and IP are shown in the table below:

Link-layer Protocol's Services	TCP's Services	IP's Services
Reliable Delivery, Error Detection, and Error Connection	Reliable Data Transfer	Guaranteed Delivery
Flow Control	Flow Control	In-order Packet Delivery

Table 2.10 LLP, TCP and IP Services

The relationship between a Link-layer protocol and TCP/IP is based on reliable communication between nodes; in this case, a Link-layer protocol interacts with the services from the upper protocols.

## 2. Comparison of frame structures

Ethernet technology is controlled by IEEE standards. In this sense, each single implementation has its own standard; for 10BASE-T, 100BASE-T, and Gigabit Ethernet, the standards defined by IEEE are 802.3i (1990), 802.3u (1995), and 802.3ab (1999) respectively. All of these are specifications for twisted pair cable.

All the aforementioned standards have the same origin: the Ethernet original specification which includes the definition of the frame structure in the link layer. This common heritage can be observed in the figure below (left), the MAC data format is for the two first implementations. Gigabit Ethernet however, has an additional field.

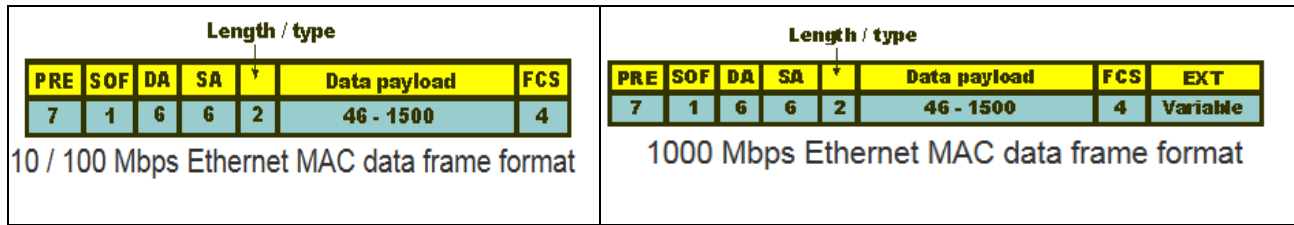


Figure 2.25 MAC Address format

Source: (Radio-Electronics.com, n.d.)

The main difference between Gigabit Ethernet and the other two specifications resides in the EXT field of the frame. According to the IEEE 802.3z, technologies based on 1000BASE-X and 1000BASE-T have different minimum frame sizes: 416 bytes for the former and 520 bytes for the latter. In order to work properly with both Gigabit technologies, the field EXT was created as a non-data variable extension (Radio-Electronics.com, n.d.).

### 3. CSMA and collision detection

According to the following figure, 4 nodes (A, B, C and D) that belong to a broadcast bus are transmitting frames through the network.

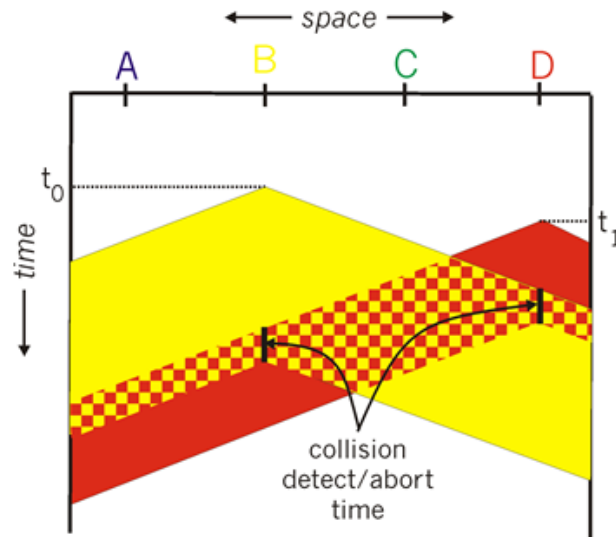


Figure 2.26 CSMA collision detection

Source: (Kurose & Ross, 2010: 466)

Taking into account the above figure, the first node that detects a collision is node D. In the case that B's frames are addressed to node A, it will partially receive B's frames starting at a time near to  $t_1$ . The entire set of frames will not be received by node A, because it will receive a jamming signal which means that there is a frame collision on the network, before the last of B's frames reach node A. In accordance with the figure above, the jamming signal represents the frames collision, and it is generated by the link-layer protocol.

Another case in which a collision happens, and only node A manages to receive the whole packet, can be illustrated as follows:

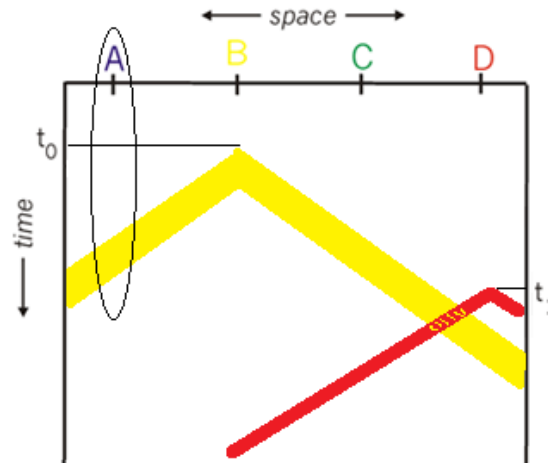


Figure 2.27 Other collision

In the previous figure, it can be seen that a collision happens just after  $t_1$ ; in this case node B starts transmitting its frames at  $t_0$ , and node A will receive the entire set of B's frames despite the collision. This is because the collision happened after node A received B's entire set of frames.

The following, straightforward scenario in which there is no collision, is seen below:

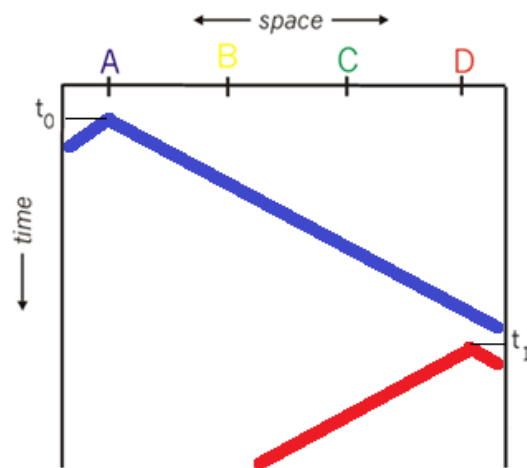


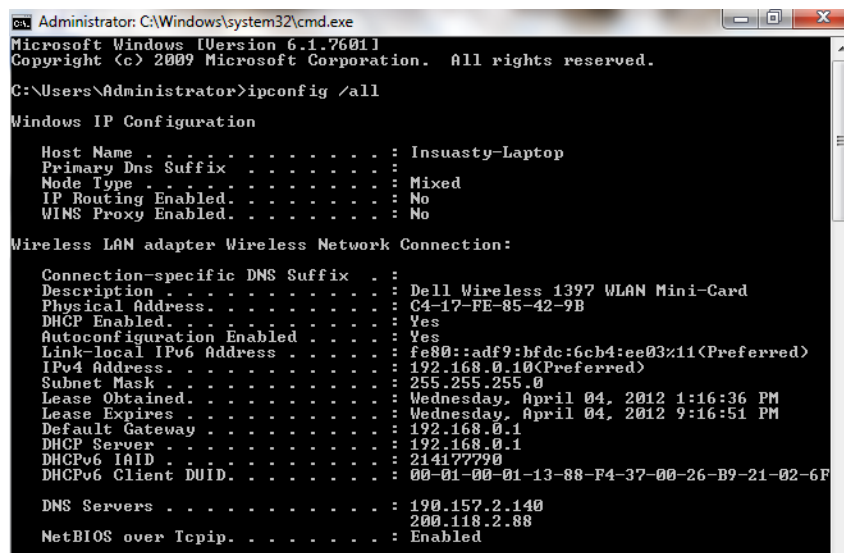
Figure 2.28 No collision detected

The “backs off” is the amount of time a node waits before starting a new transmission; these lapses of time are calculated randomly (Kurose & Ross, 2010: 464). It happens randomly in order to avoid synchronizing all the nodes at the moment of them sending frames.

#### 4. Wireshark Lab

##### PART I:

When the *ipconfig* command was used on a laptop, the following results ensued:



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ipconfig /all

Windows IP Configuration

Host Name . . . . . : Insuasty-Laptop
Primary Dns Suffix . . . . . :
Node Type . . . . . : Mixed
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Wireless LAN adapter Wireless Network Connection:

Connection-specific DNS Suffix . :
Description . . . . . : Dell Wireless 1397 WLAN Mini-Card
Physical Address. . . . . : C4-17-FE-85-42-9B
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::adf9:bfdc:6cb4:ea03%11(Preferred)
IPv4 Address. . . . . : 192.168.0.10(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Wednesday, April 04, 2012 1:16:36 PM
Lease Expires . . . . . : Wednesday, April 04, 2012 9:16:51 PM
Default Gateway . . . . . : 192.168.0.1
DHCP Server . . . . . : 192.168.0.1
DHCPv6 Iaid . . . . . : 214172720
DHCPv6 Client DUID. . . . . : 00-01-00-01-13-88-F4-37-00-26-B9-21-02-6F

DNS Servers . . . . . : 190.157.2.140
                       200.118.2.88
NetBIOS over Tcpip. . . . . : Enabled
```

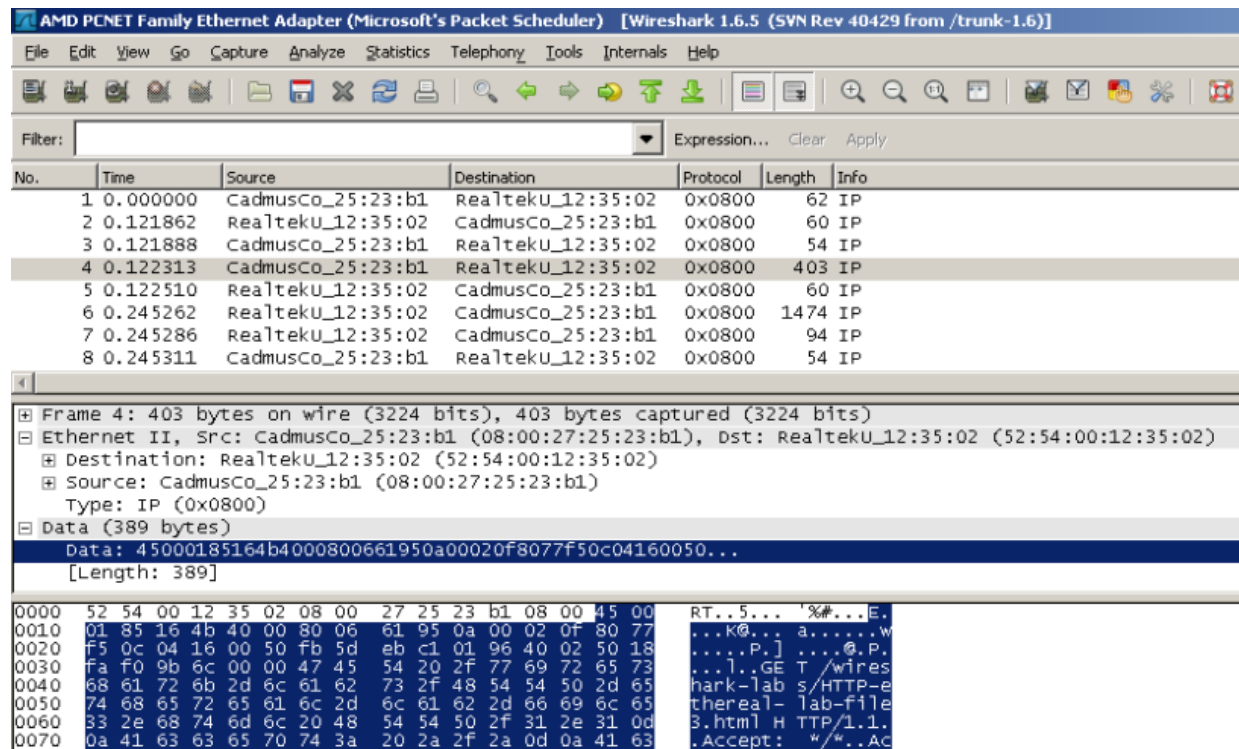
Figure 2.29 Ipconfig output

Most laptops have several network interfaces, between real and virtual ones; however, the laptop used gains access to the Internet by a unique wireless LAN adapter (Dell Wireless 1397 WLAN Mini-Card). According to the results of the *ipconfig* command, there is a DHCP server present on the local network whose IP address is 192.168.0.1. The default gateway has the same IP address as the DHCP server, and two DNS Servers are reported using the addresses 190.157.2.140 and 200.118.2.88, which correspond to our ISP’s servers.



## PART II:

Using Wireshark, the following information was established about the link-layer protocol when the web browser loaded <http://gaia.cs.umass.edu/wireshark-labs/HTTP-ethereal-lab-file3.html>.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	CadmusCo_25:23:b1	RealtekU_12:35:02	0x0800	62	IP
2	0.121862	RealtekU_12:35:02	CadmusCo_25:23:b1	0x0800	60	IP
3	0.121888	CadmusCo_25:23:b1	RealtekU_12:35:02	0x0800	54	IP
4	0.122313	CadmusCo_25:23:b1	RealtekU_12:35:02	0x0800	403	IP
5	0.122510	RealtekU_12:35:02	CadmusCo_25:23:b1	0x0800	60	IP
6	0.245262	RealtekU_12:35:02	CadmusCo_25:23:b1	0x0800	1474	IP
7	0.245286	RealtekU_12:35:02	CadmusCo_25:23:b1	0x0800	94	IP
8	0.245311	CadmusCo_25:23:b1	RealtekU_12:35:02	0x0800	54	IP

Frame 4: 403 bytes on wire (3224 bits), 403 bytes captured (3224 bits)	
Ethernet II, Src: CadmusCo_25:23:b1 (08:00:27:25:23:b1), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)	
Destination: RealtekU_12:35:02 (52:54:00:12:35:02)	
Source: CadmusCo_25:23:b1 (08:00:27:25:23:b1)	
Type: IP (0x0800)	
Data (389 bytes)	
Data: 45000185164b4000800661950a00020f8077f50c04160050...	
[Length: 389]	

0000	52 54 00 12 35 02 08 00	27 25 23 b1 08 00 45 00	RT..5...	'%#...E.
0010	01 85 16 4b 40 00 80 06	61 95 0a 00 02 0f 80 77	...K@... a.....w	
0020	f5 0c 04 16 00 50 fb 5d	eb c1 01 96 40 02 50 18	...P.] ...@.P.	
0030	fa f0 9b 6c 00 00 47 45	54 20 2f 77 69 72 65 73	...l..GE T/wires	
0040	68 61 72 6b 2d 6c 61 62	73 2f 48 54 54 50 2d 65	hark-lab s/HTTP-e	
0050	74 68 65 72 65 61 6c 2d	6c 61 62 2d 66 69 6c 65	thetreal- lab-file	
0060	33 2e 68 74 6d 6c 20 48	54 54 50 2f 31 2e 31 0d	3.html H TTP/1.1.	
0070	0a 41 63 63 65 70 74 3a	20 2a 2f 2a 0d 0a 41 63	.Accept: */*..Ac	

Figure 2.30 Wireshark's IP analysis

By filtering the IP data, the HTTP GET message was extracted from frame number 4. In this frame, the 48-bit Ethernet addresses of the destination and the source are 52:54:00:12:35:02 and 08:00:27:25:23:b1 respectively. The destination address does not correspond to the site [gaia.cs.umass.edu](http://gaia.cs.umass.edu). The 48-bit destination address corresponds to another node other than that of the main site, perhaps a router.

In the matter of the 48-bit source address, again the value does not correspond to the correct IP address. It indicates a server of the same ISP.

In conclusion, both 48-bit addresses for destination and source are MAC addresses (Kurose & Ross, 2010: 478); this means that the addresses shown in those fields within a link-layer frame are physical addresses of the devices in a certain host. In the case of the server, a MAC address could correspond to a NIC in the main gateway at the server-side. In the case of the 48-bit source address, a MAC address could correspond to a NIC in the main gateway at our ISP's side.

## **2.2.6. Hand-In Assignment 6**

### **1. Wireless Infrastructure Mode**

The infrastructure mode on wireless networks relates to when hosts belong to a wireless network using a base station. In this kind of configuration, network services such as address assignment and routing are present in communication between the hosts, with the base station as an access point to the wireless network (Kurose & Ross, 2010: 527).

With a wireless network using the infrastructure mode, hosts connected to the network can get access to external resources if the configuration of the base station allows. In traditional environments, such access points in 802.11, wireless LANs are connected to a backbone communication system to gain access to the Internet.

In contrast, it is possible to establish a wireless network without base stations; however, the set of services that are present in infrastructure mode must be provided by the hosts themselves. This type of configuration is known as *Ad Hoc* network. When a set of hosts form an Ad Hoc wireless network, they do not get access to the external resources (Kurose & Ross, 2010: 527).

## **2. Does Wireless mean Mobile?**

The short answer is: No, it does not. The definition of Wireless pertains to environments where no physical cable (wire) is used. With this in mind, it is possible to create wireless networks in which the hosts are computer desktops or even fixed laptops. This scenario is fairly common in SOHO (Small Offices Home Offices) contexts; this means that each host within the SOHO must have at least one wireless network adapter to belong to the wireless network.

Perhaps one of the main goals of a wireless network is to promote mobility of devices. However, in spite of that wireless “spirit”, it is not a given that all hosts connected to a wireless network are mobile devices, because it is possible to connect fixed workstations to a wireless network if they have wireless NICs are enabled.

For example, if a user at home has a laptop connected to the home’s wireless network, this user can get access to the Internet in any place in his/her home using a unique access point. In spite of the user having a laptop, in the context of the home’s wireless network the laptop is not considered a ‘mobile device’, because it never changes wireless network (in fact, it never changes its access point either). In this example, no matter where the laptop is in the house, it always belongs to the same wireless network. The laptop is considered a fixed workstation.

## **3. Client Buffer in media streaming**

According to the following figure, a streaming server can interact with the client’s media player by using a client buffer.

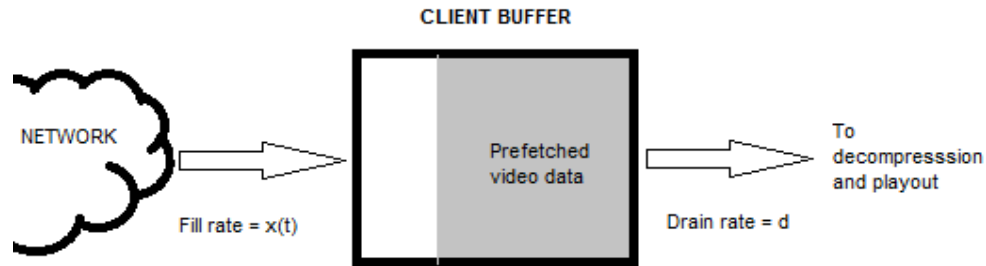


Figure 2.31 Client buffer behavior

Source: (Kurose & Ross, 2010: 611)

In this figure, the fill rate is represented as  $x(t)$  and the drain rate is  $d$ . Three possible variables will be analyzed as follows:

- **1<sup>st</sup> Situation: the buffer is big enough to contain the whole media**

This would be the ideal situation in terms of performance. If the video sent from the streaming server is smaller than the client buffer, the entire video would fit in the client buffer, so its playout would be practically immediate. Additionally in this situation  $x(t) = d$ , it is more likely there would be no missing packets, as there is free space in the client buffer.

- **2<sup>nd</sup> Situation: the buffer can only store one-third of the media**

In this situation, if the client buffer is only one-third of the sent video's size, it will have to completely fill the client buffer three times in order to complete the delivery. In a fixed playout delay approach, it is more likely that packets will be misplaced than in the previous scenario, due to the jitter effect on the transportation of media chunks over the network.

- **3<sup>rd</sup> Situation: the buffer can contain only one-tenth of the media**

In this situation, the misplacement of packets is even more likely than in the two previous scenarios. This is the same as the previous case, but with even less storage space in the client buffer. This implies that the drain rate would be faster than in the first two examples when

playing the video on the client side. When the client buffer is small, it may be a good idea to change the transport protocol from TCP to UDP, as the latter uses a non-reliable transfer approach.

#### **4. Preemptive Priority Queuing**

As its name suggests, a non-preemptive discipline ensures that no packet can be interrupted once its transmission starts (Kurose & Ross, 2010: 655). In contrast, by following the principles of preemptive priority queuing, packet 2 must be interrupted if a packet which has a higher priority arrives (for example packet 4).

Applying this approach to the video streaming exercise and having some lost packets can still have an “acceptable” level of lost information. That is to say, a video missing some packets may not alter its general viewing experience when played. In this particular case, packet 2 must be interrupted and it never reaches the final destination.

In addition, delaying the play-out schedule may not be so relevant for the client side, if the server has a preemptive approach in priority queuing. The reason for that is, some packet transmissions can be interrupted according to policies relating to packet priority on the server side. Therefore, the interrupted packets simply do not reach the client media player. Also, the technique which delays the play-out schedule could be useful in the cases where the client buffer is small, and the quantity of media is large.

### **2.2.7. Hand-In Assignment 7**

#### **1. Confidentiality and Integrity: features of messaging**

In an environment where private messages are circulating on a network, security regarding their contents is required. In terms of securing private messages, two features come into play: confidentiality and integrity.

Confidentiality is strictly related to privacy; this means that the messages have a source and a fixed destination (or a set of destinations). The destination may see the content of the sent messages, and no one else. This function of private messaging works under confidentiality criteria; in general, confidentiality is provided by encryption techniques in order to avoid unauthorized readings of private content (Kurose and Ross, 2010: 704).

In contrast, message integrity is related to the preservation of the original content while the message is traveling on the network. In addition, it is important to be sure that these messages were in fact transmitted by the appropriate source, and that the messages were not altered in transit (Kurose and Ross, 2010: 705).

In accordance with the above, confidentiality and integrity can work separately, but they are complementary tasks in terms of security. It is therefore possible to have confidentiality without integrity; in that case the message can be strongly encrypted so as no one can read its content, but there is no guarantee that the message has not been altered. It is also possible to have integrity without confidentiality, because it is feasible to establish a secure communication channel with no intruders, so messages could be transmitted, without employing any encryption technique, to their destinations.

In conclusion however, the above alternatives are not recommended because they both imply security risks. The ideal scenario, and one in which security is stronger, is one in which confidentiality and integrity work simultaneously.

## **2. SSL handshake and random nonces**

A nonce is a unique number used as an identifier number while messages are being transmitted by a specific protocol (Kurose and Ross, 2010: 717). The reason for using random nonces is to avoid attacks that use the same static nonce. With a random sequence, it is more difficult (but not impossible) to guess the identifiers of packets in an SSL-based transmission. In a scenario in which random nonces each run the protocol, intruders would have an enormous task in replaying the verification messages in the runs of SSL-based communication (Mitchell, Shmatikov, & Stern, 1997).

## **3. Application Gateway and Router Filters**

Within the Internet protocol stack, application gateways are specific software running in the application layer. They are in charge of analyzing application data in matters of security. These types of software work over the level of IP/TCP/UDP, and they establish the policies to process information. Given these conditions, an application gateway needs to extend its coverage into lower layers, such as network-layer devices. In this way, router filters in the network layer deal with IP-based packets. Their mission is to control the content in the network layer, establishing security filters. In order to work with security across different layers in the Internet Protocol Stack, an application gateway must work alongside router filters.

#### 4. Mono-alphabetic Cipher: an example

Using the encryption method below, known as a mono-alphabetic cipher, the correspondence between original and ciphered text is presented in the following table:

plaintext: abcdefghijklmnopqrstuvwxyz	
ciphertext: mnbvcxzasdfghjklpoiuytrewq	
Source: (Kurose & Ross, 2010: 693).	
Plaintext	Ciphertext
We are students of MSc	<i>s mh m iuyvcju kx hib su</i>
We do not think so	<i>s vk jku uasjf ik</i>

Table 2.11 Sample of ciphered text

#### PGP Exercise

A text file with the directions for this assignment was encrypted using PGP 6.0.2i. The process is shown in the following screenshots:

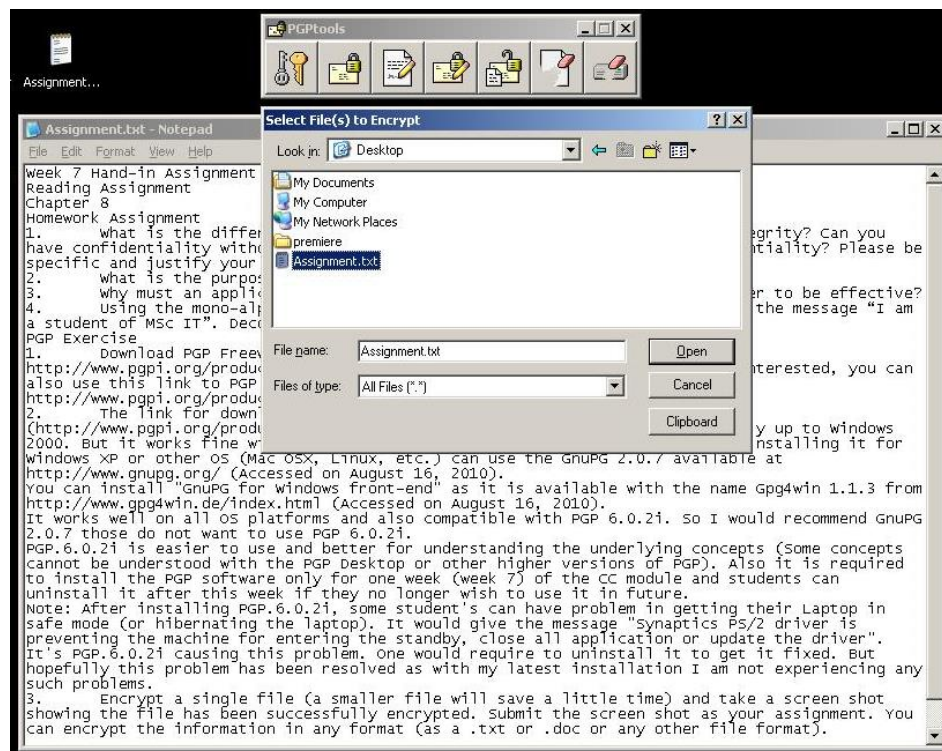


Figure 2.32 Open a source



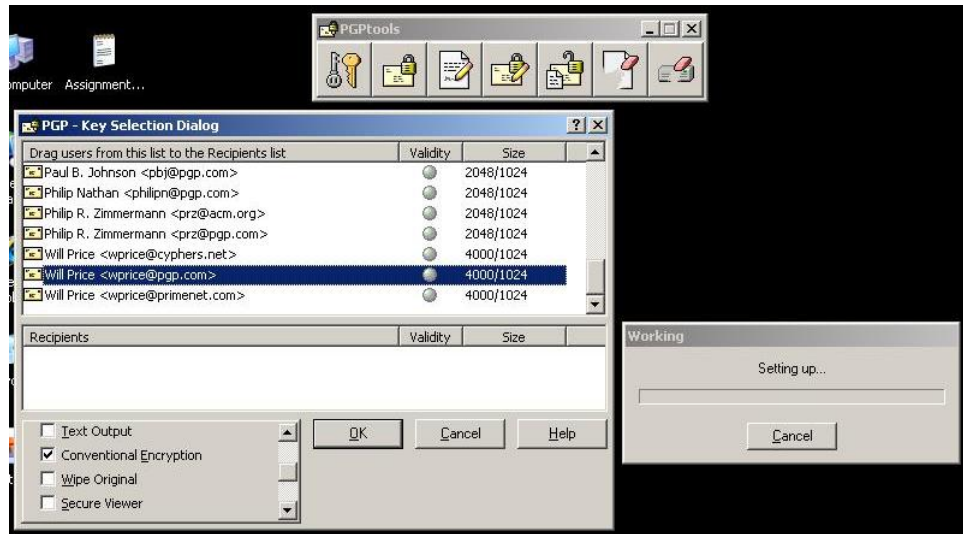


Figure 2.33 Key selection

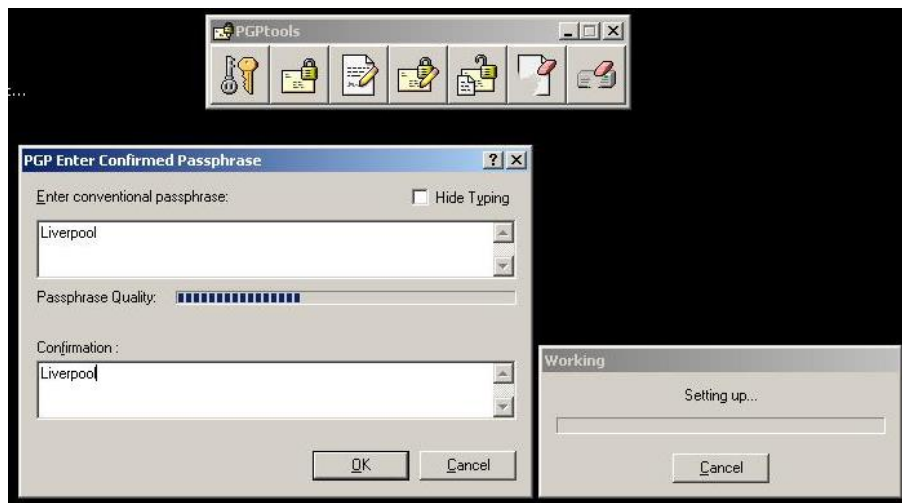


Figure 2.34 Using passphrase



Figure 2.35 Output

The encryption process was done using the selected key provided by Will Price ([wprice@pgp.com](mailto:wprice@pgp.com)) and the conventional encryption technique. The passphrase “Liverpool” was used, and as final result, a file 1KB smaller than the original was produced. Obviously its content is encrypted, so it is not possible to examine it using Windows Notepad. Instead, PGP software is required to see its content, once the correct passphrase is provided.

## **2.2.8. Hand-In Assignment 8**

### **1. The ASN.1 tree**

The tree structure of ASN.1 is based on the definition of data types, which are in direct interaction with the relationship between objects. In this “semantic” structure, all items are represented hierarchically. This proposal has its roots in ISO and the Telecommunication Standardization sector of the International Telecommunication Union (Kurose & Ross, 2010: 786).

Looking at the previous definition, the subject of the standard 1.2.840.113549 is RSADSI. Everything related to RSADSI is defined through RFC 1423. Therefore, the definition of a set of security algorithms will be discussed, including DES and RSA (MD2/MD5), among other composite features (Network Working Group, 1993: 1).

On the other hand, the location of new Microsoft file formats should be placed in the ASN.1 notation as “*{iso(1) member-body(2) us(840) microsoft(113556) ms-files(4) ms-word(2)}*”. Using this, it is easy to locate the standard in the tree structure of ASN.1 with the following path (France Telecom, 2012).

## **2. The difference between request-response message and SNMP's trap message**

The main behavior of the request-response mode is related to the interaction between entities and SNMP agents. These agents wait for an entity's request, then the request is processed in order to give a final answer. This interaction is based on queries and modification actions over the MIB object's values (Kurose & Ross, 2010: 788).

In contrast, a trap message is an unsolicited message which is raised in specific situations. This kind of message is generated by a SNMP agent instead an entity, as in the first case (Kurose & Ross, 2010: 789).

## **3. The “presentation layer” in the Internet Protocol Stack**

In a deeper document review, everything indicates that there is no presentation layer defined in the Internet protocol stack. Following this principle, the different architectures in terms of software and hardware of all devices connected to the Internet, can interact mostly due to the base functionality of the operating systems. This common, low-level OS functionality allows interaction between a heterogeneous range of connected machines; the rest of the “magic” can be accounted for by the functionality of the Internet Protocol Stack.

## **4. About BER Encoding**

Basic Encoding Rules (BER) are the set of techniques which identify the instances of objects within the ASN.1 definition language. In addition, BER uses a triplet defined by type, length and value for encoding data in transmissions (Kurose & Ross, 2010: 798). According to the example shown in the textbook, the final sequence of the instance of data type *{weight, 277}* *{lastname,*

“*greg*”} could be represented as follows: (*277 in decimal is represented as 00000001 00010101 in binary notation*)

	↓	↓
	1	21
BER = 12122gerg45		

### **3. WEB APPLICATIONS**

#### **3.1. REFLECTIONS**

##### **3.1.1. Semantic Web**

The idea of a “web of data” as opposed to a “web of documents” has been put forward by the World Wide Web Consortium. Its main purpose is to enable computers to carry out more “intelligent” tasks, associating resources with meaning (W3C, n.d.). This proposal has been advanced as an alternative to the current approach, the traditional web that consists purely of readable documents. In other words, the web currently functions as a massive document container which people can access, and computers merely store and publish documents.

The idea revolves around taking advantage of machines’ potential computing resources, in terms of semantics. Unfortunately, traditional web pages are supposed to be interpreted exclusively by humans, and as such, computing potential is being wasted because computers cannot understand the web pages’ content. This is where the semantic web comes in.

The Semantic Web is based on the principle of metadata management; where metadata expresses how complex information is related. Using metadata –data about data– it is possible to establish a more meaningful understanding of the relationships between information (Evjen et al, 2007: 757). When it comes to the “quest” for meaningful content, “semantic search is the ultimate goal for search engines” (Derrick, 2012: 9). The Semantic Web grants information systems the use of this feature, allowing them to search for content employing an almost human understanding, with a special interest in the relationships between different content.

To accomplish this it was necessary to create new ways to represent content (or knowledge) in XML style. The Resource Description Framework –RDF– allows the representation of complex relationships, creating a meaningful structure for content. Semantics are built using the definition of resources in RDF; a resource is anything available on the Web, which must then be identified by a URI. A full description contains resources (objects and subjects), and their respective relationships (predicates) that give the structure meaning (Allemang & Hendler, 2008: 31). This, a triple, (subject, predicate, and object) is the core which forms a complex semantic.

The current World Wide Web was thought of as an efficient way to share information. However, the manner in which the information is shared is hypertext style, constructed by linking documents. This is the way in which the hypertext approach works (Allemang & Hendler, 2008: 12). In contrast, the semantic approach looks for a deeper consolidation of linked documents, in order to construct meaning via metadata. In this regard, the semantics-based approach is not so different from that of hypertext approach. In fact, semantics simply use the full potential of the information contained in the hypertext approach. The relationship between hypertext and semantics is a very close one. It can be argued that the development of the Semantic Web has been achieved thanks to the hypertext model; far from being opposites, the approaches actually complement each other.

### **Automated XML processing**

At a small public university in Southern Colombia, the students are taught how to construct computer-based solutions. As such, undergraduates in computer science are learning how to create general-purpose software. In parallel with this, the university itself needs software mainly related to academic-research, in addition to complementary software applications related to financial

status, assets management, human resources, and so on, but the main need for computing always appertains to academia.

Therefore, five subsystems have been designed: academic, financial, human resources, physical resources, and document management (Insuasti et al, 2008). Each subsystem is composed of modules; for instance, the Human Resources subsystem contains payroll, CV management, and welfare services modules. Finally, each module can be understood via the interaction of several computing applications. Most of those applications are web-based.

In the construction of each module, there are several technologies and platforms involved; most of the developed applications were created with Microsoft ASP.NET technology, some modules work with Oracle JSP/Servlet, and few modules were built from an open source perspective, such as PHP. All the information is managed using an Oracle 11g enterprise database. Within this heterogeneous network of development platforms, there are also XML technologies whose function it is to allow integration between subsystems and their modules. However, this feature is still under construction.

Fortunately, mission-critical subsystems –which are related to financial matters– work with XML Web Services, due to government policy concerning transparency, portability and scalability. At a national level, this kind of integration is also still under construction. The Colombian government requires specific reports in terms of financial status, with a view to making decisions about the economic support they give public universities. All these information systems are developed in Microsoft ASP.NET.

Taking these facts into account, automated XML processing via web-based technologies seem to be the most effective medium. More specifically, professional scenarios related to software

construction, such as Microsoft .NET platform and Oracle JSP/Servlet, are excellent candidates for making the most of the XML's potential. Indeed, legacy systems are practically extinct at the University of Nariño, (systems based on Informix 4GL and some programs written in PERL); they were all replaced with new web-based applications. This decision was taken in view of all too common issues related to the systems' capacity-of-growth; in fact, migrating legacy systems into a web-based versions with XML consumption is a hard task that requires skill and experience in software design (Tervo, Kilpeläinen, & Penttinen, 2006: 4).

### **Data Mining for the Web**

Since its inception, the World Wide Web has capitalized on the potential usage of Hypertext. In 1989, Tim Berners-Lee wrote a paper called, "Information Management: A Proposal" at CERN, in which he outlined that information could be stored in pages on computer networks, and not only that, but that those pages could be intricately linked, like a spider web. In fact, the name the "Web" was taken from this concept (Liu, 2011: 2). More than two decades have passed since the birth of the WWW, and ergo, the number of devices connected to the Web has increased almost exponentially. As a consequence, the amount of information available on the Web these days is practically immeasurable.

The Web is today the largest, publicly accessible data resource in the world, and its size is a product of accelerated growth in very a short lapse of time. Furthermore, the Web manages data of all existing types, and they are represented in several different ways including media technology. Described as such it seems chaotic, however, a key factor in its success is the fact that the information available on the Web is linked (Liu, 2011: 4).



The dynamic behavior of the Web in terms of content *per-sé*, can be analyzed by comparing it to technology related to knowledge discovery in databases (a.k.a. KDD) where data mining takes advantage of huge volumes of information deposited in data warehouses. This first feature, the vast amount of information available, is also true of the World Wide Web; however, it is important to note that the information on the Web is not always stored in databases. As such, another concept was born to address finding information on the web: Web mining.

Web mining's ultimate goal is to discover useful (non-trivial) information or knowledge using the linked structure, page content, and the usage of data (Liu, 2011: 7). To accomplish this, three types of mining exist: web structure mining, web content mining, and web usage mining. If a semantic-based initiative is required, all these kinds of web mining support the construction of meaning from the data sources. By applying data mining techniques to the Web, it is possible “to analyse the behaviour of individual users, access patterns of pages or sites, properties of collections of documents” (Berendt et al, 2004: 1).

Given this, the construction of the Semantic Web involves data mining, because such construction requires “intelligent” behaviors that involve the dynamic content available on the Web. In this vein, knowledge discovery implies the usage of techniques which give meaning to complex relationships based on web links (Berendt et al, 2004: 13). In conclusion, the benefits of using data mining for the Web and the Semantic Web are reciprocal; that is to say web mining takes advantage of the Semantic Web structures and *vice versa*. Nonetheless, it is important to recognize that such labors are not easy; that is why the Semantic Web is currently still under construction, in exciting scenarios such as Semantic Web Services, and Semantic Web Stores, among others (Hendler & Lassila, 2006).

### 3.1.2. DTD and XML Schema

As XML technologies work with data, and in accordance with predefined structures, there are different ways to give structures in a composition a meaning. This can be done by using DTDs, XML Schemas, and RELAX NG (Evjen et al, 2007: 135). Following is an exploration of the first two techniques.

Firstly, a DTD (Document Type Definition) was the first way to define data types, in order to give information structures meaning in XML. DTD uses a syntax derived from SGML which differs from the pure XML style. DTD is commonly used for validation tasks in e-commerce transactions; these transactions work with named elements and their attributes through relationship structures (Bean, 2003: 28).

Following this, XML Schemas were proposed by a group of members of the World Wide Web Consortium (W3C), as a stronger alternative for giving definition and meaning to XML-based information. In this vein, XML Schemas were considered a flexible and scalable way to define metadata; at the same time, XML Schemas describe the content of an XML file by getting access to individual items using simple element identification. One of the major features related to the massive usage of XML Schemas is their XML-based syntax as, “they are self-describing and generally less cryptic than DTD” (Bean, 2003: 29).

To summarize, table 3.1 shows some differences between DTD and XML Schemas:

Criteria	DTD	XML Schema
Syntax	SGML	XML
Namespace support	Partially	Full
Reusability	Low level	High Level
Modularity	Low level	High Level

Schema evolution and Extensibility	Difficult	Easy
Embedded, Structured Self-Documentation	Not available	Available

Table 3.1 DTD and XML Schema comparison (Møller, & Schwartzbach, 2006: 112-114).

According to the above comparison, the tendency towards using XML Schema, instead of DTD, is a common denominator in programming scenarios. XML Schema provides easy scalability and extensibility, as well as the simplicity of XML's syntax. Professional development environments such as Microsoft Visual Studio have allowed the construction of complex solutions based on XML technology; the automatic generation of XML-based documents is a clear example of why XML Schema is preferable to DTD in terms of reusability.

### 3.1.3. Formatting data in XML technologies

Having glanced at the nature of XML in terms of information management, it must be recognized that data in conjunction with their data types and defined structures are at the core of this technology. XML was conceived mainly to enable information portability and data exchange in distributed systems, among other features. With this in mind, it stands to reason that XML technologies have been entirely focused on data; however, the way in which the data is represented is another key factor, which has been covered by XSLT.

XSLT is defined as a language whose purpose it is to represent XML-style information according to given templates and rules (Evjen et al, 2007: 241). With XSLT, it is possible to “read” a complete structure of nodes within an XML document, this can be done by using *XPath*. In addition, the creation of templates is an efficient way to give XML documents a format. As XSLT

works under XML syntax, it is possible to extend functionalities in terms of the visual representation of information.

In order to integrate a visual aesthetic, XSLT incorporates stylesheets. An interesting aspect of formatting data through XSLT is the interaction with basic logic; that is to say that flow control structures within the language are available as conditionals and loops. Variable management is also supported. In addition, the output generated with XSLT can be defined in several formats, including XML itself, XHTML, and RSS (Evjen et al, 2007: 268).

On the other hand, and according to the World Wide Web Consortium, CSS is a standard which gives style and presentation to HTML documents. As such, CSS (Cascade Style Sheets) work like a language in order to represent information on a web page. (W3C, 2012). It is a strategy available for presentation, which works directly on web pages. There are some differences between XSLT and CSS; for instance, the former uses a XML-based syntax, whereas the latter uses another notation. Indeed, this is beneficial a feature of XSLT in terms of flexibility and reusability (Tennison, 200: 460).

One thing both XSLT and CSS have in common is the possibility of self-generation using today's development tools. Both XSLT and CSS can be generated from the creation of previous "objects", this means that HTML pages and XML documents must be present before the creation of their style descriptors.

In conclusion, CSS and XSLT have the same aim: visual presentation. However, they work using different syntaxes, and there are also some differences in their final applications. It can be said however, as of many technologies, that they are not rivals, but that actually they complement each other.

#### **3.1.4. Giving hierarchical structure to data with XML**

XML provides great flexibility when defining information through content and structure. People working with XML have the freedom to decide which data will be managed and used. Besides this freedom, XML-based standards work with a set of rules which allow the manipulation of vocabulary, which allows us to define practically everything (Evjen et al, 2007: 135).

This feature, being able to give hierarchical structure to data, is also multipurpose as data can be organized in lists, tables and graphs. This last structure is widely applied in the field of semantics and knowledge representation. Graphs are essential in many scenarios in which complex information representation is required.

The authors had the opportunity to develop a software application based on Dr. Joseph Novak's theory of concept maps. In 2007, they worked with Computer Science undergraduates to construct a software application which worked using concept maps in the form of a game. The test group were children between the ages of 7 and 10, and the intention was to implicitly teach several topics, including biology, history and maths, using the concept maps. The software won Microsoft's Imagine Cup 2007 at its local stage in Colombia.

According to their creator, "concept maps are graphical tools for organizing and representing knowledge. They include concepts, usually enclosed in circles or boxes of some type, and relationships between concepts indicated by a connecting line linking two concepts" (Novak, 2002: 548). Our software allowed the construction of concept maps due to XML, as we were able to store the list of concepts and the list of relationships. As such, the children's work was saved so

that it could be reviewed by their teachers. In terms of computing, the graphic interface merely represented the elements within the XML files in the format of a game.

This is an excellent example in which XML files are a useful computing solution. Success, in our case, came from being able to deal with textual information and give it a graphic and game-based appearance.

On the other hand, another solution was created last year in which cardiac signals were transferred via the Internet. This project worked using embedded computing to capture the analog cardiac signals using special sensors; then the signals were digitalized into a bits stream. According to the size of those bit streams, the signals were transferred as embedded, serialized objects using Microsoft Windows Communication Foundation, instead of XML Web Services.

There are some kinds of information which do not have a formal structure, in the sense of a graphic representation with traditional data structures, such as stacks, queues, lists, matrices, trees, or graphs. This is so in the case of bit streams, where some “parsers” are required to transform the “raw” information to make it readable. In those cases, depending on the size of the information, we must look for other alternatives for managing information other than using XML-based technologies.

### **3.1.5. Securing SOAP**

These days, aspects of XML Web Services now have enhanced security, and one of their major achievements in this regard is the WS-Security initiative (Evjen et al, 2007: 657). The

current specification of WS-Security is that it is used in contexts where different types of threats could affect the messages' reliability, confidentiality, and service availability (W3C, 2004).

To fulfil this specification, the use of authentication mechanisms and certificates are required. The scenario is one in which protocols and platforms support these activities, and various technologies including LDAP, RADIUS, Kerberos, and PKI, are candidates for inclusion in SOAP-based solutions under security policies. In spite of all the alternatives, in order to secure communications in SOAP environments, SOAP packets travel over HTTP and TCP connections (Evjen et al, 2007: 657). In technical terms, HTTP works through port 80 by default, which raises the following concerns.

Attacks have taken place as a result of port 80's vulnerabilities in HTTP-based service. In the case of the Code Red worms, a family of attackers used to send HTTP requests causing buffers' to overflow in Microsoft's Internet Information Services, the HTTP Server created by Microsoft Corporation (Somogyi & Schneier, 2001: 168). These attacks happened in 2001, and fortunately Microsoft Corporation solved the problem with a service pack. However, as a consequence, alarms were triggered in the computing industry concerning the vulnerability of the HTTP-based communications through port 80.

Nowadays, there are several ways to protect communications over the TCP/IP stack and the HTTP environment. One way to secure the transit of information on the Internet is IPSec. This protocol promotes the IP-based packing of information within security parameters. IPSec works with the network layer, establishing a direct relationship with the application layer (Tiller, 2001: 38). Figure 3.1, illustrates the location of IPSec in front of the OSI stack:

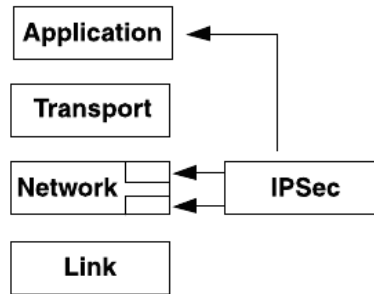


Figure 3.1 IPsec in the TCP Stack (Tiller, 2001: 38)

In this context, the use of port 80 over IPsec environments provides more security when software applications exchange their data. As port 80 is the default port for the World Wide Web publishing services in servers, “IKE –Internet Key Exchange– was a protocol for negotiation and authentication of IPsec-based virtual private networks” (Seltzer, 2005: 80). Now, there is a new version called IKEv2 (a.k.a. RFC-5996) which offers a scalable way to establish a shared state between the source and the sink of an IP datagram (Kaufman et al, 2010: 4). For better or for worse, SOAP uses the HTTP layer and therefore uses the TCP/IP stack as well. With this in mind, security on those basic levels is what matters most, in conjunction with the inner security based on WS-Security initiative.

### 3.1.6. SOAP in a small public university

In the context of our workplace, a small public university in Southern Colombia, XML Web Services are working with the financial subsystems (Insuasti et al, 2008: 45). More specifically, the financial subsystem contains the following modules: budget, treasury, portfolio, and accounting according to the Figure 3.2:



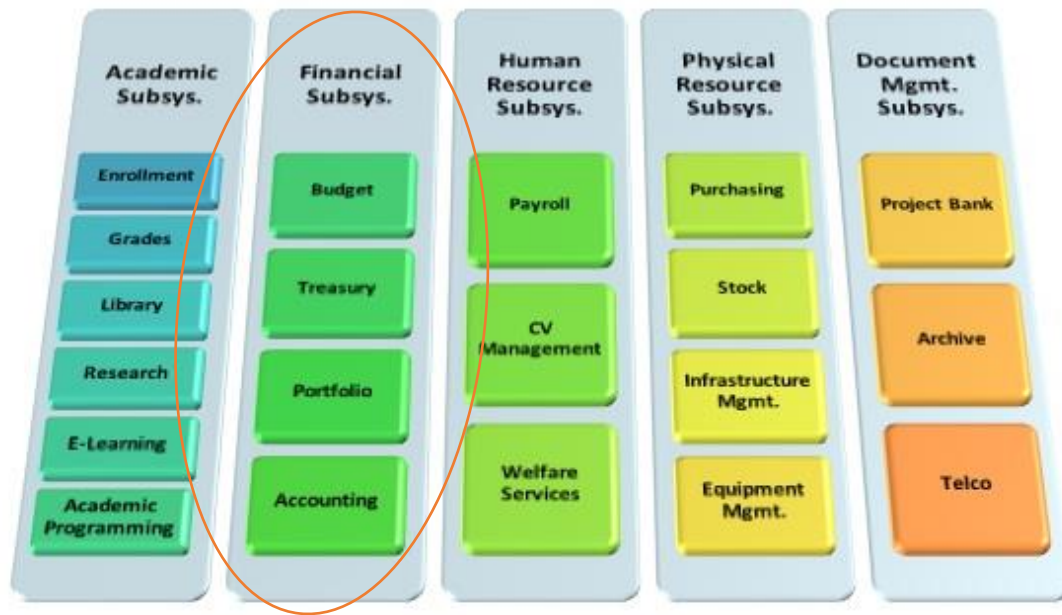


Figure 3.2 Integrated Information System at the University of Nariño (Insuasti et al, 2008: 43).

The Treasury module manages all the information related to financial status and the hierarchical distribution of bank accounts. Those bank accounts are owned by the University of Nariño, which manages a lot of public money from two main sources: the Colombian Government (95%) and undergraduate/graduate students' tuitions (5%).

Every single payment is done via bank transaction. Students must deposit their tuition into the University's bank account. Unfortunately, at this time our university does not provide the alternative of online payment. Therefore, the Treasury module consumes a set of XML Web Services provided by Colombian banks to obtain information about bank transactions. Such communication is done via SOAP layers over the HTTP base. It is important to note that the providers, in this case Colombian banks, request the creation of VPNs between them and their customers, for instance the University, in order to secure the information transfer.

HTTP is therefore still used as the base protocol that supports SOAP-based developments. Aspects related to security are supported by the providers: Colombian banks. HTTPS is used with certificates, and in addition, XML Web Services are configured to accept only predefined consumers using static IPs belonging to specific Virtual Private Networks. Of course, all parts of the SOAP stack are provided with UDDI, WSDL, and Web Methods, (Evjen et al, 2007: 618). Fortunately, the University of Nariño has not experienced any kind of online attack on this infrastructure so far.

### **3.1.7. Some considerations on native XML databases**

RDBMSs have transformed the world of information management in terms of storage capacity, security and performance. Since the beginning, databases have been used in practically all contexts in which information is a key factor in the pursuit of the organizational goals. RDBMSs –Relational Database Management Systems– organize their information structures according to the universally recognized Codd’s Model, based on tables and their relationships. In this vein, each RDBMS producer uses a model within the structure of the inner engines, to put their products on the competitive market.

Along the evolutionary process of the construction of RDBMSs, producers have incorporated several features into the core of their database engines. These features include: indexing services, text processing, analysis services, report services, administrative tasks for tuning, log control, fail recovery, backups and migration activities. All these features were conceived in order to increase the performance within quality parameters. All of the engineering

labors in this regard impact directly on the product status of the most prestigious database engines; thus, the common denominator of most database engines is the relational model.

Historically, the relational model has had excellent results in performance when dealing with huge volumes of information. Today, the size of information in complex systems is measured in petabytes, and even exabytes. With this in mind, information containers are mostly database engines working within the object-relational paradigm. These huge containers provide the computing resources to construct data warehouses which enable knowledge discovery using business intelligence techniques. In this context, performance is what matters most when dealing with enormous quantities of information.

Initially, XML technology was coupled with relational database engines, in order to manage XML-based documents as new data types as tables. Storing XML was an alternative way to handle large documents written in XML syntax. This allows the usage of XML technology with databases, however, the relational paradigm is still a work in progress (Evjen et al, 2007: 345).

In contrast, native XML databases are different in terms of storage; they manipulate XML in a more native way. There are lots of examples of native XML databases, including Oracle Berkeley DB XML, Xindice and BaseX, among others. In general terms, the salient features of native XML databases include: the document-based/node-based storage, accessibility through XQuery/XPath techniques, the use of different physical storage methods, and their independence from development platforms (Bourret, 2010).

In spite of the set of benefits related to the “pure” usage of XML-based data, native XML databases have some issues with performance and the implementation of additional services, which are common in RDBMSs. As such, several proposals are being developed to deal with problems

associated with concurrency control, response time of information retrieval, and incorporation of mining techniques (Haw & Lee, 2011: 1317). The use of native XML databases is suitable for cases in which the information unit is XML documents. In business environments, the marked preference is towards object-relational models, due to its performance features. However, native XML databases can be used for managing information about search and retrieval of data within the usage of semantics. In this regard, an interesting experience has been described by Cornish (2004: 184), where native XML databases were used to integrate hybrid systems by exploiting TextML techniques.

### **3.2. THE MAKING OF AN XML SCHEMA**

Historically, the traditional way to define structure in XML documents is by using DTD; Indeed, DTD is part of the original specification of XML technology in the World Wide Web Consortium (Evjen et al, 2007: 169). Some of DTD's features, such as its non XML-based syntax, low level of scalability, and limited vocabulary, among others, have left the path clear for alternative developments in the field of defining structure and validating XML documents; this is so in the case of XML Schemas.

In the making of an XML Schema, a file is produced alongside the XML document and the new file has the characters *xsd* as its extension. For instance, in a database, a simple table about countries can be represented in XML format; this table is defined by the ID and the country's name. Figure 3.3 shows the table's structure and its sample data.

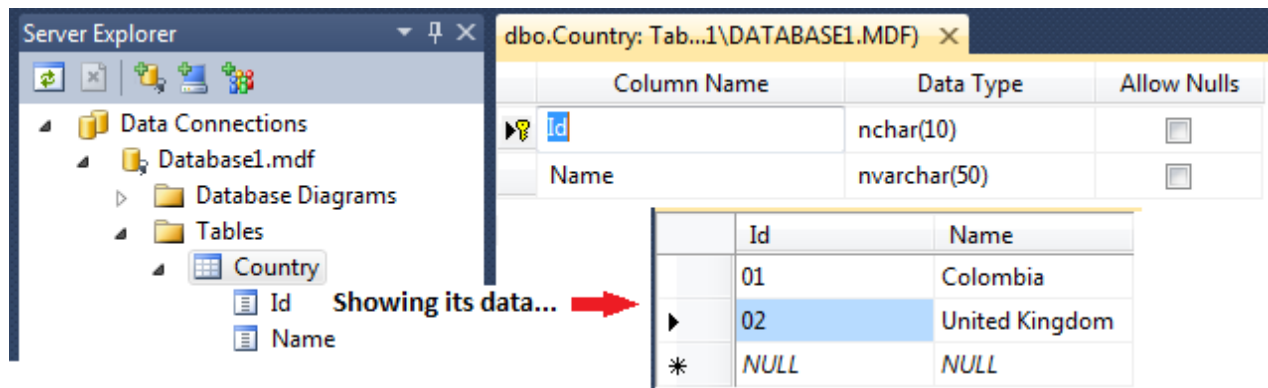


Figure 3.3 Structure and data of the table *Country* (Created with Microsoft Visual Studio 2010)

Then, when the *Country* table is dragged and dropped over a XML Schema (an *xsd* file), a generated code is available with the detailed structure definition. Figure 3.4 shows this.

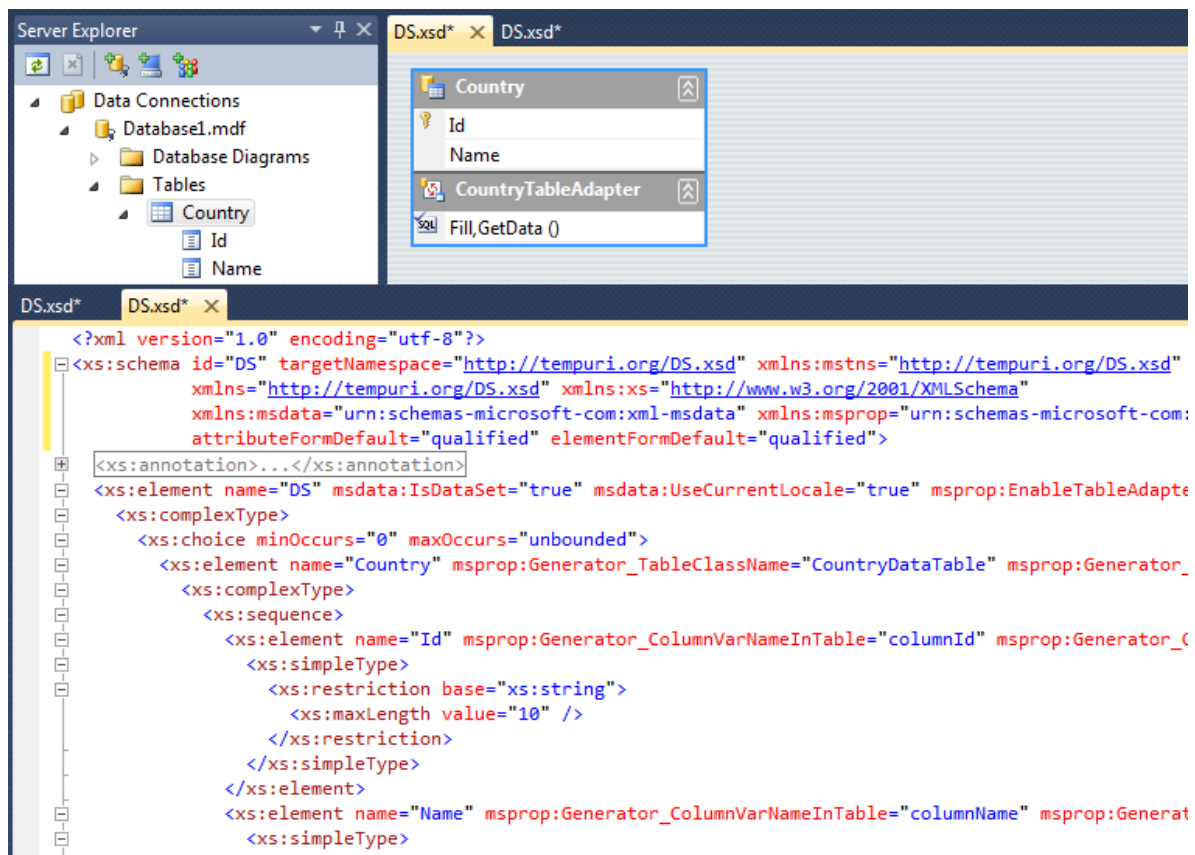


Figure 3.4 Generated code of a XML Schema (Created with Microsoft Visual Studio 2010)

Note that the structure definition, data types, referenced namespaces, and data themselves are part of the content of the *xsd* file. With this example, it can be seen that an XML Schema follows an XML-like syntax. This feature allows programmers to construct complex, scalable computing solutions easily. Returning to the example above, the *xsd* file represents a dataset expressed as an XML Schema; an *xsd* file such as this can be widely used by any application no matter the environment (console, desktop, mobile, web, cloud, etc.)

The advantages of using an XML Schema have been identified by Møller and Schwartzbach (2007: 114); and can be summarized as follows:

- XML-like syntax
- It allows the creation of user-defined data types
- It enables inheritance
- It supports evolution
- It permits embedded documentation

In conclusion, XML Schemas are very useful in the definition and validation of XML documents. In this sense, and in accordance with the example shown, the set of features of XML Schemas facilitate the actions of programmers in the construction of complex information systems. With powerful development tools, programmers are able to configure structures and data types easily; which is important in terms of productivity. Consequently, DTD is losing terrain. It seems that XML Schemas are eclipsing the use of DTD in professional scenarios.

### 3.2.1. UDDI's positioning

In a technological world where XML Web Services have been globally accepted and consumed, the development of computing solutions using heterogeneous environments –in terms of platforms, hardware, operating systems, and applications– has increasingly evolved. Taking into account the huge proliferation of SOAP-based technology, XML Web Services belong to long lists of providers, like in a telephone directory. Indeed, within the construction of an XML Web Service, it is possible to enable several inner services, with their respective web methods.

UDDI (Universal Description, Discovery, and Integration) was developed as a way to find XML Web Services on the World Wide Web. This development was caused by the common concern of how to access the specific location of a WSDL without knowing its URL (Ejven et al, 2007: 570). This sparked the need to develop discovery techniques, a quest in which giants of industry, such as Microsoft and IBM, were involved.

In one particular case, Microsoft's proposal on UDDI was established as a background service at the core of the Windows Server family, and over time, these services turned into a new technology called *BizTalk* by using a “new” framework for the resolution process of endpoints (MSDN, n.d.). UDDI is present in the general model of Web Services; it is the upper layer of the model stack according to figure 3.5.

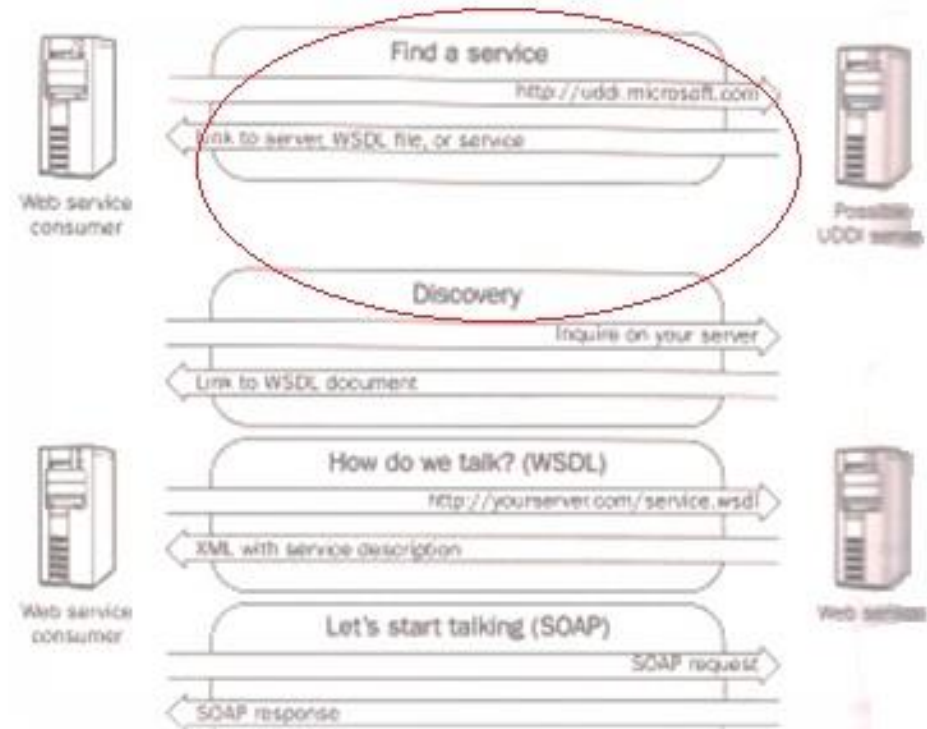


Figure 3.5 Web Service Model (Evjen et al, 2007: 608)

It is important to note that the usage of UDDI allows the categorization of XML Web Services, and at the same time, information about the services is efficiently managed, facilitating its discovery in connected environments (Januszewski, 2002). Today, UDDI is sought as a registry system with standard taxonomies in the description of available services, “UDDI began as ad hoc consortium; now housed at OASIS” (Sperberg-McQueen, 2003).

According to the above, UDDI is not an official standard promoted by the World Wide Web Consortium. Instead, OASIS Consortium, Organization for the Advancement of Structured Information Standards, is regulating the specification of UDDI through its active proposal. It requires the presence of UDDI providers, who have the registry which facilitates discovery tasks.



The controversy around UDDI relates to it being a monopoly. It is curious that since its inception, UDDI has not been considered as standard. Large registries, set up by several profit-based organizations, might promote the consumption of specific web services in accordance with the organizations' interests. This situation is quite complex, due to the commercial nature of some organizations which could be said to exploit people using their web services. This problem is very similar to that of controlling content on the Internet, in the sense that it is more a social-political (or even an economic) problem, than a technical one.

### **3.2.2. WSDL in the SOAP's World**

The Web Service Description Language –WSDL– is an XML-based language that works in the client/server environment, where SOAP messages are transported via the WSDL (Evjen et al, 2007: 607). Due to the proliferation of XML Web Services in machine-to-machine contexts, it was necessary to find ways to describe how to use these services and their functions. XML Web Services work like Web APIs, which can be consumed via a WWW platform (Yingzhou et al, 2010: 878).

SOAP, Simple Object Access Protocol, establishes the context for transferring information using an XML format. It provides the foundation for putting data in SOAP-based envelopes; consequently, the packaged data must follow strict structures according to predefined specifications. Ergo, the World Wide Web Consortium has clearly stated guidelines to support communication in heterogeneous environments via SOAP. Using these resources, the industry sector has developed their applications under the “consumption schema”, provided by SOAP and the XML Web Services initiative.

In order to promote the consumption of XML Web Services, a description from the client perspective of how to use the services was required. This was where WSDL appeared on the scene. Since its inception, WSDL was conceived as a structure closely related to SOAP-based solutions, and as such, it provides valuable information about the functionality of XML Web Services. In technical terms, the general structure of a common WSDL descriptor contains hierarchical elements that function as a formal service description. It includes: services, ports, bindings, port types, operations, messages and parts (Evjen et al, 2007: 638).

The close relationship between WSDL and SOAP was established clearly as a NOTE by the World Wide Web Consortium; this means that W3C suggest using it with a view to describing services (W3C, 2001). As a specification, WSDL has been considered *de facto* standard by the industry sector.

From a software development perspective, the use of XML Web Services has been almost universally accepted. WSDL is a common denominator in describing the functionality of those services. It is pretty easy to use and set up. WSDL has created a scenario, together with SOAP, in which XML Web Services can be used easily, thanks to well-defined descriptions, which is important as portability is crucial in our diverse world of technology.

Given this close relationship between WSDL and SOAP, an alternative way to describe XML Web Services could be assembled using semantics and ontologies. WSDL is powerful when it comes to description, but it is important to note that ontologies are also a powerful way to give meaning to conceptions. With this in mind, XML Web Services could be described using more than just functions, endpoints addresses, and parameters. Instead, we might think of describing

XML Web Services according to their functionalities in human terms, such as labor contracts for instance.

### **3.2.3. XML table-based mapping**

Databases have always been an integral part of the technology used by businesses in terms of information management. Over the last 40 years, since the inception of database technology, organizations have been able to develop in practically every field, including science, academia, business, entertainment and government, to name but a few. Everything started with Edgar Codd's famous model as a theoretical approach for organizing information using tables and relationships. The realization of his model generated a multibillion-dollar market. (Anthes, 2010: 18).

Production of RDBMS –Relational Database Management Systems– unveiled a new era of technological development in which information played a fundamental role within any organization. In the 1970s, the first RDBMSs were born following the Codd's model. Software development oriented to information systems had to interact with RDBMSs in several ways, such as mainframe environments, multiuser systems, client-server scenarios, and distributed systems among others. The common root –the Codd's model– has been used in the structure of databases, tables, columns and rows to manage information.

Over time, more requirements have been raised in line with people's expectations regarding information management; for instance, mobile devices such as smartphones, tablets, notebooks, and so on, are part of an interesting subgroup where computing capability is required in order to manage information. For these mobile scenarios, information systems working with databases have a great responsibility in terms of the integration, portability and reliability of the data that

they manage. Here, XML is presented as a globally accepted platform able to promote those features.

Given that the usage of XML-based platforms is an efficient technique for moving data into varied computing environments, most of the companies in charge of developing RDBMSs have seen in XML an excellent resource to facilitate the integration of systems oriented to information management. Several companies, including industry giants like Oracle, Microsoft, and IBM, have developed new data types with the purpose of supporting XML-style data. In addition, the inner infrastructure of their database engines has been powered by XML technology for tasks such as storage, importation, exportation, and even administrative activities (Evjen et al, 2007: 323). This is where XML table-based mapping is useful for administrative tasks.

In terms of information portability in cross-platforms, tasks related to data importation/exportation are made possible with the support of XML. In this regard, XML Web Services are an efficient computing resource that are widely used in distributed systems. Nowadays, practically all kinds of software applications are able to consume XML Web Services; these services can extract information from databases, thus client applications get the required information in XML format. However, the schema of the generated XML document is mostly a representation of information that does not necessarily match databases structures. This kind of information representation is not related to XML table-based mapping in the strictest sense; nonetheless, it is highly used in businesses, e-commerce and online transactions.

### **3.2.4. XML technologies and open standards**

XML technology originates from an older standard called SGML, which was coded as ISO 8879. In fact, XML technology was born from the need to work in web scenarios (W3C, 2010). Its predecessor, SGML, used markup language for data representation, but the World Wide Web Consortium saw the need to create a universal way to summarize information, in which data itself and data types could work together (Evjen et al, 2007: 6).

When it comes to technology adaptation's effect on the market, it has been said that "open standardization seeks to increase the size of the market" (Aggarwal, Dai & Walden, 2006: 117). In other words, open standards prevent competition arising between incompatible products and proprietary standards, and the associated costs. In this in particular there are opportunities for enterprises on the market to increase their number of customers, due to the wide availability of XML technologies.

An important benefit which relates to professional production of software under open standards, is the massive growth of potential users. An open standard breaks down the barrier of incompatibility, promoting a much higher consumption of technology. In this vein, companies in the software business can get more clients by taking advantage of the interest in data portability, flexibility, and universal information management.

The W3C has spent many years developing open standards for the web, adapting to new requirements and modern expectations. Thus, W3C recommendations include HTML, CSS, XHTML, XML, etc. Most browsers will pick up and implement these standards over their lifetimes to become more adept. The trend is to look for interoperability in different contexts, and in this

context, XML is an excellent platform which meets the expectations of companies in pursuit of profit.

### **3.2.5. XML Web Services and Cloud Computing**

Over the last few years, an important trend has emerged which is IT based on virtualization. The virtualization of computing resources is at the core of cloud computing (Baun et al, 2011: 5). This can be analyzed in terms of physical resources usage, management and consolidation, energy consumption and space required, among others. However, the most important aspect related to virtualization, is the increasing computing power needed for modern requirements.

Nowadays, cloud computing is used by several organizations to install their applications; as such, end users –clients– can consume those applications via the web. From a consumption perspective, most end users do not understand the difference between a cloud-based application and a web application hosted in a traditional manner.

In order to promote the proliferation of computing solutions, cloud-based applications take advantage of the web environment. In addition, potential clients that belong to mobile contexts can become consumers of a cloud-based solution. With this in mind, an important element is what is used to allow heterogeneous clients to consume applications on the cloud; this element is related to XML Web Services.

Taking into account the timeline in IT, XML Web Services were used before the existence of cloud computing. In addition, today's cloud computing solutions use XML web services with a view to taking advantage of mobile application potential in a wide variety of devices. For instance,

in cloud computing technology provided by Microsoft Corporation, applications, data, firewall rules, network and the operating systems are part of virtual machines within the context of Windows Azure. Thus, web applications created in Windows Azure are completed by interacting with XML web services. In this way, mobile clients can consume XML web services from a cloud computing source.

From the outset, XML Web Services have been an easy and extensible way to move datasets through the World Wide Web (Evjen et al, 2007: 565). Furthermore, a trend like cloud computing benefits from the characteristics of XML Web services. XML Web Services are already a strong base for wider trends. Nonetheless, they themselves are still evolving –like all kinds of technology– into much more reliable services in their security and performance. Evidence to support this idea was established by Tere, Jadhav and Mudholkar, in their study about the need to invoke web services in a dynamic way (2012: 79). A dynamic web service invocation framework (DWSIF) is needed where a UDDI proxy monitors the registries through private repositories and dispatches the services they find to the clients.

Taking all this into account, it can be safely asserted that XML Web Services will go on developing for a long time to come via trends such as cloud computing. It stands to reason that XML Web Services will evolve like all IT based technology. Perhaps there will be other techniques in the future that allow the exchange of large datasets in a better way than the current XML Web Services; however, XML Web Services will be in existence for a long time yet.

### 3.2.6. XML Web Services within a public university

The authors work for a small public university in southern Colombia. The main responsibility lies in teaching computer programming to undergraduates of computer science; however, they have also had the opportunity to create some computing solutions for the university administration.

The integrated information system of the university has five main areas: Academic Subsystem, Financial Subsystem, Human Resources Subsystem, Physical Resources Subsystem, and Document Management Subsystem. The distribution of functions is expressed in the figure 3.6

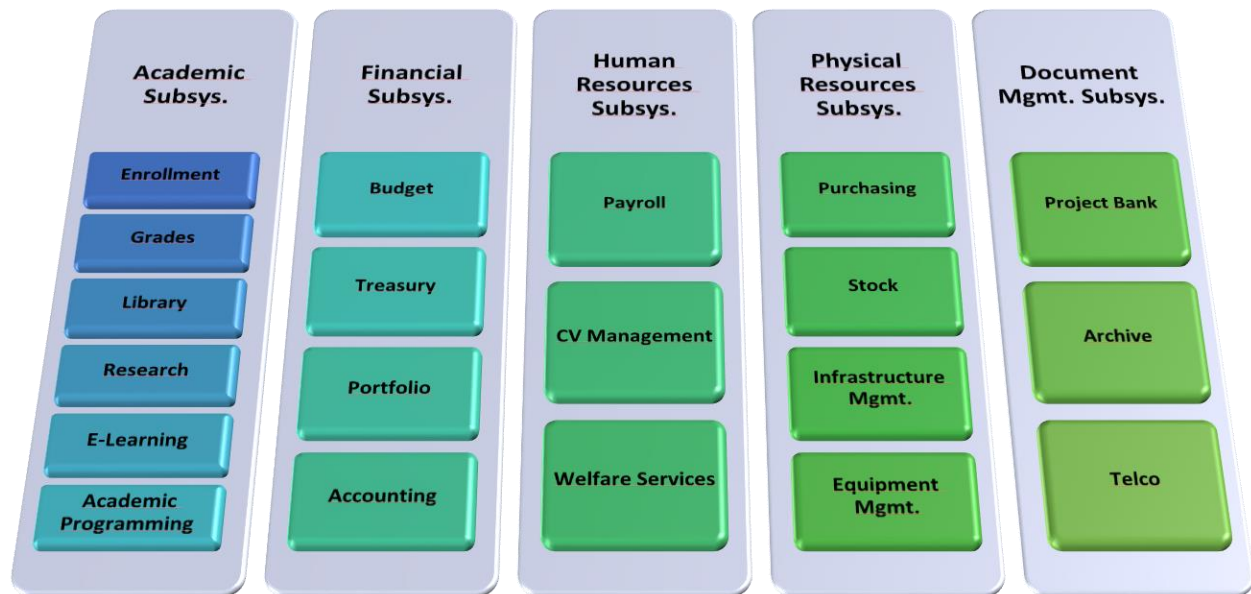


Figure 3.6 Integrated Information System at the University of Nariño (Insuasti et al, 2008).



In technical terms, most of the modules above have been developed as web applications using Microsoft .NET technology and Oracle Java technology; the main database engine is Oracle Database 11g and some PostgreSQL repositories. There are client-server environments for modules and some distributed architectures in the e-learning platform and the welfare services.

Some XML Web Services have been created with Microsoft Visual Studio 2010 Ultimate in order to support the access for mobile clients, and these modules are: Library, Research, Project bank, and Archive, among others. Unfortunately, communication with government authorities has not been integrated.

According to the regulations in Colombia, the Ministry of Education is the authority that rules the public system of higher education. Therefore, the Ministry of Education needs information from public universities in matters of student enrollment, rates of academic retention and academic dropout, coverage, social inclusion, etc. There are national information systems that manage this kind of data, but unfortunately the communications are not through XML Web Services. All the feedback is given via Microsoft Excel files.

As a result, the information is not always accurate. There are delays in updating the information of the entire nation, so it is impossible to see the current national situation in real time. With this in mind the creation of XML Web services is a key factor in keeping information up to date. However, this should obviously be a government initiative.

### 3.3. WORKSHOPS

#### 3.3.1. Hand-In Assignment 1

##### Exercise 1:

Microsoft Visual Studio 2010 Ultimate SP1 has been used here as an XML Editor. In order to check the syntax of the exercise's script, the following sequence is required: First, the creation of a new File from the dropdown main menu.

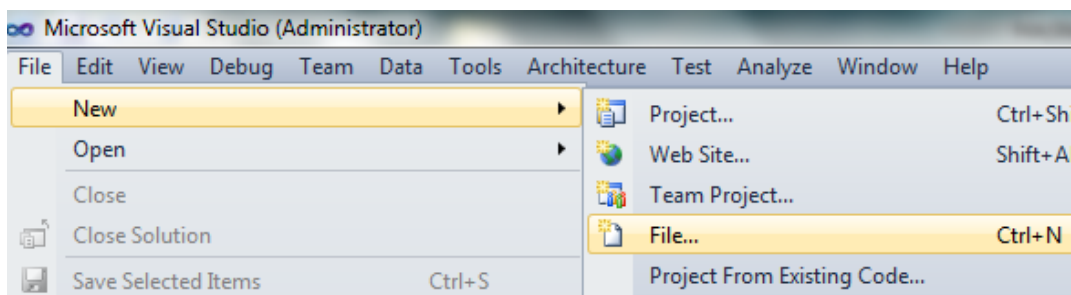


Figure 3.7 Then, an XML file is selected. Finally, Open is pressed.

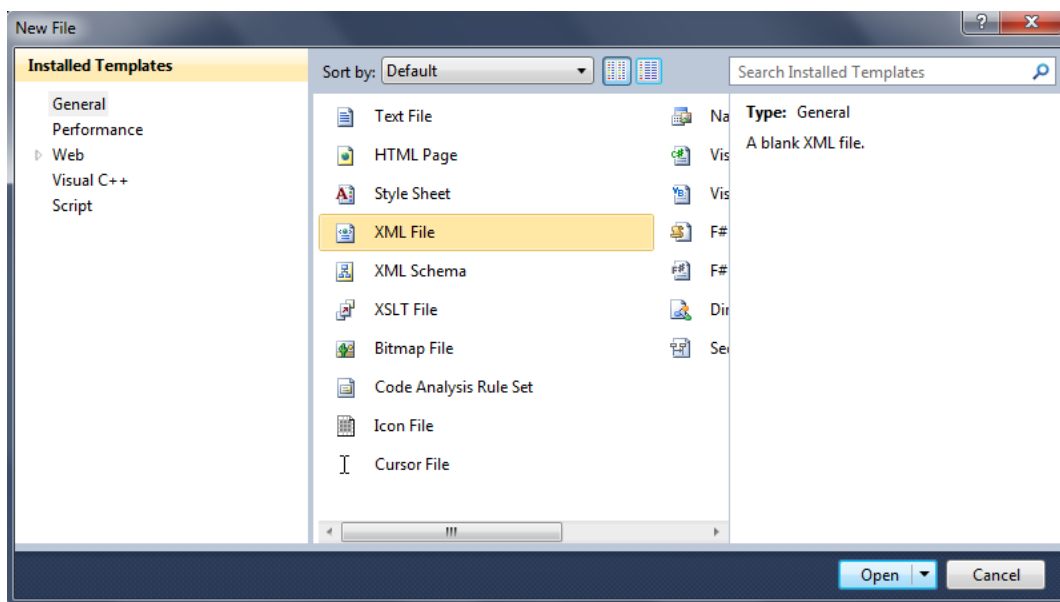


Figure 3.8 Creating a XML file

In the editor below is the exercise's code:

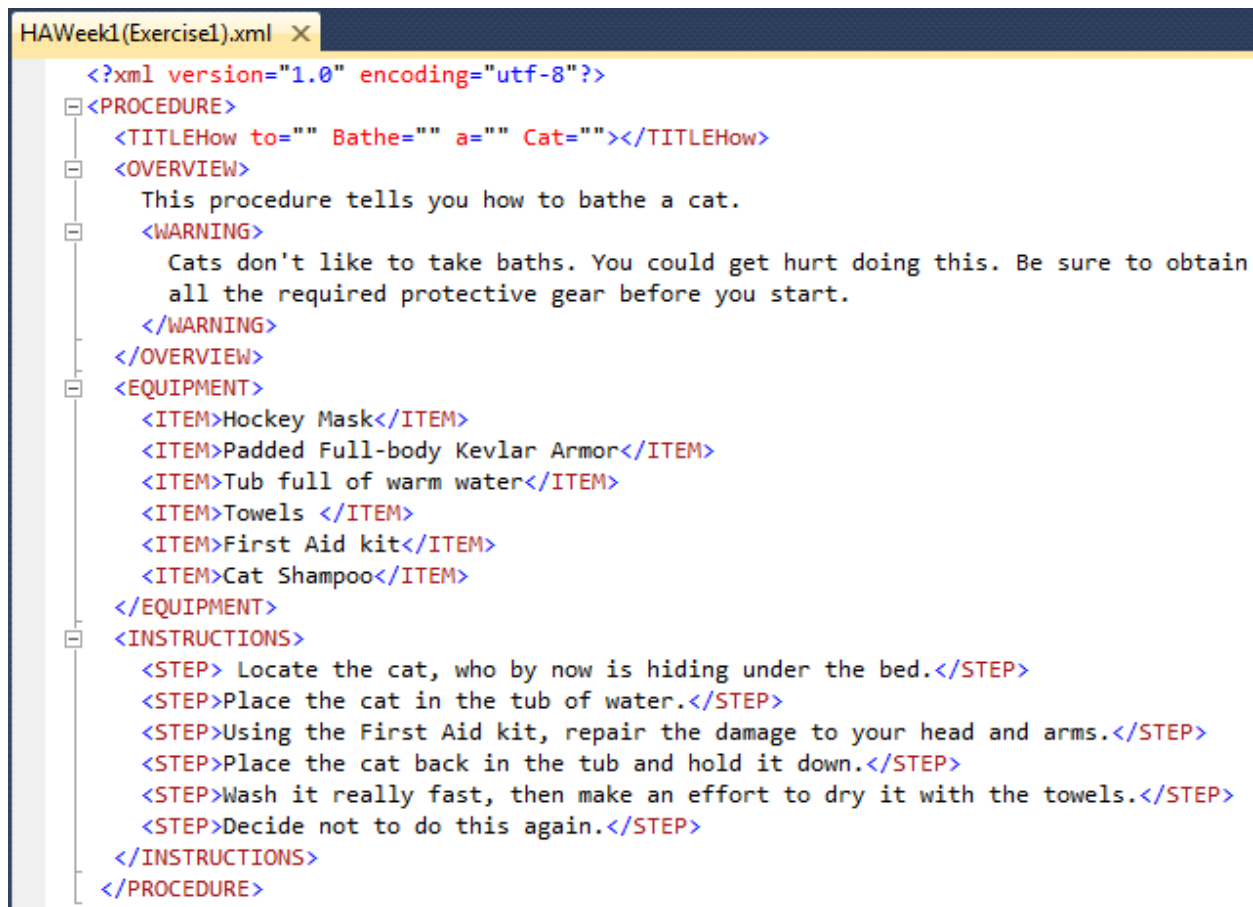
Figure 3.9 Editing a XML file

Obviously, the XML code above has several errors in terms of its structure. The following table shows the list of the errors found:

No	Description
1	The preprocessor directive is missing: <?xml version="1.0" encoding="utf-8"?>
2	The first element is not closed: </PROCEDURE> is missing
3	The closing symbol for <TITLEHow is missing
4	<WARNING> tag needs a closing tag or a unary end symbol like />
5	</WARNING> appears in the wrong place. There is another inner tag </OVERVIEW>
6	<EQUIPEMENT> perhaps is a typo. <EQUIPMENT> is more convenient
7	<ITEM> tag is required for the second item within the <EQUIPMENT> level
8	<ITEM> tag should be changed for <STEP> tag within the <INSTRUCTIONS> section

Table 3.2 Description of code

When the previous errors are corrected, the XML file takes the following, well-formed appearance:



```
<?xml version="1.0" encoding="utf-8"?>
<PROCEDURE>
  <TITLEHow to="" Bathe="" a="" Cat=""></TITLEHow>
  <OVERVIEW>
    This procedure tells you how to bathe a cat.
  </OVERVIEW>
  <WARNING>
    Cats don't like to take baths. You could get hurt doing this. Be sure to obtain
    all the required protective gear before you start.
  </WARNING>
  <EQUIPMENT>
    <ITEM>Hockey Mask</ITEM>
    <ITEM>Padded Full-body Kevlar Armor</ITEM>
    <ITEM>Tub full of warm water</ITEM>
    <ITEM>Towels </ITEM>
    <ITEM>First Aid kit</ITEM>
    <ITEM>Cat Shampoo</ITEM>
  </EQUIPMENT>
  <INSTRUCTIONS>
    <STEP> Locate the cat, who by now is hiding under the bed.</STEP>
    <STEP>Place the cat in the tub of water.</STEP>
    <STEP>Using the First Aid kit, repair the damage to your head and arms.</STEP>
    <STEP>Place the cat back in the tub and hold it down.</STEP>
    <STEP>Wash it really fast, then make an effort to dry it with the towels.</STEP>
    <STEP>Decide not to do this again.</STEP>
  </INSTRUCTIONS>
</PROCEDURE>
```

Figure 3.10 Complete code

This file is attached to this document as Exercise1 (Week2).xml

## **Exercise 2:**

In order to generate an XML file for this exercise, the database scheme in Microsoft SQL Server 2008 Express was reproduced as shown in the following figure. This database belongs to an ASP.NET empty project created in Microsoft Visual Studio 2010 with Microsoft .NET Framework 4.

This project will return the results of a query from the database scheme once its tables are filled with some data. To do that, a dataset item is added to the project in the solution explorer. This dataset item visually contains the full database scheme, which was inserted via drag and drop.

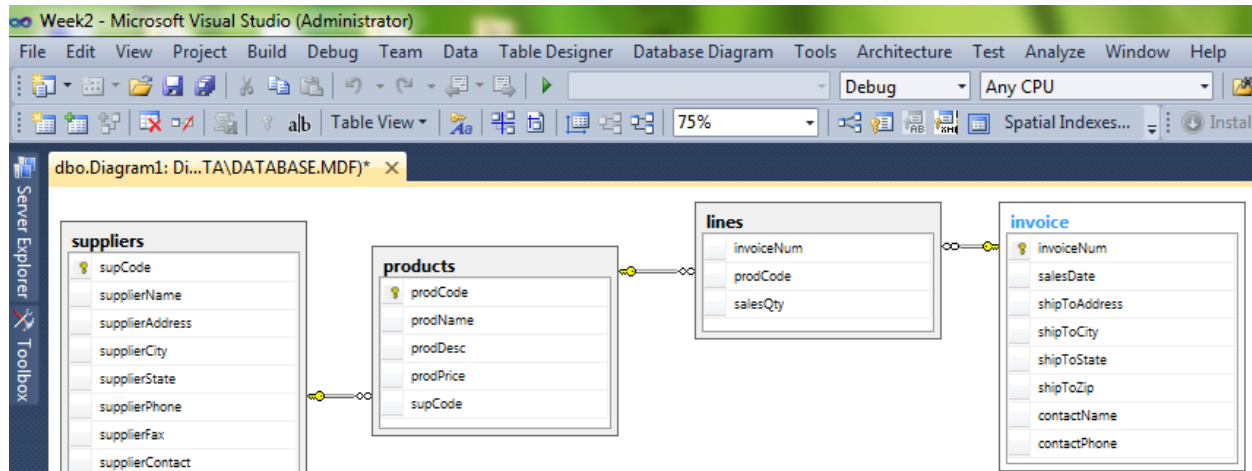


Figure 3.11 Entity-relation diagram

We created an SQL function as follows:

```
CREATE FUNCTION dbo.GetInvoice
(
)
RETURNS TABLE
AS
RETURN
(
SELECT      invoice.invoiceNum, invoice.salesDate, invoice.shipToAddress, invoice.shipToCity,
invoice.shipToState, invoice.shipToZip, invoice.contactName,
              invoice.contactPhone, products.prodCode, products.prodName, products.prodDesc,
products.prodPrice, suppliers.supCode, suppliers.supplierName,
              lines.salesQty
FROM        invoice INNER JOIN
              lines ON invoice.invoiceNum = lines.invoiceNum INNER JOIN
              products ON lines.prodCode = products.prodCode INNER JOIN
              suppliers ON products.supCode = suppliers.supCode
)
```

Then, the dataset uses this function in order to create a new inner object. The source code for the web service is shown below:

```
using System;
using System.Data;
using System.Web.Services;
using DSTableAdapters;

[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class _Default : System.Web.Services.WebService
{
    private GetInvoiceTableAdapter ta = null;
    private DS.GetInvoiceDataTable dt = null;
    //-----

    public _Default ()
    {
        ta = new GetInvoiceTableAdapter();
        dt = new DS.GetInvoiceDataTable();
    }
    //-----

    [WebMethod]
    public DS.GetInvoiceDataTable GetInvoices()
    {
        ta.Fill(dt);
        return (dt);
    }
    //-----
}
```

When the XML web service runs in a web browser, it shows the following XML-based information:



Figure 3.12 Generated web page

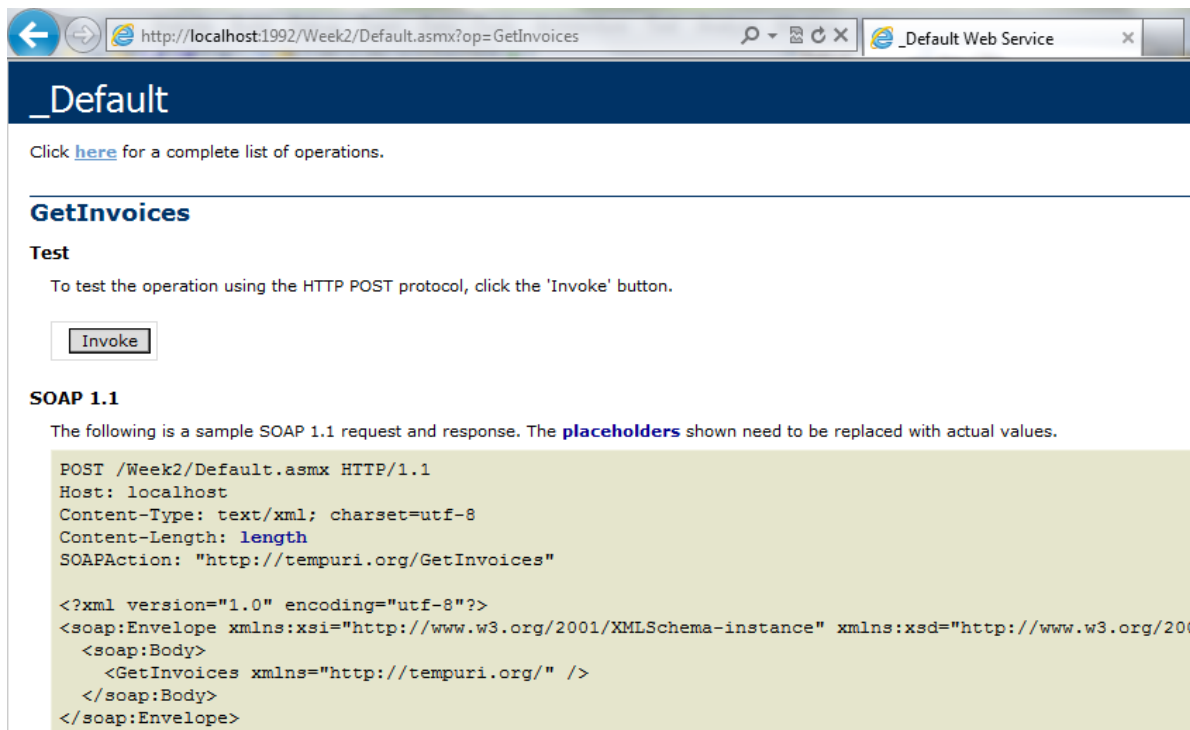
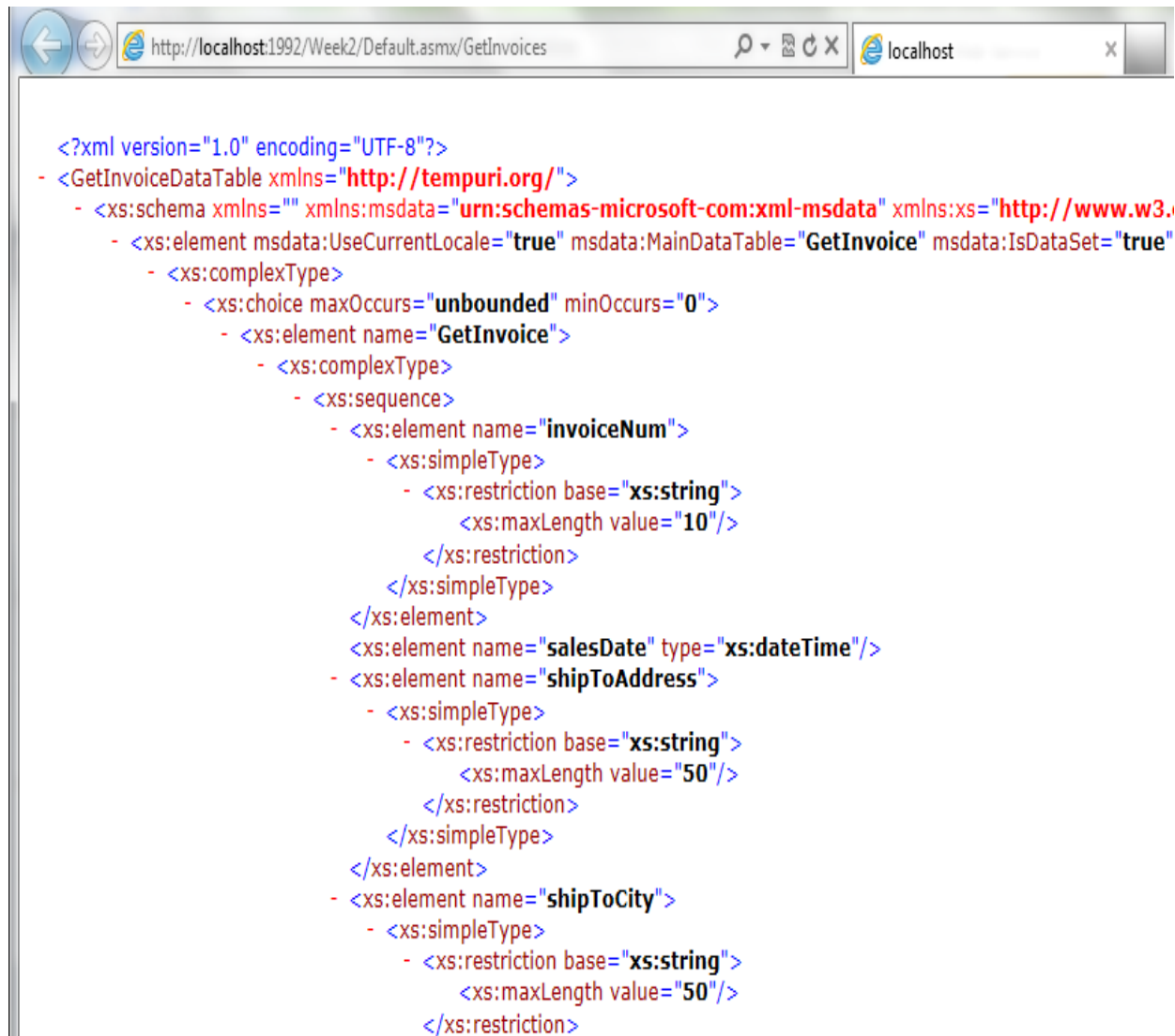


Figure 3.13 Invoking a web method

A screenshot of a web browser window. The address bar shows the URL 'http://localhost:1992/Week2/Default.aspx/GetInvoices'. The browser window displays an XML schema definition (XSD) for a web service. The schema is written in XML and includes various namespaces and elements. The root element is 'GetInvoiceData' with a namespace of 'http://tempuri.org/'. It contains a 'schema' element with a namespace of 'urn:schemas-microsoft-com:xml-msdata' and a 'choice' element with a namespace of 'xs:'. The 'choice' element contains three elements: 'invoiceNum' (a string with a maximum length of 10), 'salesDate' (a dateTime), and 'shipToAddress' (a string with a maximum length of 50). The 'shipToAddress' element is further defined by a 'restriction' element with a base of 'xs:string' and a maximum length of 50. The 'shipToCity' element is also defined by a 'restriction' element with a base of 'xs:string' and a maximum length of 50. The schema is written in XML and includes various namespaces and elements. The root element is 'GetInvoiceData' with a namespace of 'http://tempuri.org/'. It contains a 'schema' element with a namespace of 'urn:schemas-microsoft-com:xml-msdata' and a 'choice' element with a namespace of 'xs:'. The 'choice' element contains three elements: 'invoiceNum' (a string with a maximum length of 10), 'salesDate' (a dateTime), and 'shipToAddress' (a string with a maximum length of 50). The 'shipToAddress' element is further defined by a 'restriction' element with a base of 'xs:string' and a maximum length of 50. The 'shipToCity' element is also defined by a 'restriction' element with a base of 'xs:string' and a maximum length of 50.

```
<?xml version="1.0" encoding="UTF-8"?>
- <GetInvoiceDataTable xmlns="http://tempuri.org/">
  - <xs:schema xmlns="" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:xs="http://www.w3.org/2001/XMLSchema" msdata:UseCurrentLocale="true" msdata:MainDataTable="GetInvoice" msdata:IsDataSet="true">
    - <xs:complexType>
      - <xs:choice maxOccurs="unbounded" minOccurs="0">
        - <xs:element name="GetInvoice">
          - <xs:complexType>
            - <xs:sequence>
              - <xs:element name="invoiceNum">
                - <xs:simpleType>
                  - <xs:restriction base="xs:string">
                    <xs:maxLength value="10"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="salesDate" type="xs:dateTime"/>
            - <xs:element name="shipToAddress">
              - <xs:simpleType>
                - <xs:restriction base="xs:string">
                  <xs:maxLength value="50"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            - <xs:element name="shipToCity">
              - <xs:simpleType>
                - <xs:restriction base="xs:string">
                  <xs:maxLength value="50"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</GetInvoiceDataTable>
```

Figure 3.14 Results

The entire XML file is attached to this document as Exercise2 (Week2).xml

### **Exercise 3:**

As previously noted, the XML file above uses XML namespaces because it was generated by a query based on the creation of an XML web service.



The first namespace is `<GetInvoiceDataTable xmlns="http://tempuri.org/">`. In this case, all the information about GetInvoiceDataTable is extracted by a local solution (tempuri.org as default). This web method returns a special data type called DS.GetInvoiceDataTable which is defined by the dataset DS.

Alongside this, two more XML namespaces are used in this exercise:  
`xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"`  
`xmlns:xs="http://www.w3.org/2001/XMLSchema"`. These namespaces are defined by Microsoft and the World Wide Web Consortium respectively. The schema xs is used in the identification of each element within the XML file.

Finally, the complete solution is provided as a compressed zip file called Exercise3 (Week2). This file contains the folder for installing the XML Web Service. Microsoft SQL Server 2008 Express and Microsoft .NET Framework 4.0 are required. If you would like to try this XML Web Service, it is available online at <http://190.254.4.13/Week2/Default.aspx>

### **3.3.2. Hand-In Assignment 2**

#### **Exercise 1:**

In Microsoft Visual Studio 2010 Ultimate Edition, an ASP.NET empty project was created with the provided XML document. In this environment, unfortunately this IDE does not have a DTD file type so it is necessary to create a DTD manually since a text file. Figure 3.15 shows the creation process of a DTD for the sample XML file.

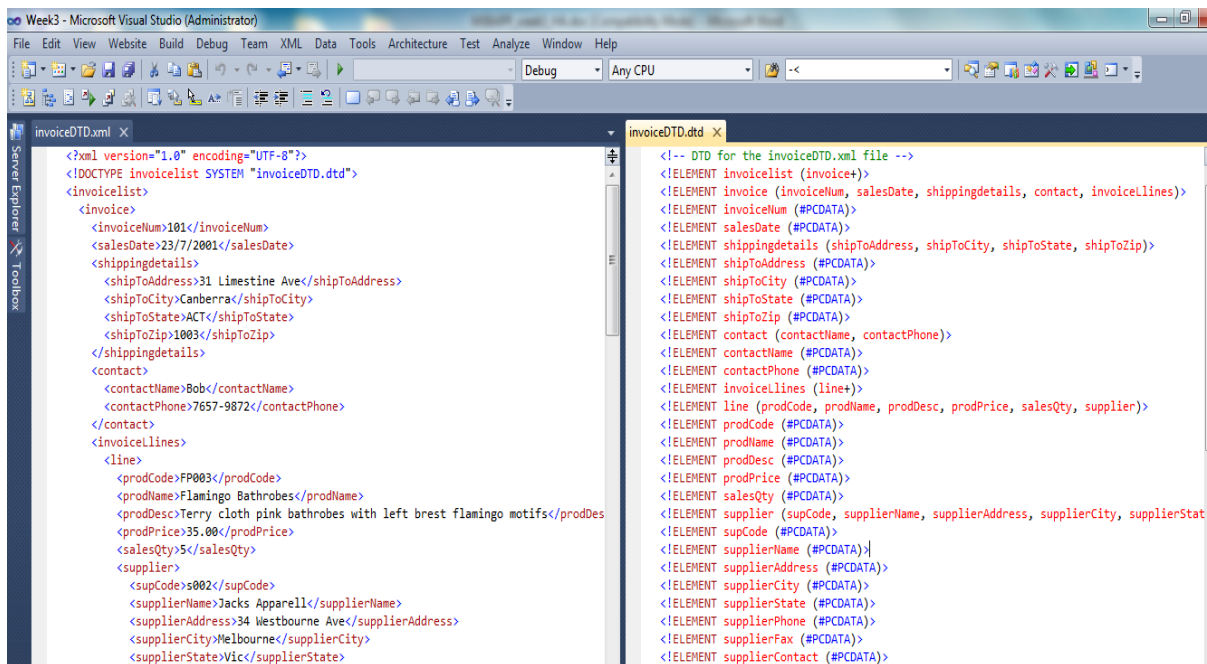


Figure 3.15 Manual creation of the DTD file according to the invoiceDTD.xml file.

## Exercise 2:

A XML Schema is fairly easy to create in Microsoft Visual Studio; indeed, it is auto-generated with an option of the XML menu when an XML document is opened. Figure 3.16 shows this process.

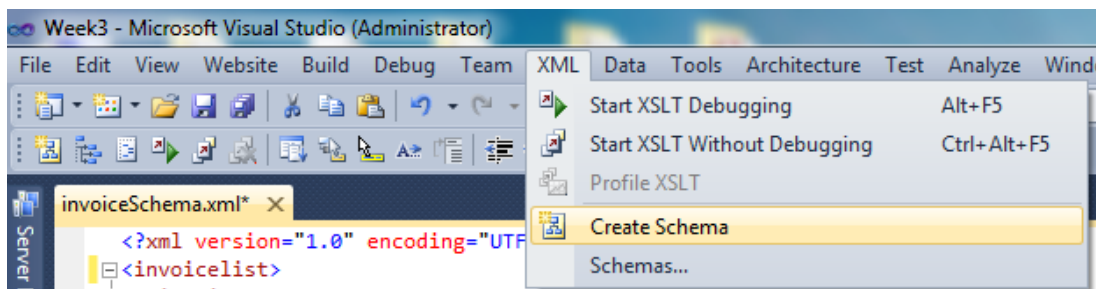


Figure 3.16 Creation of a XML Schema since an active XML document.

In doing this, an XML Schema file is created; it is called invoiceSchema.xsd. This new file contains the definition of a structure that fits with the original XML document. The XML Schema

provides the structure and the definition of data types for each element of the original XML document. This operation is shown in Figure 3.17.

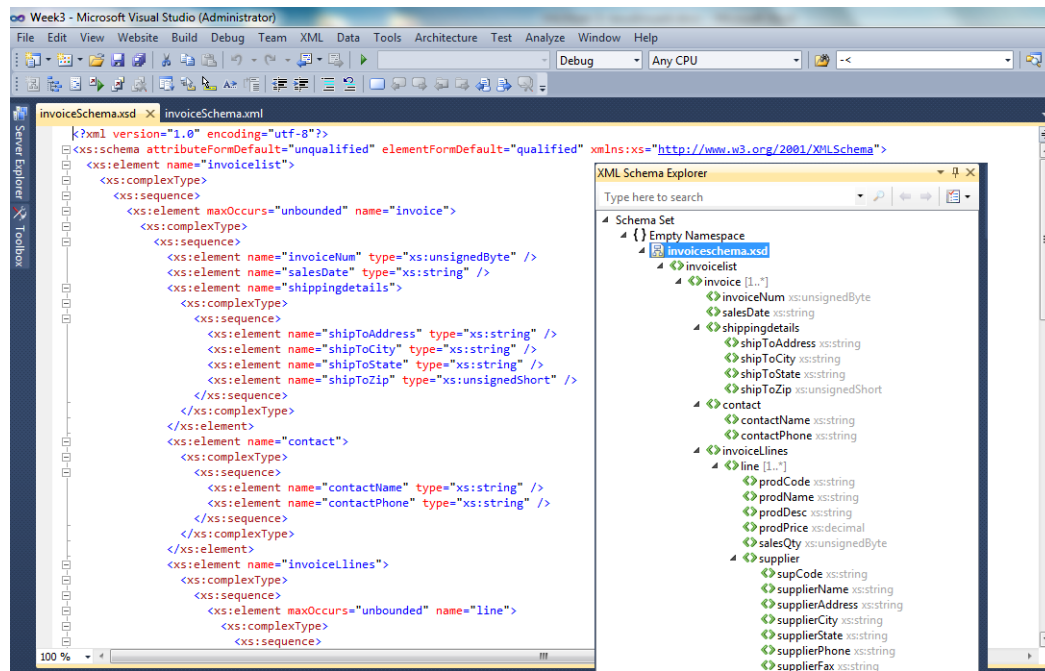


Figure 3.17 XML Schema created via Microsoft Visual Studio 2010.

### **Exercise 3:**

DSD is a schema language which gives XML documents structure. Its second specification was developed by the University of Aarhus and AT&T Labs Research. According to Møller and Schwartzbach (2006: 159), DSD has an XML-based syntax, it has Boolean logic and it manages regular expressions, declaration of rules and *declare/required* sections, among others. It is an interesting alternative way of giving XML documents structure; however, due to the massive popularity of XML Schemas, DSD usage is growing slowly.

DSD specification 2.0 is defined as a language (AT&T & BRICS, 2005). This language incorporates the following features: schema definition, rules, Boolean expressions, regular

expressions, references solver, normalization, uniqueness and pointers, among others. An interesting point about DSD2 is its syntax, which was based on Backus-Naur Forms for its grammar.

Another way to give XML documents structure is RELAX NG. This was proposed by OASIS –Organization for the Advancement of Structured Information Standards– then a standard was created by ISO (Møller & Schwartzbach, 2006: 168). RELAX NG aims to give simple management of syntax and powerful expressiveness. It is true that RELAX NG has the same goals as DSD2, however, their core concepts are quite different. RELAX NG is based on grammars, while DSD2 is based on rules.

In conclusion, all the techniques used to define structure in XML documents are similar; how they do it differs according to their specifications. In terms of usage, XML Schema comes first due to its massive consumption in diverse professional environments. In this vein, powerful companies such as Microsoft and Oracle among others use XML Schema as a default option when it comes to structure definition of information.

#### **Exercise 4:**

Unfortunately, as with the first exercise, the IDE is not able to validate an XML document through DSD. Ergo, a new file must be created manually. However, it is possible to validate a XML document through DSD via Java-based parsers (AT&T & BRICS, 2005). Figure 3.18 shows this process.

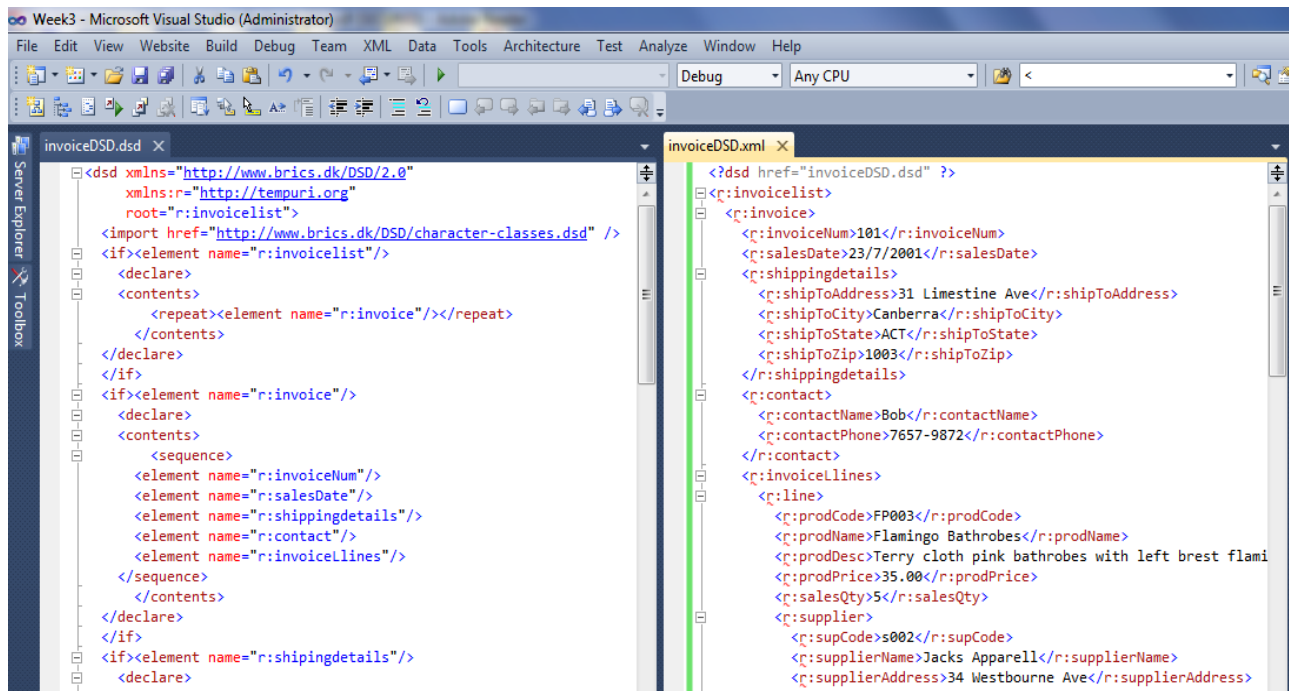


Figure 3.18 Application of DSD.

Note the XML document has validation problems because Microsoft Visual Studio 2010 does not recognize the schema definition. Finally, three compressed files are attached to this document: `ex1.zip`, `ex2.zip`, and `ex3.zip`. Each compressed file contains two files: the XML document and its scheme according to the exercises (dtd, xsd, and dsd).

### 3.3.3. Hand-In Assignment 3

#### Exercise 1:

For the XML document given, a HTML page was created to scan the document's structure, including nodes, values and attributes. This web page requires the usage of JavaScript in order to produce the structure. Its code is shown below:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
</head>
<body>
<script type="text/javascript">
    var xmlDoc = new ActiveXObject("Microsoft.XMLDOM");
    var tab = "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&";
    //-----
    function printElement(indent, node)
    {
        var i;
        var j;
        var attr;
        //
        if (node.nodeType == 3)
        {
            document.write("<br />" + indent + node.nodeValue);
        }
        else
        {
            document.write("<br />" + indent + "[" + node.nodeName + "]");
            //---showing attributes
            attr = null;
            attr = node.attributes;
            if (attr != null)
            {
                for (j = 0; j < attr.length; j++)
                {
                    document.write("<b> Attribute: " + attr[j].name + " = " + attr[j].value + ";</b>");
                }
            }
        }
    }
</script>

```

```

//-----
for (i = 0; i < node.childNodes.length; i++)
{
    printElement(indent + tab, node.childNodes[i]);
}
document.write("<br />" + indent + "[" + node.nodeName + "]");
}
}
//-----

xmlDoc.async="false";
xmlDoc.load("./orders.xml");
if (xmlDoc.parseError.errorCode != 0)
{
    document.write("<br />Error Code: ");
    document.write(xmlDoc.parseError.errorCode);
    document.write("<br />Error Reason: ");
    document.write(xmlDoc.parseError.reason);
    document.write("<br />Error Line: ");
    document.write(xmlDoc.parseError.line);
    document.write("<br />");
}
else
{
    document.write("<strong>(Week4) Exercise 1 - Jesus Insuasti</strong><br />");
    printElement("", xmlDoc.documentElement);
}
</script>
</body>
</html>

```

This code was based on Evjen et al's book (2007: 367). When this HTML page is opened in a web browser, it produces the following output:

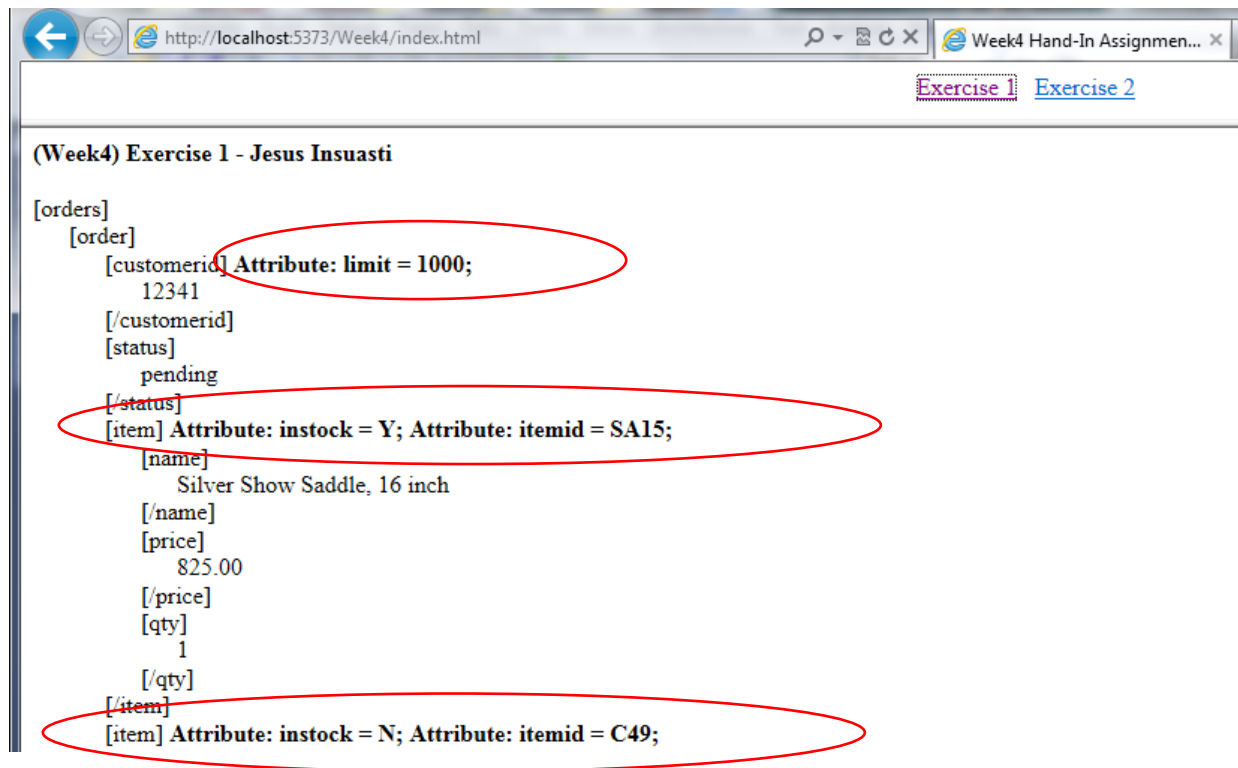


Figure 3.19 Getting the XML hierarchy

The structure shows the XML hierarchy through nodes and their values. In addition, it presents the information about the nodes' attributes with their respective values. This exercise is contained in the compressed project that was attached to this document; the HTML page is called *exercise1.html*

### **Exercise 2:**

For this exercise, the following code was created to produce a normal output from the *orders2.xml* file; and then, a new order is added by scripting. It is important to note that the function which shows the entire XML structured is reused.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```



```
<title></title>  
</head>  
<body>  
    <script type="text/javascript">  
        xmlDoc = new ActiveXObject("Microsoft.XMLDOM");  
        tab = "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&";  
        xmlDoc.async = "false";  
        xmlDoc.load("./orders2.xml");  
        if (xmlDoc.parseError.errorCode != 0)  
        {  
            document.write("<br />Error Code: ");  
            document.write(xmlDoc.parseError.errorCode);  
            document.write("<br />Error Reason: ");  
            document.write(xmlDoc.parseError.reason);  
            document.write("<br />Error Line: ");  
            document.write(xmlDoc.parseError.line);  
            document.write("<br />");  
        }  
        else  
        {  
            document.write("<strong>(Week4) Exercise 2 - Jesus Insuasti</strong><br />");  
            document.write("<hr />");  
            document.write("<b>Initializing Structure...</b><br />");  
            printElement("", xmlDoc.documentElement);  
            document.write("<hr />");  
            document.write("<b>Adding elements into the Structure...</b><br />");  
            var orderElement = xmlDoc.createElement("order");  
            var customeridElement = xmlDoc.createElement("customerid");  
            customeridElement.text = "012345";  
            customeridElement.setAttribute("limit", "9000");  
            var statusElement = xmlDoc.createElement("status");  
            statusElement.text = "done";  
            var itemElement = xmlDoc.createElement("item");  
            itemElement.setAttribute("instock", "Y");  
            itemElement.setAttribute("itemid", "JHIP");  
            var nameElement = xmlDoc.createElement("name");  
            nameElement.text = "Jesus-style sport shoes";  
            var priceElement = xmlDoc.createElement("price");  
            priceElement.text = "345";  
            var qtyElement = xmlDoc.createElement("qty");  
            qtyElement.text = "100";  
            orderElement.appendChild(customeridElement);  
            orderElement.appendChild(statusElement);  
            itemElement.appendChild(nameElement);  
            itemElement.appendChild(priceElement);  
            itemElement.appendChild(qtyElement);  
            orderElement.appendChild(itemElement);  
            xmlDoc.documentElement.appendChild(orderElement);  
            printElement("", xmlDoc.documentElement);
```

```

}
//-----
function printElement(indent, node)
{
    var i;
    var j;
    var attr;
    if (node.nodeType == 3)
    {
        document.write("<br />" + indent + node.nodeValue);
    }
    else
    {
        document.write("<br />" + indent + "[" + node.nodeName + "]");
        //----showing attributes
        attr = null;
        attr = node.attributes;
        if (attr != null)
        {
            for (j = 0; j < attr.length; j++)
            {
                document.write("<b> Attribute: " + attr[j].name + " = " + attr[j].value + "</b>");
            }
        }
        //-----
        for (i = 0; i < node.childNodes.length; i++)
        {
            printElement(indent + tab, node.childNodes[i]);
        }
        document.write("<br />" + indent + "/" + node.nodeName + "]");
    }
}
//-----
</script>
</body>
</html>

```

The output of this HTML page is shown as follows. As in the case of the first exercise, this second exercise is written into a HTML page called *exercise2.html* within the solution project.

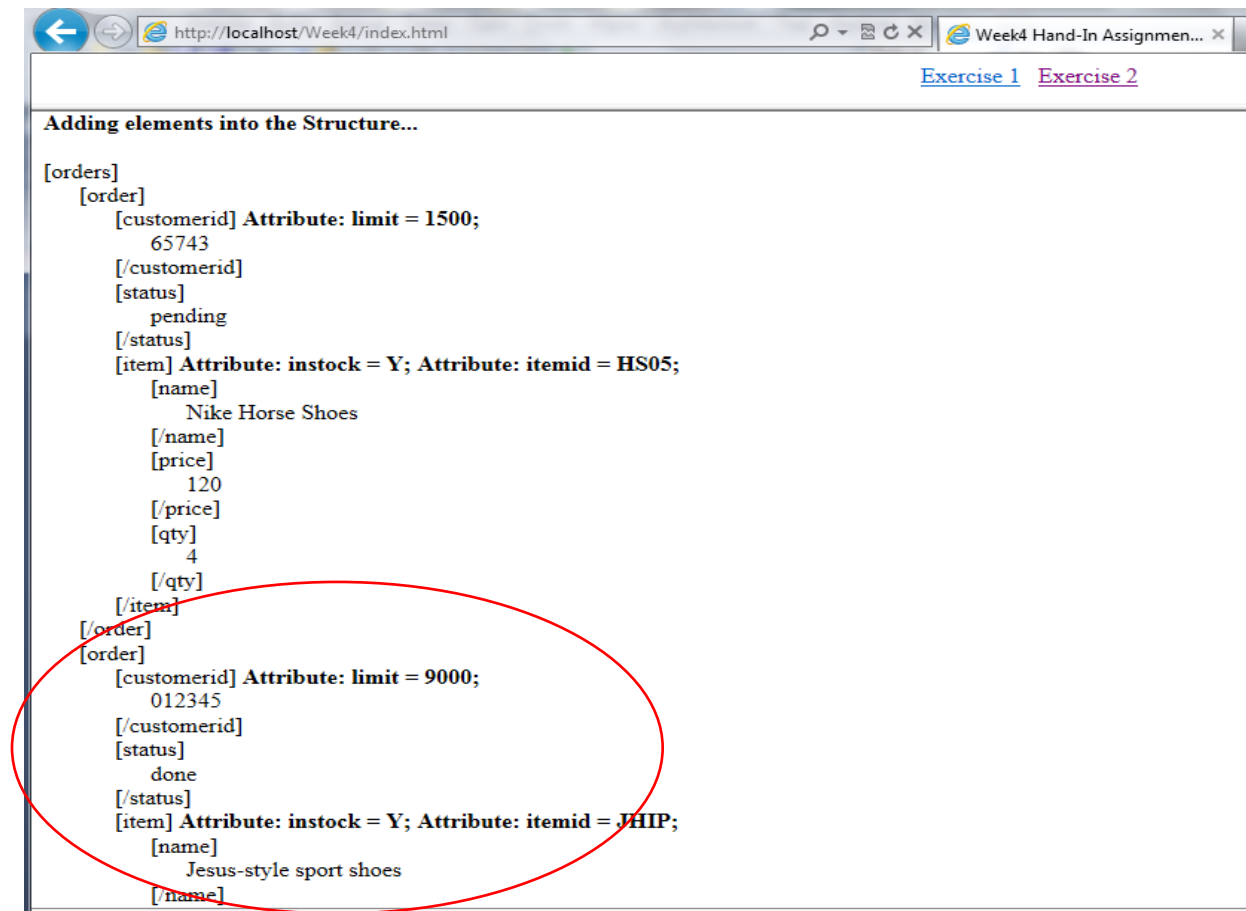


Figure 3.20 Getting results from a XML

This above is a partial screen because the first part shows the original content of the orders2.xml file. Finally, the entire assignment is available online at: <http://190.254.4.13/Week4/index.html>

### 3.3.4. Hand-In Assignment 4

#### Exercise 1:

Look at the “flamingo” database provided, in this exercise, an associated result set about invoices and their line items must be extracted. XML is an excellent way to extract information

from databases, promoting portability (Evjen et al, 2007: 323). In order to do this, an SQL query must be created by using a join between two tables. Due to the complexity of this query, a visual tool has been used in order to generate the SQL code; to do that, Microsoft Visual Studio was used, especially a built-in tool called *Server Explorer*, which is able to connect data sources. In the server explorer, a new query was created through visual resources as seen in the figure below:

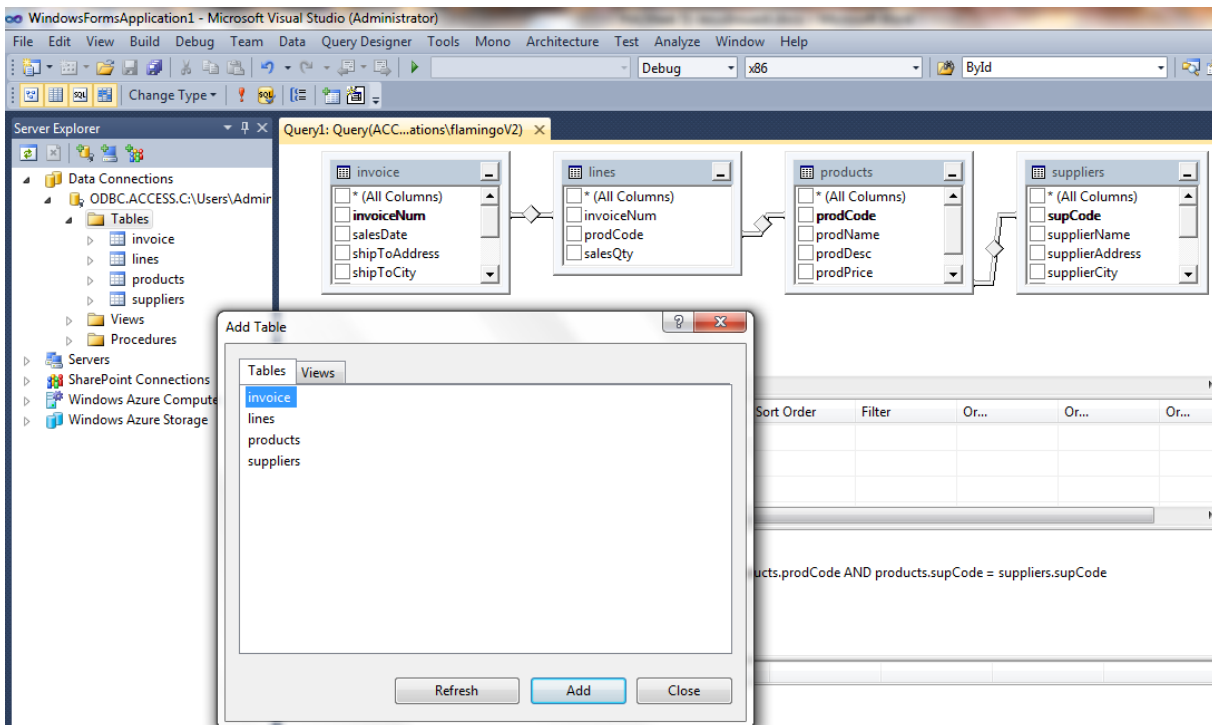


Figure 3.21 Microsoft Visual Studio 2010, the usage of the Server Explorer with Query generators.

When the connection was established, all the structures and relationships could be read into the query generator. The next step was about the selection of the items (fields) from the tables to include in the query. Thus, an SQL query was automatically generated as shown in figure 3.22. When the query was designed, the following SQL statement was generated:

```

SELECT invoice.invoiceNum, products.prodCode, products.prodName, products.prodDesc,
lines.salesQty, products.prodPrice, invoice.contactName, invoice.shipToAddress,
invoice.shipToCity, invoice.shipToState, invoice.shipToZip, invoice.salesDate,
suppliers.supplierName, suppliers.supplierContact
FROM invoice, lines, products, suppliers
WHERE invoice.invoiceNum = lines.invoiceNum AND lines.prodCode = products.prodCode AND
products.supCode = suppliers.supCode AND (invoice.shipToState = 'NSW')

```

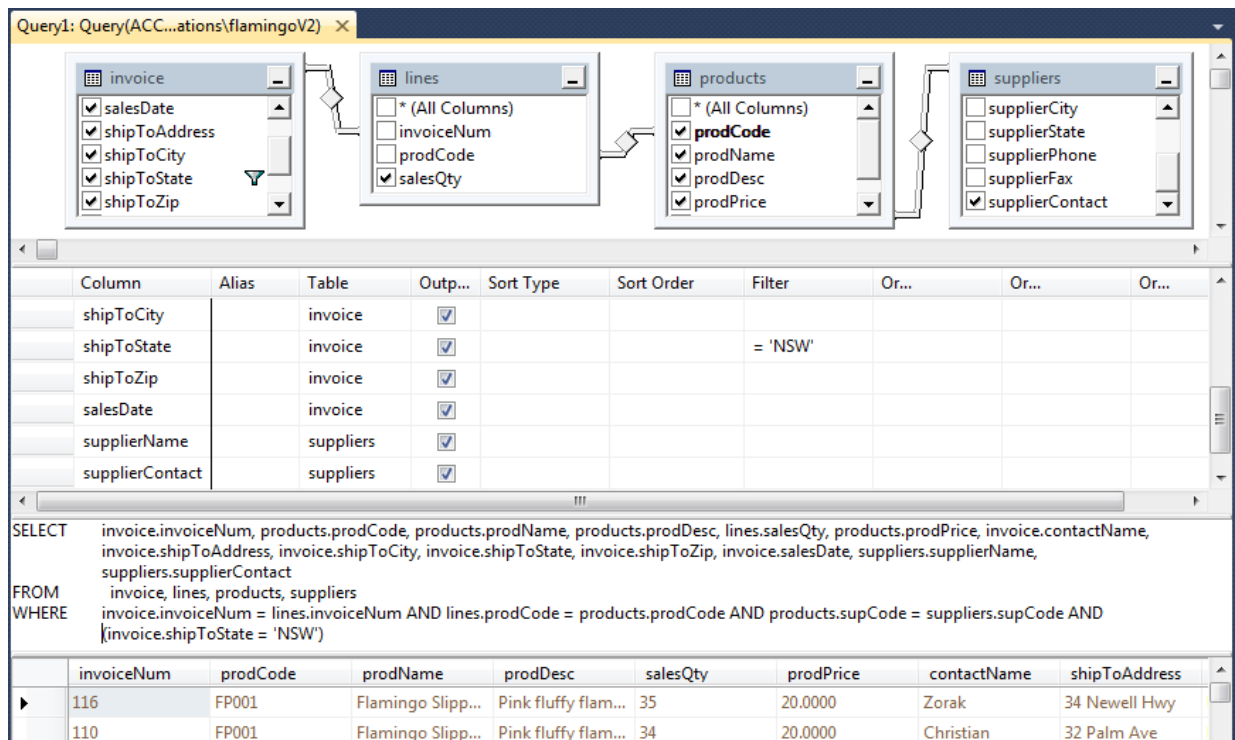


Figure 3.22 Generated query in Microsoft Visual Studio 2010

Once the SQL statement was generated, it was used within the DB2XML tool with in order to extract the results in XML format. For that, three steps were needed: connection settings, the creation of an output file and the execution of the generated query. In terms of connection, the database name in DB2XML was set as *jdbc:odbc:flamingo* (where the flamingo data source name was previously configured in the system's ODBC). Then an XML file called *week5ex1.xml* was

DB2XML Application

DB2XML V 1.4

Exit Help Properties: Save Load

DTD Database XML Attributes XML Elements Stylesheet

Database query: ☐ Complete Database ☒ Partial Database

```
[flamingo] SELECT invoice.invoiceNum, products.prodCode, products.prodName, products.prodDesc, lines.salesQty, products.prodPrice, invoice.contactName,
invoice.shipToAddress, invoice.shipToCity, invoice.shipToState, invoice.shipToZip, invoice.salesDate, suppliers.supplierName,
suppliers.supplierContact
FROM invoice, lines, products, suppliers
WHERE invoice.invoiceNum = lines.invoiceNum AND lines.prodCode = products.prodCode AND products.supCode = suppliers.supCode AND
(invoice.shipToState = 'NSW')
```

☐ Standard out ☒ File output

File name C:\Users\Administrator\Desktop\week5ex1.xml Browse

Generate XML

Specification of Database

Driver class: sun.jdbc.odbc.JdbcOdbcDriver

Database URL: jdbc:odbc:flamingo

User name:

Password:

Parameters: encoding=UTF8

ok

The generated file has the following code which is attached to this assignment. It was renamed as Exercise1(Week5).xml in order to be attached to this document (Figure 3.24):

```
Exercise1(Week5).xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet href="http://sanfrancisco/xml/db2xml/test.xsl" type="text/xsl"?>
<!--
Generated with DB2XML version 1.4
http://www.informatik.fh-wiesbaden.de/~turau/DB2XML/index.html
Database: jdbc:odbc:flamingo
Date: Wed Sep 05 16:00:28 COT 2012
Driver: JDBC-ODBC Bridge (ACEODBC.DLL) 2.0001 (Microsoft Access database engine)
Database system: ACCESS 04.00.0000
-->
<database URL="jdbc:odbc:flamingo">
<flamingo QUERY="SELECT invoice.invoiceNum, products.prodCode, products.prodName, products.prodDesc, lines.salesQty, products.prodPrice, invoice.contactName, invoice.shipToState FROM invoice, lines, products, suppliers WHERE invoice.invoiceNum = lines.invoiceNum AND lines.prodCode = products.prodCode AND products.supCode = suppliers.supCode AND (invoice.shipToState = 'NSW')">
<flamingo_rec>
<invoiceNum>116</invoiceNum>
<prodCode>FP001</prodCode>
<prodName>Flamingo Slippers</prodName>
<prodDesc>Pink fluffy flamingo slippers with rubber soles</prodDesc>
<salesQty>35</salesQty>
<prodPrice>20.0000</prodPrice>
<contactName>Zorak</contactName>
<shipToAddress>34 Newell Hwy</shipToAddress>
<shipToCity>Narrandera</shipToCity>
<shipToState>NSW</shipToState>
```

184 | Page

## **Exercise 2:**

In looking for a more detailed description of the first exercise, an XML Web Service was developed based on Microsoft .NET Technology. It was created using Microsoft Visual Studio 2010. An empty ASP.NET project was created in addition to a Web Service called “*Default.asmx*” –along with its code *Default.cs*– and all the construction was based on ODBC technology within Microsoft .NET environment (*namespace System.Data.Odbc*) which includes: *OdbcConnection*, *OdbcDataAdapter*, and *DataSet*. This will be covered in detail later.

Another connection was created in Microsoft Visual Studio 2010 through a tool called Server Explorer. This connection used Microsoft ODBC as its data source and the identification “flamingo” in the System DNS catalog. This connection is shown in Figure 3.25.

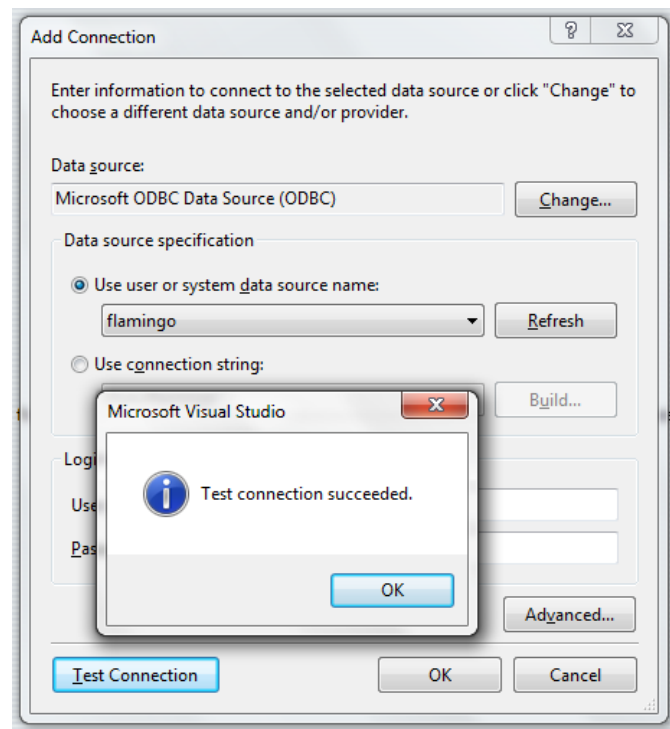


Figure 3.25 Connection setting for flamingo based on ODBC.

Once the connection was established, a new query was created to include every single field of the tables set. This query was structured as shown in Figure 3.26.

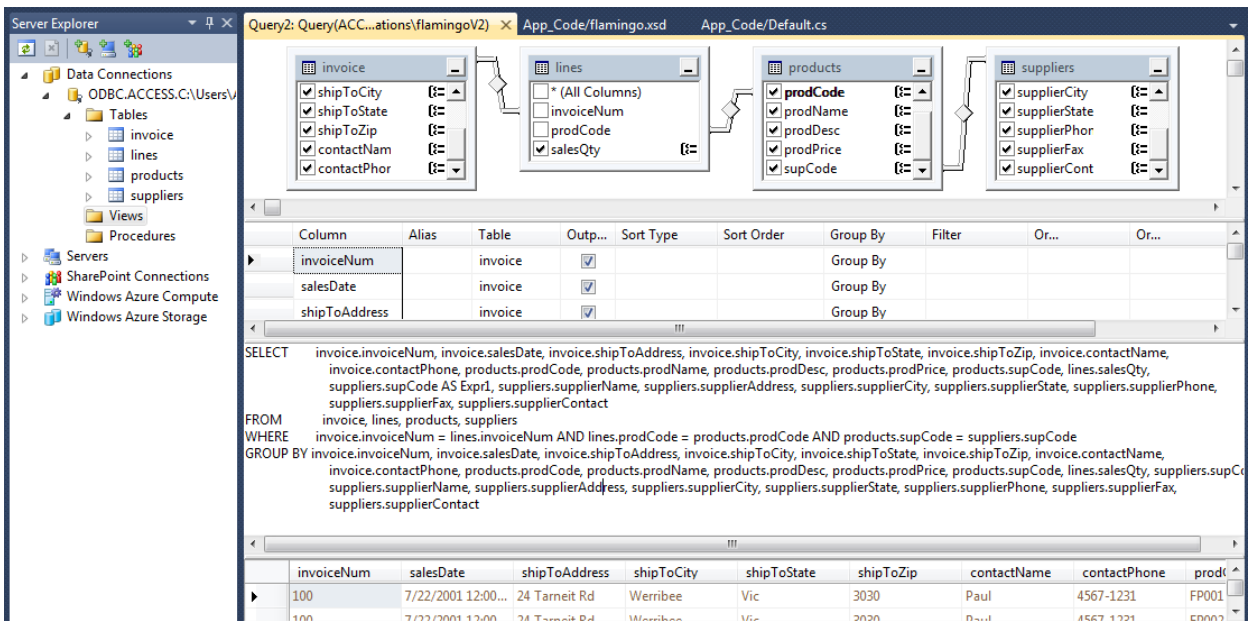


Figure 3.26 New Query including detailed information of the table set.

This new query was the base for the construction of the XML Web Service. The SQL Code is necessary when creating a web method of extracting information. In simple terms, the XML Web Service worked on the traditional model of ASP.NET in asynchronous mode. Three active elements are required, and an interesting way to explain asynchronous mode in ASP.NET with DataSets, is by using the following metaphor:

Computing Concept	Metaphor
Database Container	A water tank
Data Adapter	A tap
DataSet	A bucket

Table 3.3 Metaphors



If one wants some water (data), one must connect a tap (data adapter) with a water tank (database container); thus, when the tap is turned, water flows into the bucket (filling the dataset). One already has some water, even if the tap is closed (asynchronous mode).

This metaphor is often used when teaching the asynchronous mode in Microsoft .NET technology.

According to above, the code for the XML Web Service is fairly simple written in Microsoft C#. The entire code is provided as follows in three main parts:

```
using System;

using System.Data;
using System.Data.Odbc;
using System.Web.Services;
using System.Configuration;
using flamingoTableAdapters;

[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class _Default : System.Web.Services.WebService
{
    private OdbcConnection xcn = null;
    private OdbcDataAdapter xda = null;
    private DataSet xds = null;

    //-----
    public _Default ()
    {
        xcn = new
OdbcConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString);
        xcn.Open();
        xda = new OdbcDataAdapter("SELECT invoice.invoiceNum, ...", xcn);
        xds = new DataSet();
    }
    //-----
    [WebMethod]
    public DataTable GetInvoices()
    {
        xds.Clear();
        xda.Fill(xds);
        return (xds.Tables[0]);
    }
    //-----
}

}
```

The code is divided into three parts by red brackets on the right side:

- PART I** (Lines 1-10): Includes the using statements and the class declaration.
- PART II** (Lines 11-15): Includes the constructor method.
- PART III** (Lines 16-20): Includes the WebMethod.



This final XML document is saved as Exercise2(Week5).xml and it is attached to this assignment. In its structure, a detailed description of the query in terms of column mapping with its respective data types can be seen. Then, each row of the query is provided by following the previous description. This XML document is more detailed in comparison with the first exercise in this assignment.

### **Exercise 3:**

At this stage, it is necessary to extract the exact structure from the relational model and put it into XML format. To do that, all tables are dragged into the DataSet of the same project, flamingo.xsd. See Figure 3.28:

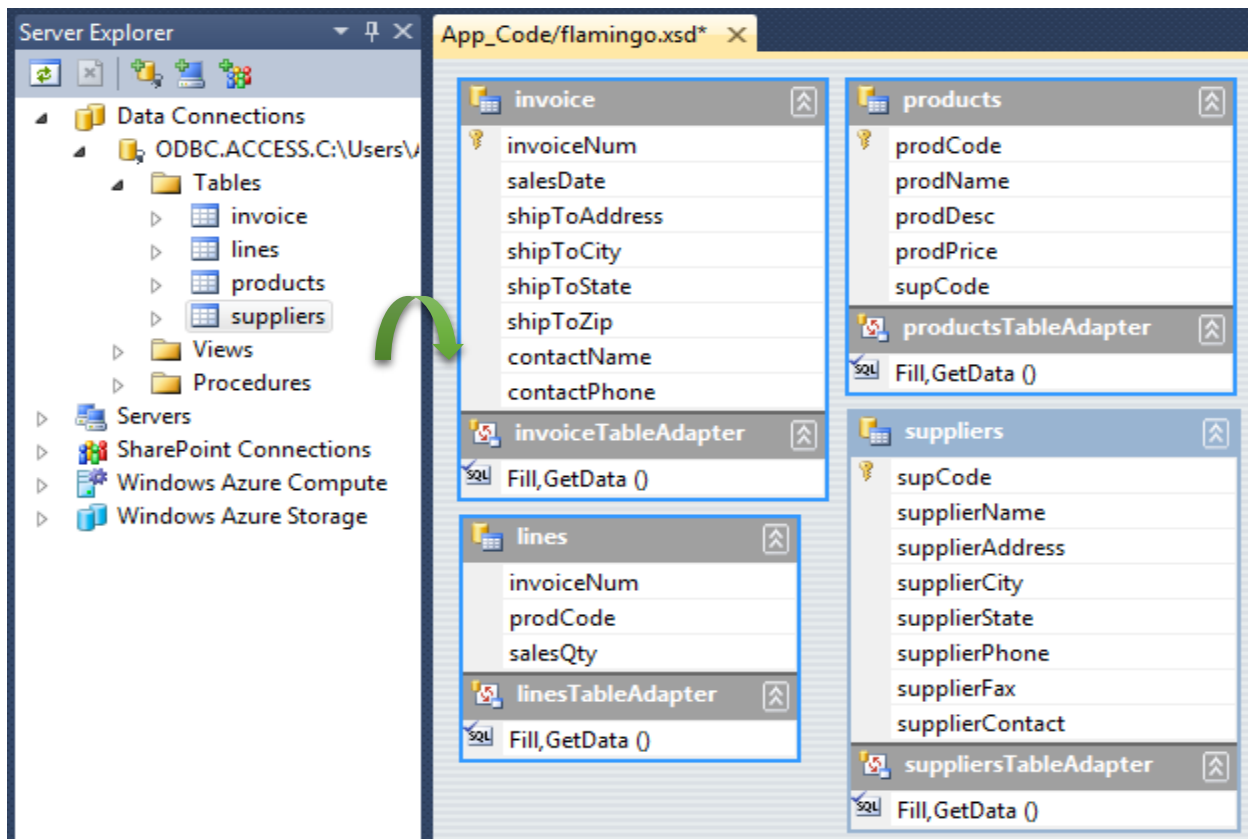


Figure 3.28 Dragging tables from the server explorer into the dataset

By saving the changes in the dataset file (flamingo.xsd), a complete description of each table is automatically generated. This file is saved as Exercise3(Week5).xml. A summarized view is shown in Figure 3.29 taking into account the size of this XML document.

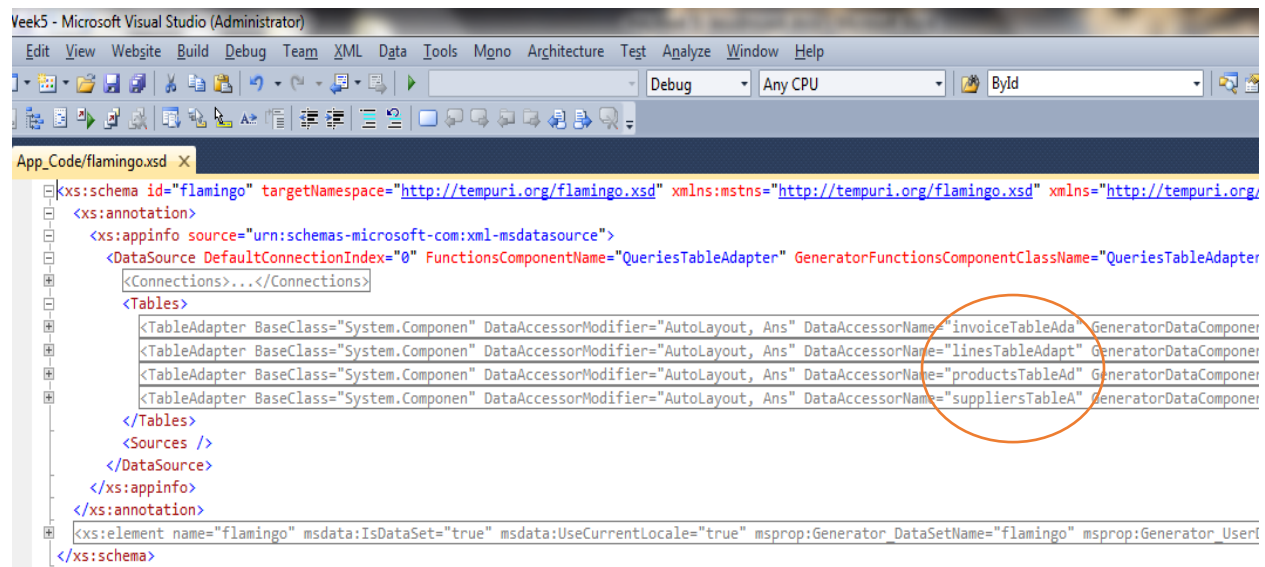


Figure 3.29 Summarized view of the XML document with the entire structure of the table set.

### 3.3.5. Hand-In Assignment 5

#### **Part 1:**

Thinking about XML Web Services as a way to transform data (Evjen et al, 2007: 565), the general scheme of a function comes to mind. A function –web method in SOAP jargon– needs some parameters (if necessary) and produces a result. In this vein, an XML Web Service was built to convert binary numbers into their decimal equivalents using Microsoft ASP.NET technology.

In Microsoft Visual Studio 2010, a new web site called Week7 was created as an ASP.NET Empty Project working with Microsoft .NET Framework 2.0. Then, two files were created: a

default web service (B2D.asmx) and a configuration file (web.config). All this is shown in Figure 3.30 below.

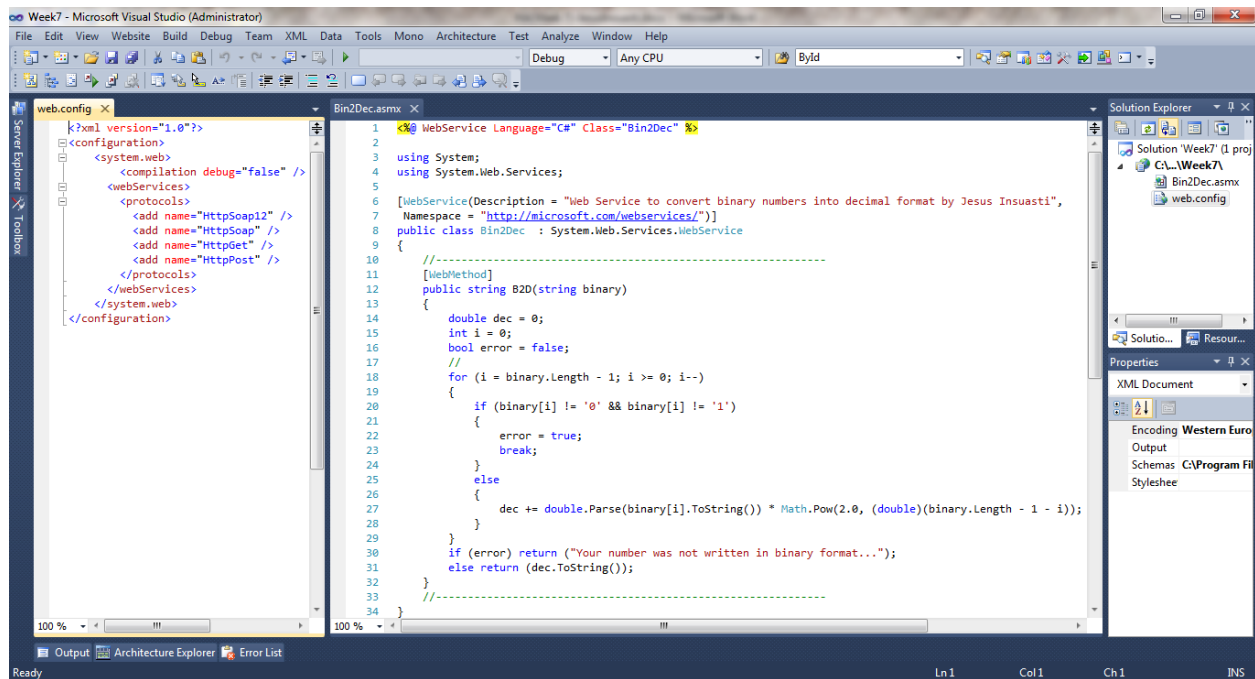


Figure 3.30 The creation of the XML Web Service with two files: the configuration file and the web service file.

According to the C# code, the web method requires a binary number written as a string. This string is scanned in reverse with the sum of the powers of two, just like the traditional algorithm. This web method can be tested online thanks the inner configuration of the *web.config* file. This XML Web Service is available on the student server at: <http://student.ohecampus.com/projects/Group92/Stud04/Bin2Dec.asmx> ; as shown in Figure 3.31.

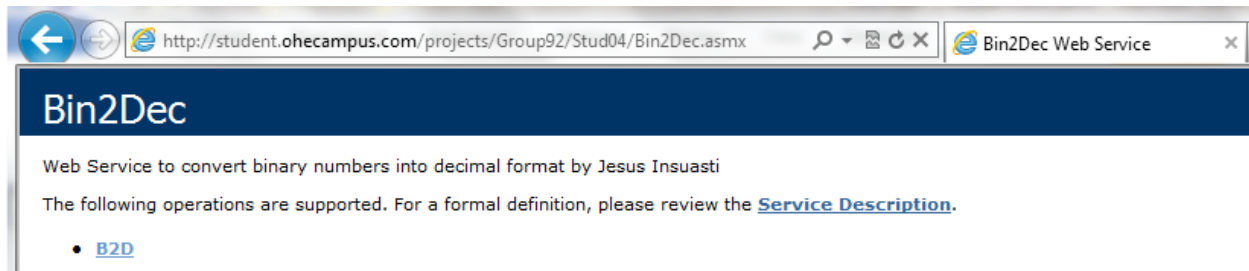


Figure 3.31 The XML Web Service available on the student server

Its description can be generated through WSDL when the GET parameter is attached to the URL: `?WSDL`

The test form is available online; the input parameters are provided in order to get results. An example is provided in Figure 3.32.

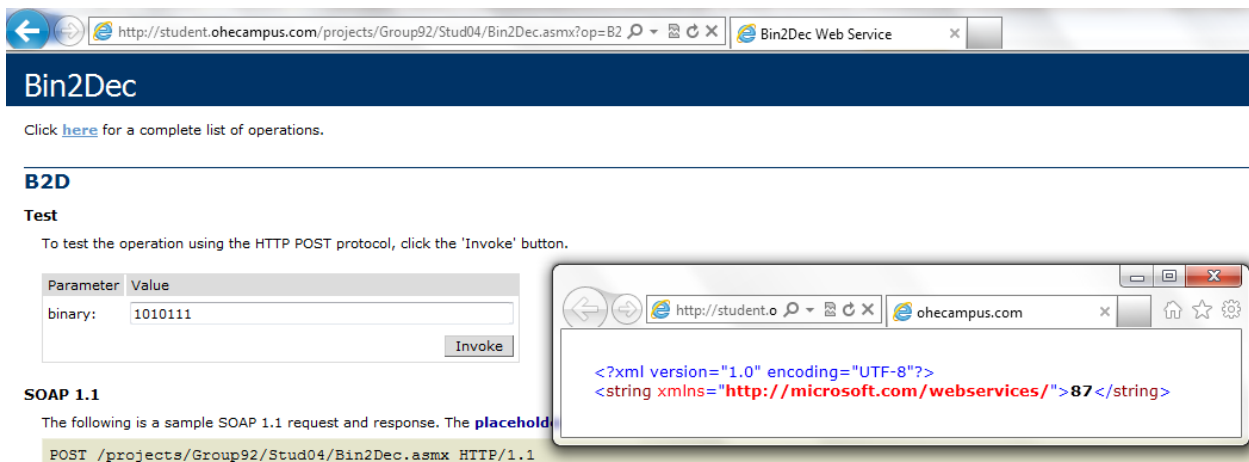


Figure 3.32 Remote consumption of the XML Web Service

In this way, this XML Web Service is ready for remote consumptions via SOAP. This part is added to this document as Part1.zip

## **Part 2:**

Two XML Web Services were picked which were built by our classmates: the first one is called *DistanceConverterService* created by Christopher Adams (available at: <http://student.ohcampus.com/projects/Group92/stud01/DistanceConverterService.asmx>) and the second one is called *CurrencyConverterService* created by Chye Har Tyler (available at: <http://student.ohcampus.com/projects/Group92/Stud16/CurrencyConverterService.asmx>).

According to their WSDLs, the first XML Web Service has two Web Methods to convert Miles into Kilometers (MtoKM) and vice versa (KMtoM). Both web methods require a float value and return another float value as result of the conversion process. The second XML Web Service is quite similar to the previous one; this service deploys two web methods to convert currencies: American Dollars into British Pounds (USDtoPounds) and vice versa (PoundsToUSD). The difference between these services is the data types used in the transactions. The second web service works with double data type for its web method (in parameters and results).

Taking into account the above, two ASP Scripts were created with the following common code:

```
<%@ Language = "VBScript" %>
<% Response.Buffer = True %>
<%
    Response.Write("<b>CURRENCY CONVERTER Client Script by: Jesus Insuasti </b><br />")
    Response.Write("XML Web Service provided by: !!! THE AUTHOR !!! <br /><hr />")
    Response.Write("We want to convert !!! INPUT VALUE AND DATA TYPE !!! into !!! OUTPUT TYPE
!!!. Thus, the result is: ")
    Const SoapServer = "student.ohcampus.com"
    Const SoapPath = "!!! THE SERVICE'S PATH !!!"
    Dim xmlhttp, soapMsg
    Set xmlhttp = Server.CreateObject("Microsoft.XMLHTTP")
    Set xmldom = server.CreateObject("Microsoft.XMLDOM")
    soapMsg = "<?xml version=""1.0"" encoding=""utf-8""?>"
```

```

soapMsg = soapMsg & "<soap:Envelope xmlns:xsi='\"http://www.w3.org/2001/XMLSchema-
instance\"' xmlns:xsd='\"http://www.w3.org/2001/XMLSchema\"'
xmlns:soap='\"http://schemas.xmlsoap.org/soap/envelope/\"'>"
soapMsg = soapMsg & "<soap:Body>"
soapMsg = soapMsg & " <!!! WEB METHOD !!! xmlns='\"http://tempuri.org/\"'>"
soapMsg = soapMsg & " <!!! INPUT IDENTIFIER !!!>1000</!!! INPUT IDENTIFIER !!!>"
soapMsg = soapMsg & " <!!! WEB METHOD !!!>"
soapMsg = soapMsg & "</soap:Body>"
soapMsg = soapMsg & "</soap:Envelope>"
xmlhttp.Open "POST", "http://" & SoapServer & SoapPath, False
xmlhttp.setRequestHeader "Man", "POST" & SoapPath & " HTTP/1.1"
xmlhttp.setRequestHeader "Host", SoapServer
xmlhttp.setRequestHeader "Content-Type", "text/xml; charset=utf-8"
xmlhttp.setRequestHeader "SOAPAction", "http://tempuri.org/!!! WEB METHOD !!!"
xmlhttp.send(soapMsg)
If (xmlhttp.Status = 200) Then
    soapResponse = xmlhttp.responseText
    Set xmldom = xmlhttp.responseXML
    Set result = xmldom.selectSingleNode("//!!! WEB METHOD !!!Response/!!! WEB METHOD
!!!Result")
    Response.Write(result.text)
Else
    xmlGeoSoap = ""
    Response.Write("Server Error...<br>")
    Response.Write("status = " & xmlhttp.status)
    Response.Write("<br>" & xmlhttp.statusText)
    Response.Write("<br><pre>" & Request.ServerVariables("ALL_HTTP") & "</pre>")
End If
Set xmlhttp = nothing
%>

```

With this common code, the highlighted elements correspond to the following table according with the type of consumption:

Common Element	DistanceConverter.asp	CurrencyConverter.asp
!!! THE AUTHOR !!!	Christopher Adams	Chye Har Tyler
!!! INPUT VALUE AND DATA TYPE !!!	10 Kilometers	1000 US\$
!!! OUTPUT TYPE !!!	Miles	British Pounds
!!! THE SERVICE'S PATH !!!	/projects/Group92/stud01/Distance ConverterService.asmx	/projects/Group92/Stud16/CurrencyConverterService.asmx
!!! WEB METHOD !!!	KMtoM	USDtoPounds
!!! INPUT IDENTIFIER !!!	A	value

Table 3.4 Common code



All the files have been compressed into a single file called HA(Week7)-JesusInsuasti.zip. This file contains the resultant XML Web Service, a configuration file and two scripts according to the filenames in the last table. Those files are described below:

Default.html	-> the main web page that links the entire solution
Bin2Dec.aspx	-> our XML web service
CurrencyConverter.asp	-> script for consuming the first service
DistanceConverter.asp	-> script for consuming the second service
Web.config	-> the configuration file (for allowing test forms)

### **3.3.6. Hand-In Assignment 6**

#### **Exercise 1:**

One way to draw a graph using a set of assertions is by doing it directly in a professional RDF editor (Evjen et al, 2007: 762). In this case, Altova SemanticWorks 2009 was used. Using the following sentences, a collection of subjects, predicates and objects is required. Thus, the subjects are: “Paul”, “the MSc”, and “the WA module”; objects are: “the WA subject”, “MSc modules”, “Laureate” and “MSc module”; finally the predicates are: “developed”, “consists of many”, “is an instructor for”, “is a course offered by” and “is an”.

1. Paul developed the WA Subject
2. The MSc consists of many MSc Modules
3. Paul is an instructor for Laureate
4. The MSc is a course offered by Laureate
5. The WA module is an MSc Module

To do this in Altova SemanticWorks, a new file is created by using a RDF/OWL level. Then, in the Resources window, twelve items are added with the custom prefix ex., see figure 3.33.

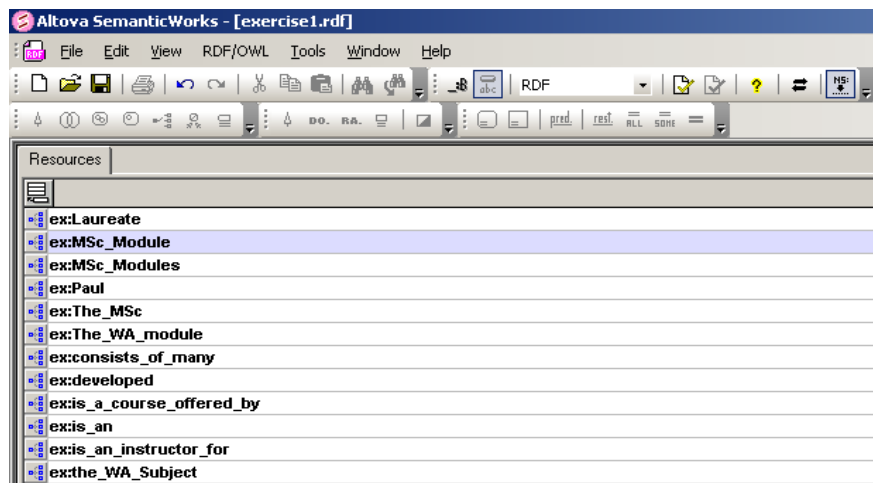


Figure 3.33 Twelve resources are the base of the graph.

By clicking on the icon in front of each resource, it is possible to create graphs integrating different elements. Each subject was edited in order to construct the graph shown in Figure 3.34. This file is saved as Exercise1.rdf.

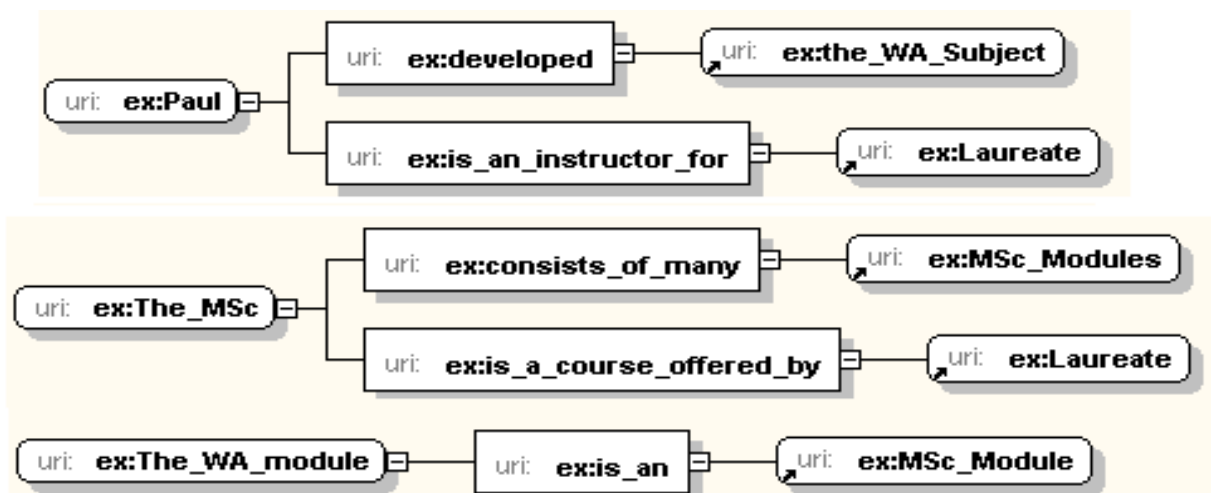


Figure 3.34 Graph about the 5 assertions.

## **Exercise 2:**

To make a copy of Exercise1.rdf, a new file called Exercise2.rdf is created by following the resources defined as follows:

- Paul: Paul.Darbyshire@vu.edu.au
- KIT: <http://www.ohecampus.com/home/>
- MSc: <http://www.ohecampus.com/programmes/msc.phtml>
- WA module: [http://www.ohecampus.com/programmes/msc\\_module.phtml/WA](http://www.ohecampus.com/programmes/msc_module.phtml/WA)

With this information, the RDF has changed its content. The source code is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

<rdf:RDF      xmlns:ex="http://www.example.org/"      xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-
schema#">

    <rdf:Description rdf:about="mailto:Paul.Darbyshire@vu.edu.au" >

        <ex:developed>

            <rdf:Description rdf:about="http://www.example.org/the_WA_Subject"/>

        </ex:developed>

        <ex:is_an_instructor_for>

            <rdf:Description rdf:about="http://www.example.org/Laureate"/>

        </ex:is_an_instructor_for>

    </rdf:Description>

    <rdf:Description rdf:about="http://www.ohecampus.com/programmes/msc.phtml">

        <ex:consists_of_many>

            <rdf:Description rdf:about="http://www.example.org/MSc_Modules"/>
```

```

    </ex:consists_of_many>

    <ex:is_a_course_offered_by>

        <rdf:Description rdf:about="http://www.example.org/Laureate"/>

    </ex:is_a_course_offered_by>

</rdf:Description>

<rdf:Description rdf:about="http://www.ohecampus.com/programmes/msc_module.phtml/WA">

    <ex:is_an>

        <rdf:Description rdf:about="http://www.example.org/MSc_Module"/>

    </ex:is_an>

</rdf:Description>

</rdf:RDF>

```

### **Exercises 3 and 4:**

A new file called Exercise3.rdf is created. In this file, new resources are incorporated: Emanuel (emanuel@mail.shenkar.ac.il) and Sally (Sally@ohecampus.com). Firstly, they are added with the default prefix ex. Then, the new elements interact with the previous model according to the following new assertions:

1. Paul and Emanuel created the WA module together which forms part of the MSc offered by KIT.
2. Sally says that the WA Module created by Paul and Emanuel is very good.

The first assertion must be broken down into three sentences like this:

- 1A. The WA module forms part of the MSc
- 1B. The MSc is offered by KIT
- 1C. Paul and Emanuel created the WA module together which forms part of the MSc offered by KIT.

With this, a new list of resources is produced in accordance with the two assertions. This can be seen in Figure 3.35.

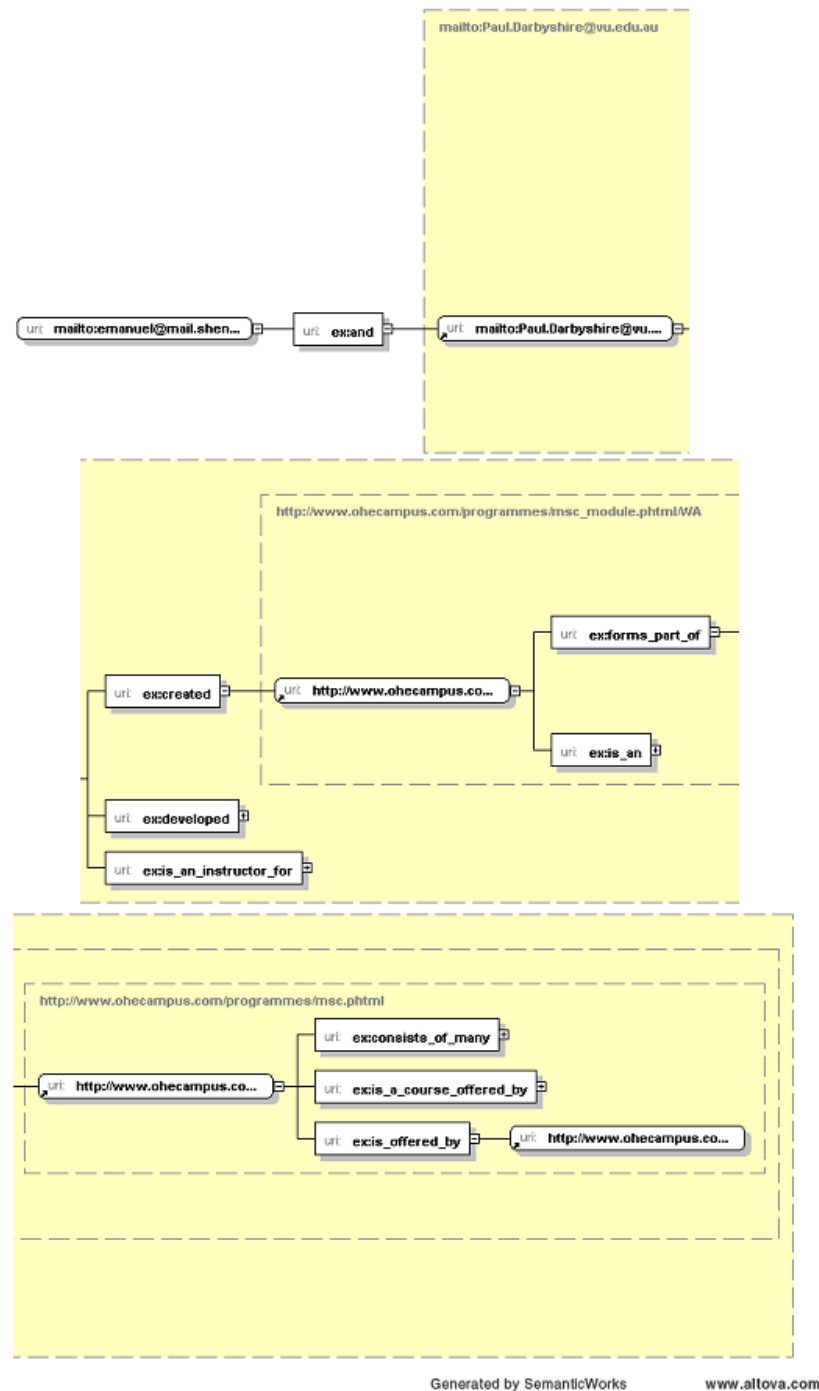


Figure 3.35 The graph of the second new assertion: Paul and Emanuel created the WA module together which forms part of the MSc offered by KIT.

For the second assertion, a new predicate was reinterpreted in order to simplify the graph. It is the result of joining a verbal expression ("says that") with a qualifier ("is very good"). Thus, the subject Sally utilizes a predicate referring to the previous construction. In this way, the graph is simplified according to the Figure 3.36.

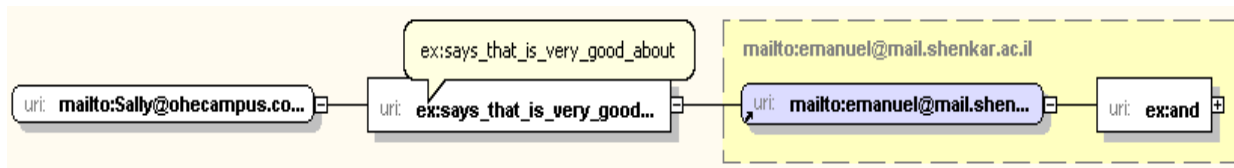


Figure 3.36 The graph for the second new assertion: Sally says that the WA Module created by Paul and Emanuel is very good.

Summarizing all these actions, the final code is saved in the file called Exercise3.rdf is shown below:

```
<?xml version="1.0"?>
<rdf:RDF xml:base="file:///C:/Documents%20and%20Settings/Administrator/Desktop/exercise2.rdf"
  xmlns:ex="http://www.example.org/" xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-
  schema#">
  <rdf:Description rdf:about="mailto:Paul.Darbyshire@vu.edu.au">
    <ex:developed>
      <rdf:Description rdf:about="http://www.example.org/the_WA_Subject"/>
    </ex:developed>
    <ex:is_an_instructor_for>
      <rdf:Description rdf:about="http://www.example.org/Laureate"/>
    </ex:is_an_instructor_for>
    <ex:created>
      <rdf:Description
rdf:about="http://www.ohcampus.com/programmes/msc_module.phtml/WA"/>
    </ex:created>
```

```

</rdf:Description>
<rdf:Description rdf:about="http://www.ohecampus.com/programmes/msc.phtml">
    <ex:consists_of_many>
        <rdf:Description rdf:about="http://www.example.org/MSc_Modules"/>
    </ex:consists_of_many>
    <ex:is_a_course_offered_by>
        <rdf:Description rdf:about="http://www.example.org/Laureate"/>
    </ex:is_a_course_offered_by>
    <ex:is_offered_by>
        <rdf:Description rdf:about="http://www.ohecampus.com/home/" />
    </ex:is_offered_by>
</rdf:Description>
<rdf:Description rdf:about="http://www.ohecampus.com/programmes/msc_module.phtml/WA">
    <ex:is_an>
        <rdf:Description rdf:about="http://www.example.org/MSc_Module"/>
    </ex:is_an>
    <ex:forms_part_of>
        <rdf:Description
rdf:about="http://www.ohecampus.com/programmes/msc.phtml"/>
        </ex:forms_part_of>
    </rdf:Description>
    <rdf:Description rdf:about="mailto:emanuel@mail.shenkar.ac.il">
        <ex:and>
            <rdf:Description rdf:about="mailto:Paul.Darbyshire@vu.edu.au"/>
        </ex:and>
    </rdf:Description>
    <rdf:Description rdf:about="mailto:Sally@ohecampus.com">
        <ex:says_that_is_very_good_about>
            <rdf:Description rdf:about="mailto:emanuel@mail.shenkar.ac.il"/>
        </ex:says_that_is_very_good_about>
    </rdf:Description>
</rdf:RDF>

```

All the exercises are attached to this document in a compressed file.

## BIBLIOGRAPHY

- ACM (1968) 'Hollerith Punched Card Code' *Communications of the ACM*. April 1968. 11(4). pp.275-281, 7p.
- Adams, M. (2009) 'Empirical evidence and the knowledge-that/knowledge-how distinction'. *Synthese*. 170(1). pp.97-114. DOI: 10.1007/s11229-008-9349-z.
- Aggarwal, N., Dai Q., & Walden, E. (2006) 'Do Markets Prefer Open or Proprietary Standards for XML Standardization? An Event Study', *International Journal of Electronic Commerce*, Fall 2006, 11(1), pp.117-136, DOI: 10.2753/JEC1086-4415110105.
- Ahmed, H. et al (2011) 'Designing a WISHBONE Protocol Network Adapter for an Asynchronous Network-on-Chip', *IJCSI International Journal of Computer Science Issues*, 8(4) - No 2, July 2011.
- Aho, A., et al. (2006) *Compilers: Principles, Techniques, and Tools*. 2nd Edition. Pearson – Addison Wesley. Boston. USA. ISBN-13: 978-0321486813
- Allemang, D. & Hendler, J. (2008) *Semantic web for the working ontologist: modeling in RDF, RDFS and OWL*, Morgan Kaufmann Publishers/Elsevier, Amsterdam, Holland.
- Androutsellis-Theotokis, S., & Spinellis, D. (2004) 'A Survey of Peer-to-Peer Content Distribution Technologies', *ACM Computing Surveys*, 36(4), pp.335-371, December 2004. DOI: 10.1145/1041680.1041681, [Online] Available at: <http://www.spinellis.gr/pubs/jrnl/2004-ACMCS-p2p/html/AS04.html>, (Retrieved at March 3<sup>rd</sup>, 2012).
- Anthes, G. (2010) 'Happy Birthday, RDBMS!'. *Communications of the ACM*, May 2010, 53(5), pp.16-17, DOI: :10.1145/1735223.1735231.



- ARIN (2012) 'ARIN's Whois RESTful Web Service', American Registry for Internet Numbers, [Online] Available at: <https://www.arin.net/resources/whoisrws/index.html>, (Retrieved at: March 24th, 2012).
- Arora, K., Kumar, K., & Sachdeva, M. (2011) 'Impact Analysis of Recent DDoS Attacks', *International Journal on Computer Science & Engineering*, 2011, 3(2), pp.877-884, 8p.
- AT&T, & BRICS (2005) 'Document Structure Description', DSD Home Page, [Online] Available at: <http://www.brics.dk/DSD/>, (Retrieved at: August 22<sup>nd</sup>, 2012).
- Avgeriou, P. & Zdun, U. (2005) 'Architectural Patterns Revisited – A Pattern Language'. *10th European Conference on Pattern Languages of Programs*. Germany.
- Baun, C. et al. (2011) *Cloud computing: web-based dynamic IT services*, Springer, New York (NY), USA.
- BBC R&D (n.d.) DIRAC, The video compression family using open technology, BBC Research & Development, Available [Online] at: <http://www.bbc.co.uk/rd/projects/dirac/>
- Bean, J. (2003) *XML for data architects: designing for reuse and integration*, Morgan Kaufmann Publishers, Amsterdam, Holland.
- Berendt, B. et al (2004) *Web Mining: From Web to Semantic Web*, First European Web Mining Forum, EWMF2003 Cavtat-Dubrovnik, Croatia, Berlin / Heidelberg, 2004.
- Berghel, H. (2002) 'Hijacking the Web', *Communications of the ACM*, April 2002, 45(4), pp.23-27, 5p.
- Berners-Lee, T., & Connolly, D. (1993) 'Hypertext Markup Language (HTML): A Representation of Textual Information and MetaInformation for Retrieval and Interchange, 1st Draft', *CERN*, Genève, Switzerland, [Online] Available at: <http://www.w3.org/MarkUp/draft-ietf-iiir-html-01.txt>, (Retrieved at: March 10th, 2012).

- Bitan T., & Karni, A. (2004) 'Procedural and declarative knowledge of word recognition and letter decoding in reading an artificial script'. *Cognitive Brain Research*. Elsevier. USA
- Borer, T., Davies, T., Wilson, P. (2006) 'Open technology compression codecs: The BBC perspective'. *TVB Europe – United Business Media*, 15(9), pp.36-37
- Bourret, R. (2010) 'Native XML Databases', *XML and Databases consulting home page*, [Online] Available at: <http://www.rpbouret.com/xml/ProdsNative.htm>, (Retrieved at: September 1th, 2012).
- Brookshear, J. G. (2009) *Computer Science: An Overview*. 10th Edition. Addison-Wesley. Boston MA. USA
- Carr, N. (2008). Is Google making us Stupid? *The Atlantic*. July-August. Available [Online] at: <http://www.theatlantic.com/doc/200807/google>
- Chandra, K. (2003) 'STATISTICAL MULTIPLEXING', *Wiley Encyclopedia of Telecommunications*, 2003 1st Edition, 5(), pp.2420-2432, 13p, [Online] Available at: <http://morse.uml.edu/~kchandra/publications/stat-mux-EOT-02.pdf>, (Retrieved at: March 7<sup>th</sup>, 2012).
- Cisco Systems (2000) *Introduction to Gigabit Ethernet – Technology Brief*, Cisco Systems Home Page, [Online] Available at: [http://www.cisco.com/warp/public/cc/techno/lnty/etty/ggetty/tech/gigbt\\_tc.pdf](http://www.cisco.com/warp/public/cc/techno/lnty/etty/ggetty/tech/gigbt_tc.pdf), (Retrieved at: March 31<sup>st</sup>, 2012)
- Cisco Systems. (2008) 'Embedded Resource Manager (ERM)', *Cisco IOS Software Releases 12.3 T – Cisco's Home Page*, [Online] Available at: [http://www.cisco.com/en/US/docs/ios/12\\_3t/12\\_3t14/feature/guide/gt\\_rmimg.pdf](http://www.cisco.com/en/US/docs/ios/12_3t/12_3t14/feature/guide/gt_rmimg.pdf), (Retrieved at: April 21<sup>st</sup>, 2012).
- Cornish, A. (2004) 'Using a Native XML Database for Encoded Archival Description Search and Retrieval', *Information Technology & Libraries*, December 2004, 23(4), pp.181-184.

- Coulson, G. et al. (2012) 'Flexible Experimentation in Wireless Sensor Networks', *Communications of the ACM*, January 2012, 55(1), pp.82-90, DOI: 10.1145/2063176.2063198.
- CWHK (2009) 'DNS attack downs Internet in parts of China', *ComputerWorld Hong Kong*, June 2009, 26(5), pp.76-76.
- Damasio, A. (1994) *Descartes Error: Emotion, Reason, and the Human Brain*. Gosset/Putnam Press. New York. USA
- DARPA (1981) 'TRANSMISSION CONTROL PROTOCOL Specification (RFC 793)', *Internet Engineering Task Force*, [Online] Available at: <http://tools.ietf.org/html/rfc793>, (Retrieved at: March 17<sup>th</sup>, 2012).
- De la Rosa, A. (2001) 'Markup languages applied to the transformation of documentary structures', *El Profesional de la Información*. Jan/Feb, 2001, 10(1/2), pp.4-22.
- Derrick, S. (2012) 'Smarter, more relevant', *Marketing (00253650)*. September 5<sup>th</sup> 2012, special section pp.8-9.
- Dirac Video Compression (2008) Dirac Specification, Version 2.2.3, Available [Online] at: <http://diracvideo.org/download/specification/dirac-spec-latest.pdf>
- DuHamel, B. (1987) Atari System Reference Manual, Available [Online]: <http://www.classiccmp.org/dunfield/atarixl/atariref.pdf>
- Edmondson-Yurkanan, C. (n.d.) 'A Technical History of the ARPANET - A Technical Tour: IMP-to-IMP', *Department of Computer Science at University of Texas*, [Online] Available at: [http://www.cs.utexas.edu/~chris/think/ARPANET/Technical\\_Tour/impimp.shtml#overview](http://www.cs.utexas.edu/~chris/think/ARPANET/Technical_Tour/impimp.shtml#overview), (Retrieved at: April 21<sup>st</sup>, 2012).

- EMC Corp. (2012) ‘How large a key should be used in the RSA cryptosystem?’, *RSA Laboratories*, [Online] Available at: <http://www.rsa.com/rsalabs/node.asp?id=2218>, (Retrieved at: April 14th, 2012).
- Evjen, B. et al. (2007) *Professional XML*, Wiley Publishing Inc. Indianapolis (IN), USA.
- Feixue, H., Yong, Z., & Zhijie, L. (2010) 'Simulation Evaluation Algorithms of LSRP and DVRP in Bank Network', *Computer & Information Science*, 3(1), pp.20-28, Computers & Applied Sciences Complete, EBSCOhost, (Retrieved at March 28<sup>th</sup>, 2012).
- France Telecom (2012), ‘OID Repository’, [Online] Available at: <http://www.oid-info.com/get/1.2.840.113556.4.2>, (Retrieved at: April 25th, 2012).
- Furst, C., Volk, M., Makeschin, F. (2010) ‘Squaring the Circle? Combining Models, Indicators, Experts and End-Users in Integrated Land-Use Management Support Tools’ *Environmental Management*. 46(6). p829-p833
- Gabbrielli, M., Martini, S. (2010) *Programming Languages: Principles and Paradigms*. London. Springer-Verlag London Limited.
- Ganesan, R., & Selvakumar, A. (2011) ‘A New Balanced Encryption Algorithm with Elevated Security Base on Key Update’, *European Journal of Scientific Research*, September 2011, 60(2), pp.177-194.
- Giampaolo, D. (1998) *Practical File System Design*. 1<sup>st</sup> Edition. Morgan Kaufmann Publishers Inc. San Francisco (CA). USA
- Glen Brookshear, J. (2009) *Computer Science: An Overview*. 10<sup>th</sup> ed. Boston: Pearson Education
- Goralski, W. (2009) *The illustrated network: how TCP/IP works in a modern network*, [electronic book], *Elsevier/Morgan Kaufmann Publishers*, Amsterdam – Holland

- Graham-Rowe, D. (2007). 'The drive for holography' *Nature Photonics*. April 2007. 1(4). pp.197-200, 4p, DOI: 10.1038/nphoton.2007.36
- Greengard, S. (2012) 'The War against Botnets', *Communications of the ACM*, February 2012, 55(2), pp.16-18, 3p, DOI: 10.1145/2076450.2076456.
- Gregg, M., & Watkins, S. (2006) Hack The Stack [Electronic Book] : Using Snort And Ethereal To Master The 8 Layers Of An Insecure Network, Rockland, MA: Syngress Pub., Retrieved at (March 17<sup>th</sup>, 2012).
- Hamdan, O. et al (2010) 'New Comparative Study Between DES, 3DES and AES within Nine Factors', *Journal of Computing*, March 2010, 2(3), pp.152-157.
- Haw, S., & Lee, C. (2011) 'Data storage practices and query processing in XML databases: A survey', *Knowledge-Based Systems*. December 2011, 24(8), pp.1317-1340, DOI: 10.1016/j.knosys.2011.06.006.
- Hendler, J. & Lassila, O. (2006) 'Current Status and Future Promise of the Semantic Web', Semantics Forum 2006, [Online] Available at: <http://www.cs.rpi.edu/academics/courses/fall07/semantic/Semantics2006-keynote.ppt>, (Retrieved at: September 22nd, 2012).
- Holcombe, M. (2008) Running an Agile Software Development Project. Wiley. Hoboken, NJ, USA
- ICANN (2011) 'ICANN Affirmation of Commitments: Responsibilities Inventory', *ICANN Home Page*, [Online] Available at: <http://aoctracking.icann.org/>, (Retrieved at: March 10th, 2012).
- IEBlog (2011) 'Internet Explorer 9 Network Performance Improvements', *Windows Internet Explorer Engineering Team Blog – Microsoft Corp.*, [Online] Available at:

<http://blogs.msdn.com/b/ie/archive/2011/03/17/internet-explorer-9-network-performance-improvements.aspx>, (Retrieved at: March 10<sup>th</sup>, 2012)

- IEEE (2007) ‘IEEE Standard for Information technology— Telecommunications and information exchange between systems— Local and metropolitan area networks— Specific requirements Part 11: Wireless LAN Medium Access Control’, *IEEE Standard 802.11*, 2007, [Online] Available at: <http://standards.ieee.org/getieee802/download/802.11-2007.pdf>, (Retrieved at: March 3<sup>rd</sup>, 2012).
- Ignite Technologies (2012) ‘Delivery Technology’, *Ignite Live Streaming – Ignite Technologies Home Page*, [Online] Available at: <http://ignitetech.com/platform/delivery-technology.aspx>, (Retrieved at: March 3<sup>rd</sup>, 2012).
- Insuasti, J. et al (2008) ‘Propuesta del Sistema de Información Integrado’, *Universidad de Nariño Press*, Language: Spanish, Pasto, Colombia.
- Januszewski, K. (2002) ‘The Importance of Metadata: Reification, Categorization, and UDDI’, *Microsoft Developer Network*, [Online] Available at: <http://msdn.microsoft.com/en-us/library/ms953942.aspx>, (Retrieved at: September 15<sup>th</sup>, 2012).
- Jorgensen, D. (2002) *Developing .NET Web Services with XML*. Rockland, MA. Syngress Pub. Inc.
- Kakish, M. (2011) ‘ENHANCING THE SECURITY OF THE RSA CRYPTOSYSTEM’, *International Journal of Research & Reviews in Applied Sciences*, 2011, 8(2), pp.239-246.
- Kalicharan, N. (2008) *Data Structures in C*. CreateSpace. USA
- Kan, S. (2002) *Metrics and Models in Software Quality Engineering*, 2nd Edition. Addison-Wesley Professional. Available [Online] at: <http://our.safaribooksonline.com/book/software-engineering-and-development/0201729156/software-quality-metrics-overview/ch04lev1sec1>

- Kaufman, C. et al. (2010) 'Internet Key Exchange Protocol Version 2 (IKEv2)', RFC-5996, *Internet Engineering Task Force (IETF)*, September 2010, [Online] Available at: <http://tools.ietf.org/html/rfc5996>, (Retrieved at: September 8<sup>th</sup>, 2012).
- KaZaA (2010) 'KaZaA Web Site Privacy Policy', *KaZaA Web Site*, [Online] Available at: <http://www.kazaa.com/#!/privacy>, (Retrieved at: March 3<sup>rd</sup>, 2012)
- Kehoe, T. (2011). Week 4-Numbers. *Discussion Board*. COMPUTER STRUCTURES.20110526.201 (UKL1.CMPSTR.20110526.201). University of Liverpool
- King, M. (2012) 'Reduce Privacy Threats by Deleting Cookies', *PC World*, February 2012, 30(2), pp.89-89.
- Koskinen, J. (2010) Software Maintenance Costs. University of Jyväskylä. Finland. Published: September 10, 2010. Available [Online] at: <http://users.jyu.fi/~koskinen/smcosts.htm>
- Kubiawicz, J. D. (2011) 'The OceanStore Project: Providing Global-Scale Persistent Data', *UC Berkeley Computer Science Division*, [Online] Available at: <http://oceanstore.cs.berkeley.edu/>, (Retrieved at: March 3<sup>rd</sup>, 2012).
- Kurose, J., & Ross, K. (2010) Computer Networking: A Top-down Approach 5<sup>th</sup> Edition, Pearson Education Inc. – Addison-Wesley, USA.
- Labriola, D. (2006) 'Battle of the New DVDs' *PC Magazine*. 10/3/2006. 5(17). pp.77-84, 6p.
- Labriola, D., Mangis, C. (2005) 'Discs after DVD: Blue-Light Specials' *PC Magazine*. 24(10). pp.129-135, 6p.
- Lanois, P. (2011) 'Privacy in the Age of the Cloud', *Journal of Internet Law*, December 2011, 15(6), pp.3-17.

- Lavin, M. (2006) 'Cookies: What do consumers know and what can they learn?', *Journal of Targeting, Measurement & Analysis for Marketing*, July 2006, 14(4), pp.279-288.
- Liu, B. (2011) *Web data mining: exploring hyperlinks, contents, and usage data*; Springer, Berlin, Germany.
- Loshin, P. (2004) *IPv6: theory, protocol, and practice*. 2<sup>nd</sup> Edition, [electronic book], Morgan Kaufmann, Amsterdam – Holland.
- Malcolm, G. (2001) *Introduction to Imperative Languages*, University of Liverpool, Last modified: Fri Oct 5 11:20:56 BST 2001 Available [Online]: <http://www.csc.liv.ac.uk/~grant/Teaching/COMP205/intro.html>,
- Marsan, C. (2009) 'DNS "blacklist" unveiled', *Network World*, August 2009, 26(26), pp.21-21.
- Microsoft Corp. (2009) IIS Smooth Streaming Transport Protocol, [Online] Available at: <http://go.microsoft.com/?linkid=9682896>, (Retrieved at: March 17<sup>th</sup>, 2012).
- Microsoft Corp. (2012) Media: The Official Microsoft IIS Site, [Online] Available at: <http://www.iis.net/media>, (Retrieved at: March 17<sup>th</sup>, 2012).
- Microsoft Corp. (n.d.) *Windows Server 2008 R2 Overview*. Microsoft Corporation. Available [Online] at: <http://www.microsoft.com/windowsserver2008/en/us/overview.aspx>
- Mitchell, J., Shmatikov, V., & Stern, U. (1997) 'Finite-State Analysis of SSL 3.0 and Related Protocols', *Computer Science Department, Stanford University*, [Online] Available at: <http://dimacs.rutgers.edu/Workshops/Security/program2/ssl/p.html>, (Retrieved at: April 18<sup>th</sup>, 2012).
- Møller, A., & Schwartzbach, M. (2006) *An Introduction to XML and Web Technologies*, Pearson/Addison-Wesley, UK.



- Moody, M. (2002) 'a brief(ish) history of p2p', *University of Florida – Interactive Media Lab*, [Online] Available at: <http://iml.jou.ufl.edu/projects/Fall02/Moody/history.html>, (Retrieved at: March 3<sup>rd</sup>, 2012).
- MSDN (n.d.) 'The Resolver and Adapter Provider Framework', *Microsoft Developer Network*, [Online] Available at: [http://msdn.microsoft.com/en-us/library/ee250250\(v=BTS.10\).aspx](http://msdn.microsoft.com/en-us/library/ee250250(v=BTS.10).aspx), (Retrieved at: September 15th, 2012).
- Mueller, M., & Chango, M. (2008) 'Disrupting Global Governance: The Internet Whois Service, ICANN, and Privacy', *Journal of Information Technology & Politics*, 5(3), pp.303-325, DOI: 10.1080/19331680802425503.
- Network Working Group (1993) 'Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers', *Internet Engineering Task Force*, [Online] Available at: <http://www.ietf.org/rfc/rfc1423.txt>, (Retrieved at: April 25th, 2012).
- Network Working Group (1999) 'Request for Comments: 2616', *The Internet Engineering Task Force (IETF)*, [Online] Available at: <http://www.ietf.org/rfc/rfc2616.txt>, (Retrieved at: March 10<sup>th</sup>, 2012).
- Network Working Group (2004) 'Whois Protocol Specification', *The Internet Engineering Task Force*, [Online] Available at: <http://tools.ietf.org/html/rfc3912>, (Retrieved at: March 24th, 2012)
- Network World (2010). 'Best of the Tests 2010'. *Network World*. 27(4). pp.24-28, 4p.
- Novak, J. (2002) 'Meaningful learning: the essential factor for conceptual change in limited or appropriate propositional hierarchies (LIPHs) leading to empowerment of learners', *Science Education*, 86(4) pp.548-571.

- Ojesanmi, O. A. (2009) 'Asynchronous Transfer Mode (ATM) Network', *Computer Science & Telecommunications*, 2009(6), pp.70-82, 13p.
- Oversi Networks Ltd. (2010) 'OverCache P2P Caching and Delivery Platform', Oversi Home Page, [Online] Available at: <http://www.oversi.com/en/products/overcache-msp/overcache-p2p.html>, (Retrieved at: March 3<sup>rd</sup>, 2012).
- Ozsu, M. & Valduriez, P. (2011) *Principles of Distributed Database Systems*. Springer Science+Business Media. New York. USA.
- Panksepp, J. (1998). *Affective Neuroscience: The foundations of human and animal emotions*. New York: Oxford University Press.
- Phifer, L. (2000) 'The Trouble with NAT', *The Internet Protocol Journal*, 3(4), [Online] Available at: [http://www.cisco.com/web/about/ac123/ac147/ac174/ac182/about\\_cisco\\_ipj\\_archive\\_article\\_09186a00800c83ec.html](http://www.cisco.com/web/about/ac123/ac147/ac174/ac182/about_cisco_ipj_archive_article_09186a00800c83ec.html), (Retrieved at: March 24<sup>th</sup>, 2012).
- Piscitello, D. (2006) 'It's Time for Enterprises to Take on DNS Security', *Business Communications Review*, November 2006, 36(11), pp.44-47.
- Poordavoodi, A., Khayyambashi, M. & Hamin, J. (2010) 'Using Replicated Data to Reduce Backup Cost in Distributed Databases'. *Journal of Theoretical & Applied Information Technology*. 2010. 14(1/2). pp.23-29.
- Radcliff, D. (2005) 'How to prevent pharming', *Network World*, July 2005, 22(28), pp.39-39.
- RealNetworks Inc. (2008) *REALPLAYER ENTERPRISE 2.1 DEPLOYMENT GUIDE*, Revision Date: 18 September 2008, Seattle (WA) USA, [Online] Available at:

- <http://docs.real.com/docs/enterprise/EnterpriseDeploymentGuide.pdf>, (Retrieved at: March 17<sup>th</sup>, 2012).
- RealNetworks Inc. (2012) Our Story, RealNetworks Home Page, [Online] Available at: <http://www.realnetworks.com/our-story/>, (Retrieved at: March 17<sup>th</sup>, 2012).
  - Rob, P. & Coronel, C. (2007) Database Systems: Design, Implementation, and Management. 8<sup>th</sup> Edition. Thomson- Course Technology. Boston (MA). USA.
  - Robins A., Roundtree J., Roundtree N. (2003) 'Learning and Teaching Programming: A Review and Discussion', *Computer Science Education*, 13 (5), pp.145-146.
  - Rosen, E. (1981) 'Vulnerabilities of Network Control Protocols: An Example – RFC 789', *Internet Engineering Task Force*, Bolt Beranek and Newman Inc., [Online] Available at: <http://tools.ietf.org/html/rfc789>, (Retrieved at: April 21<sup>st</sup>, 2012).
  - Rosen, R. (2009). This is Your Brain on the Web. *The Wilson Quarterly*. 33(4). pp.50-51.
  - Royce, W. (1970) 'Managing the Development of Large Software Systems'. *Proceedings of IEEE WESCON 26*. August 1970. pp.328-338.
  - Ruiz, V. et al (2009) 'Vulnerability in DNS', *Tono: Revista Técnica de la Empresa de Telecomunicaciones de Cuba, S.A.*, 6(2/3), pp.104-109, 6p.
  - Sadeghi, M. et al. (2012) 'Security Analysis of Routing Protocols in Wireless Sensor Networks', *International Journal of Computer Science Issues (IJCSI)*, January 2012, 9(1-3), pp.465-472.
  - Seacord, R., Plakosh, D., & Lewis, G. (2003) Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices. Addison Wesley. Carnegie Mellon Software Engineering Institute. USA

- Seltzer, L. (2005) 'Make Your Network Safer by Attacking It', *PC Magazine*, July 2005, 24(12), pp.80-80.
- Silberschatz, A., Galvin, P., & Gagne, G. (2005) *Operating Systems Concepts*. 7<sup>th</sup> Edition. Wiley. Hoboken NJ. USA
- Singleton, A. (2008) 'Linux at the BBC' *Linux Format*, Issue 105, pp.48-51, Available [Online] at: <http://www.linuxformat.com/includes/download.php?PDF=LXF105.bbc.pdf>
- Somogyi, S., & Schneier, B. (2001) 'The Perils of Port 80', *Communications of the ACM*, October 2001, 44(10), pp.168-168.
- SpeedTest.NET (n.d.) Available [Online] at: [www.speedtest.net](http://www.speedtest.net)
- Sperberg-McQueen, C. M. (2003) 'UDDI in one slide', *Web Services and the W3C – World Wide Web Consortium Talk*, [Online] Available at: <http://www.w3.org/2003/Talks/0818-msm-ws/slide23-0.html>, (Retrieved at: September 15<sup>th</sup>, 2012)
- Stone, D. (1999). *Computer Hacking*. Retrieved August 27, 2007 from: <http://www.uni.uiuc.edu/~dstone/hacking.html>.
- Taylor, R., Medvidovic, N., & Dashofy, E. (2009) *Software Architecture: Foundations, Theory, and Practice*. 1<sup>st</sup> Edition. Wiley. USA
- Tennison, J. (2000) 'The many faces of XSLT', *Markup Languages: Theory & Practice*. Fall 2000, 2(4), pp.459-468, DOI: 10.1162/109966200750428759.
- Tere, G., Jadhav, B., & Mudholkar, R. (2012) 'Dynamic Invocation of Web Services', *Advances in Computational Research*, March 2012, 4(1), pp.78-82.
- Tervo, H., Kilpeläinen, P., & Penttinen, T. (2006) 'Automating the XML conversion and XSL formatting of textual legacy data', *Proceedings of the Estonian Academy of Sciences, Engineering*. March 2006, 12(1), pp.3-15.

- Tiller, J. (2001) A technical guide to IPSec virtual private networks, [electronic book], CRC Press, Boca Raton (FL) USA.
- Trikha, B. (2010) 'A Journey from floppy disk to cloud storage' *International Journal on Computer Science and Engineering*, 2 (4), pp.1449-1452, ISSN: 0975-3397
- Upson, S. (2010). Rat Brain Robot Grows Up. Available [Online] at:  
<http://spectrum.ieee.org/automaton/biomedical/bionics/rat-brain-robot-grows-up>
- Vandelli, W. (2009) 'Readiness of the ATLAS Trigger and Data Acquisition system for the first LHC beams' *Nuclear Physics B Proceedings Supplement*, 197(1). pp.306-308, 3p. DOI: 10.1016/j.nuclphysbps.2009.10.091
- Voddler Group AB (2012) 'Help & Support', Voddler Home Page, [Online] Available at:  
<http://www.voddler.com/en/help/download/>, (Retrieved at: March 3<sup>rd</sup>, 2012).
- W3C (2001) 'Web Services Description Language (WSDL) 1.1', *Note from the World Wide Web Consortium*, [Online] Available at: <http://www.w3.org/TR/wsdl>, (Retrieved at: September 15<sup>th</sup>, 2012). In this vein, WSDL is not a standard in the strict sense of the word; in contrast, it is a suggestion by the W3C
- W3C (2004) 'Web Services Architecture', *W3C Working Group Note*, [Online] Available at:  
<http://www.w3.org/TR/ws-arch/#security>, (Retrieved at: September 8<sup>th</sup>, 2012).
- W3C (2010) 'XML ESSENTIALS', *World Wide Web Consortium*, [Online] Available at:  
<http://www.w3.org/standards/xml/core>, (Retrieved at: August 11<sup>th</sup>, 2012).
- W3C (2012) 'HTML & CSS', *World Wide Web Consortium*, [Online] Available at:  
<http://www.w3.org/standards/webdesign/htmlcss>, (Retrieved at: August 25<sup>th</sup>, 2012).
- W3C (n.d.) 'Semantic Web', *World Wide Web Consortium's Home Page*, [Online] Available at: <http://www.w3.org/standards/semanticweb/>, (Retrieved at: September 20<sup>th</sup>, 2012).

- Wikipedia (2011). *Software Framework*. [Online] Available from: [http://en.wikipedia.org/wiki/Software\\_framework](http://en.wikipedia.org/wiki/Software_framework) (Accessed: July 2<sup>nd</sup>, 2011)
- Wikipedia (n.d.) 'Solution of the Poincaré conjecture', Wikipedia [Online]. Available at [http://en.wikipedia.org/wiki/Solution\\_of\\_the\\_Poincar%C3%A9\\_conjecture](http://en.wikipedia.org/wiki/Solution_of_the_Poincar%C3%A9_conjecture) (Accessed 25-Jun-2011).
- Winston, P. (1984) *The AI Business: Commercial Uses of Artificial Intelligence*. MIT Press. Cambridge (MA). USA.
- Wirth, N. (1976) *Algorithms + Data Structures = Programs*. Prentice-Hall. ISBN 978-0-13-022418-7. 0130224189.
- Wirth, N. (1985). *Algorithms and Data Structures*. Prentice-Hall. ISBN-13: 978-0130220059.
- Wong, W. (2006) *Working within the framework*. *Electronic Design*. 54(2). pp.43-48. 5p.
- Wooldridge, M. (2000) *Reasoning about Rational Agents*. 1<sup>st</sup> Edition. MIT Press. USA.
- Yilmaz, G. (2006) 'Comparison of SOAP-Based Technologies: .Net Remoting and Asp.Net Web Services'. *Journal of Aeronautics & Space Technologies*. 2(4). pp.23-28. 6p.
- Yingzhou, Z. et al. (2010) 'Web Service Generation through Program Slicing', *Communications & Network*, November 2010, 2(4), pp.877-887, DOI: 10.4236/ijcns.2010.311119.
- Young, R., Lixuan Z., Prybutok, V. (2007) 'Hacking into the Minds of Hackers'. *Information Systems Management*. Fall 2007. 24(4). pp.281-287. 7p. DOI: 10.1080/10580530701585823.