Editor Dana Simian

IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT

Proceedings of the International Students Conference on Informatics, ICDD

2011, Sibiu, ROMANIA

Editor DANA SIMIAN

IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT

Proceedings of International Students Conference on Informatics
ICDD

April 7th – 9th, 2011
Sibiu, Romania

"Lucian Blaga" University of Sibiu

“Lucian Blaga” University, Sibiu, 2011

Editor Dana Simian

All papers in this volume were peer review by two independent reviewers

ISSN 2069-964X

Associated editor Laura Cacovean

Proceedings of International Students Conference on Informatics ICDD
April 7th – 9th, 2011, Sibiu, Romania

Preface

This volume contains refereed papers which were presented at the First International Students Conference on Informatics - ICDD – 2011. The conference was held between April 7th – 9th, at Sibiu, Romania. It was organized by the Department of Informatics at the Faculty of Sciences, from " Lucian Blaga" University of Sibiu.

The conference is addressed to bachelor and master level students. Conference aim is to bring together students from different universities from all over the world to discuss and present their researches on informatics and related fields (like computational algebra, numerical calculus, bioinformatics, etc) and their original results.

We thank all the participants for their interesting talks and discussions. We also thank the members of the scientific committee for their help in reviewing the submitted papers and for their contributions to the scientific success of the conference and to the quality of this proceedings volume.

April 2011

Dana Simian
Conference chairman

Motto:

"There are no limits, only your imagination"

Proceedings of International Students Conference on Informatics, ICDD – 2011
IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT
Sibiu, Romania

Editor *Prof. PhD. Dana Simian*

Scientific committee

Contents:

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Malware Analyzer

ALEXE Iulian, DODA Gheorghe, BUHA Radu, MIREA Felix
Teacher Coordinator: Lector PhD. Mircea Iosif Neamtu

Abstract

Nowadays in the technological era when the total amount of information is growing rapidly and the Internet has also become something common and insecure, most users on the web easily fall prey to viruses form accessing various links received form strangers or downloading unknown software from the Internet. Therefore we have developed a method to analyze the unknown applications, which the users run and install on their computer, and see the changes made in the computer.

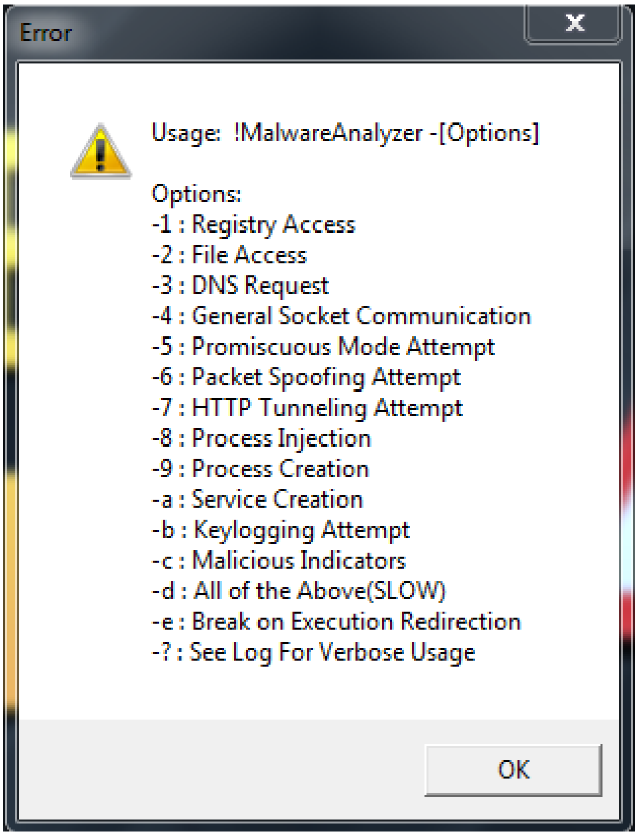
1 Introduction

Today's antiviruses, in their attempt of detecting viruses that are not in their signatures database, use a heuristic scanning method that analyzes system functions called by the running application, functions used by most viruses. If an application uses a lot of API call functions also used by viruses, then the antivirus targets the application as suspect. That's why we have implemented this method, similar to the methods used in antivirus research departments from various antivirus companies, which acquire the suspicious files submitted by users for analysis, and concludes if the application is a virus.

2 Description

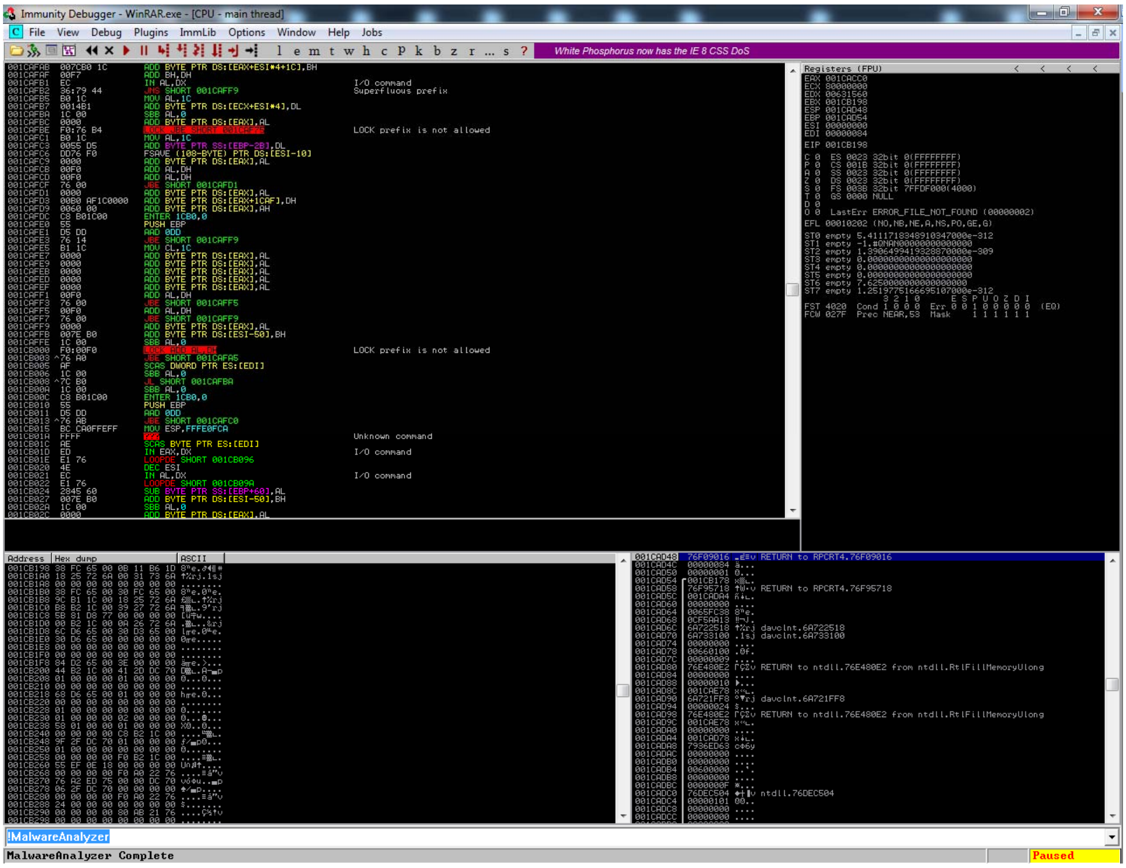
Our Malware Analyzer program was written in Python. Python is an interpreted, general-purpose high-level programming language whose design philosophy emphasizes code readability. Python aims to combine "remarkable power with very clear syntax", and its standard library is large and comprehensive. Its use of indentation for block delimiters is unique among popular programming languages. Python supports multiple programming paradigms, primarily but not limited to object-oriented, imperative and, to a lesser extent, functional programming styles. It features a fully dynamic type system and automatic memory management, similar to that of Scheme, Ruby, Perl, and Tcl. Like other dynamic languages, Python is often used as a scripting language, but is also used in a wide range of non-scripting contexts. Python is a multi-paradigm programming language. Rather than forcing programmers to adopt a particular style of programming, it permits several styles: object-oriented programming and structured programming are fully supported, and there are a number of language features which support functional programming and aspect-oriented programming (including by metaprogramming and by magic methods). Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. An important feature of Python is dynamic name resolution (late binding), which binds method and variable names during program execution. Python can also be used as an extension language for existing modules and applications that need a programmable interface. Basically Python is a programming language that lets you work more quickly and integrate your systems more effectively. Python runs on Windows, Linux/Unix, Mac OS X, and has been ported to the Java and .NET virtual machines.

We will be running the Malware Analyzer with Immunity Debugger latest version by typing "***!MalwareAnalyzer***" in the command line. After that, a menu will be displayed containing all the available commands:

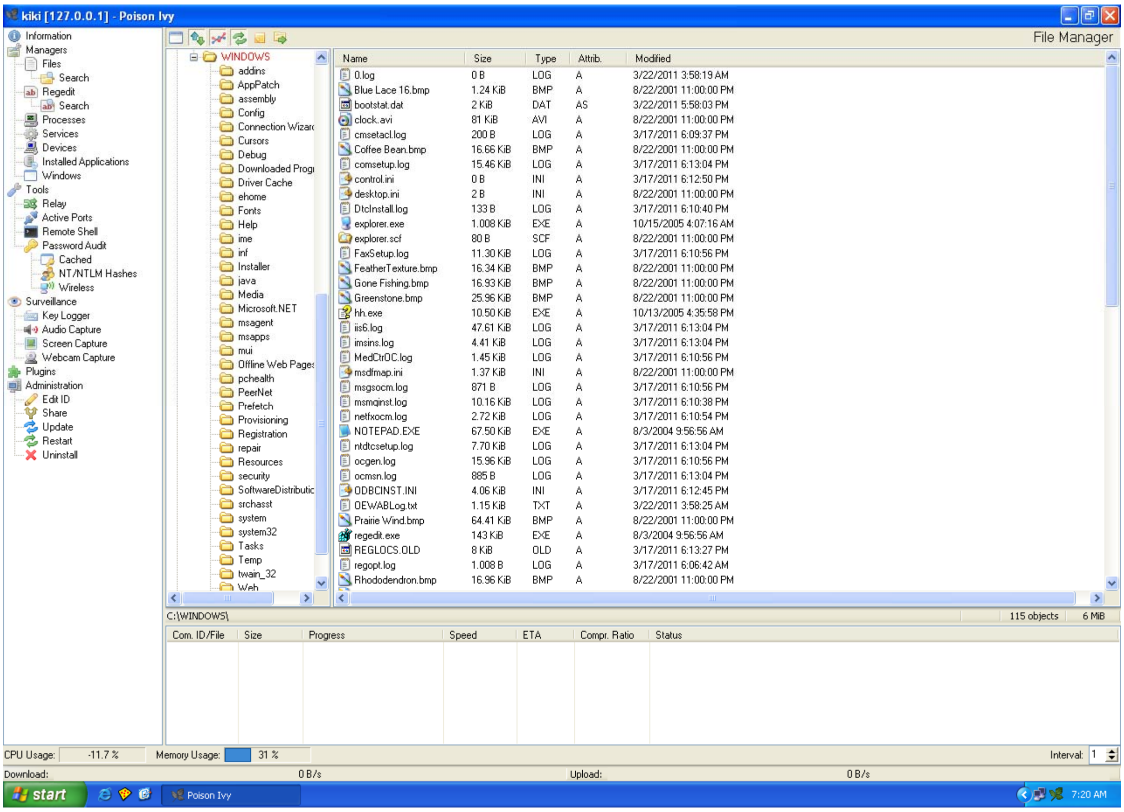


After choosing one of the available commands we will attach to the wanted process.

Immunity Debugger is a powerful way to write exploits and reverse engineer binary files. It builds on a solid user interface with function graphing; the industry's first heap analysis tool built specifically for heap creation, and a large and well supported Python API for easy extensibility. Immunity Debugger is a debugger with functionality designed specifically for the security industry. It has a very simple and understandable interface and it is a robust and powerful scripting language for automating intelligent debugging.



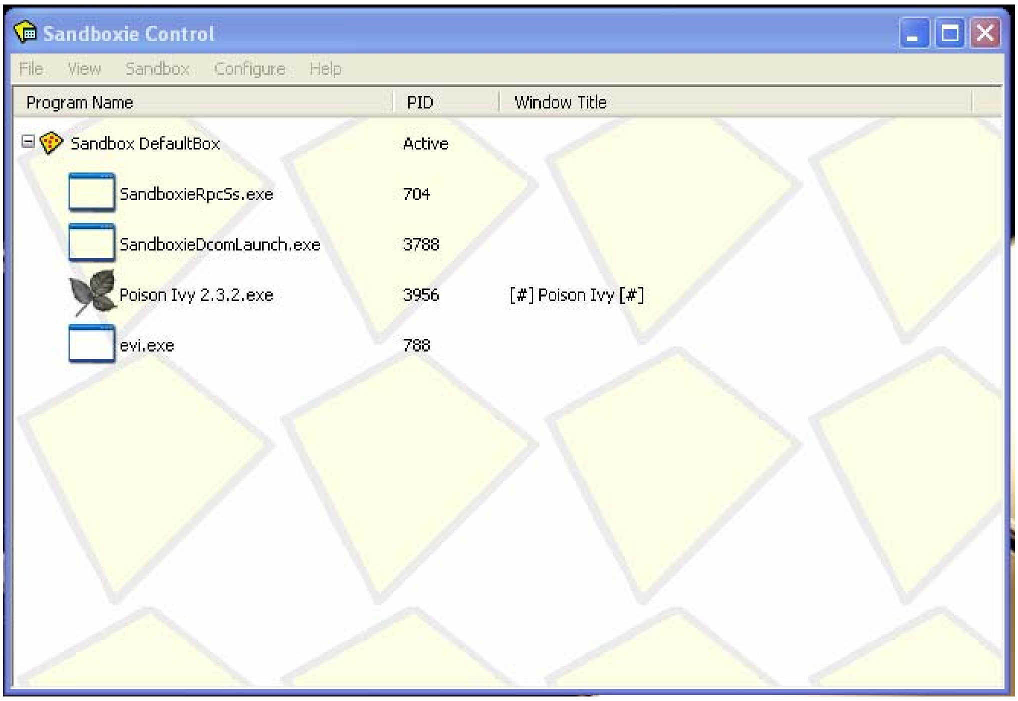
For our demonstration we will analyze a known Trojan, Poison Ivy. A Trojan is software that appears to perform a desirable function for the user prior to run or install, but (perhaps in addition to the expected function) steals information or harms the system. The term is derived from the Trojan horse story in Greek mythology. Unlike viruses, Trojan horses do not replicate themselves but they can be just as destructive. One of the most insidious types of Trojan horse is a program that claims to rid a computer of viruses but instead introduces viruses onto the computer. Trojan horses may allow a hacker remote access to a target computer system. Once a Trojan horse has been installed on a target computer system, a hacker may have access to the computer remotely and perform various operations, limited by user privileges on the target computer system and the design of the Trojan horse. Trojan horses require interaction with a hacker to fulfil their purpose, though the hacker need not be the individual responsible for distributing the Trojan horse. It is possible for individual hackers to scan computers on a network using a port scanner in the hope of finding one with a malicious Trojan horse installed, which the hacker can then use to control the target computer. In our case Poison Ivy is a remote administration tool (a RAT), which is used to remotely connect and manage a single or multiple computers with a variety of software tools such as shell control (from command prompt), computer control (power off/on/log off if remote feature is supported), registry management (query/add/delete/modify) or file management (download/upload/execute/etc.). Its primary function is for one computer operator to gain access to remote PCs. One computer will run the "client" software application, while the other computer(s) operate as the "host(s)".



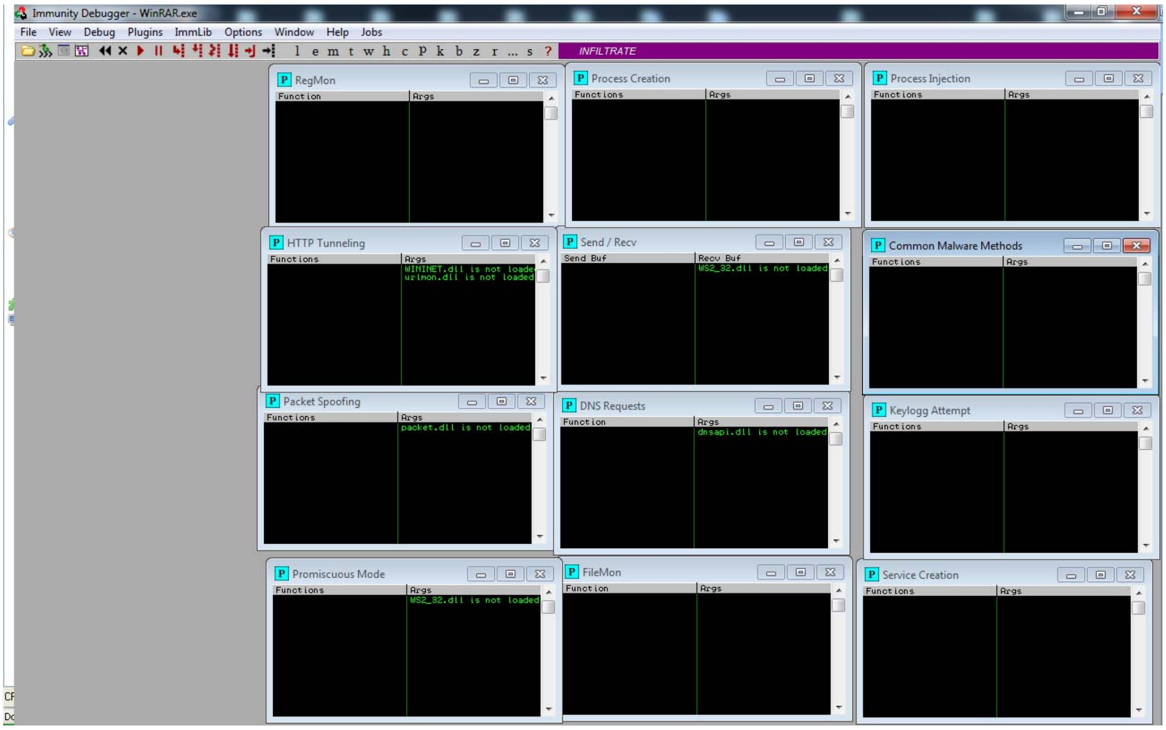
Since running viruses on a computer isn't advised, we will be running Poison Ivy in an isolated environment such as Sandboxie. Sandboxie is a proprietary sandbox-based isolation program developed for 32- and 64-bit Windows NT-based operating systems. It creates a sandbox-like isolated operating environment in which applications can be run or installed without permanently modifying the local or mapped drive.

An isolated virtual environment allows controlled testing of untrusted programs and web surfing. Sandboxie runs your programs in an isolated space which prevents them from making permanent changes to other programs and data in your computer. In computer security, a sandbox is a security mechanism for separating running programs. It is often used to execute untested code, or untrusted programs from unverified third-parties, suppliers and untrusted users. The sandbox typically provides a tightly-controlled set of resources for guest programs to run in, such as scratch space on disk and memory.

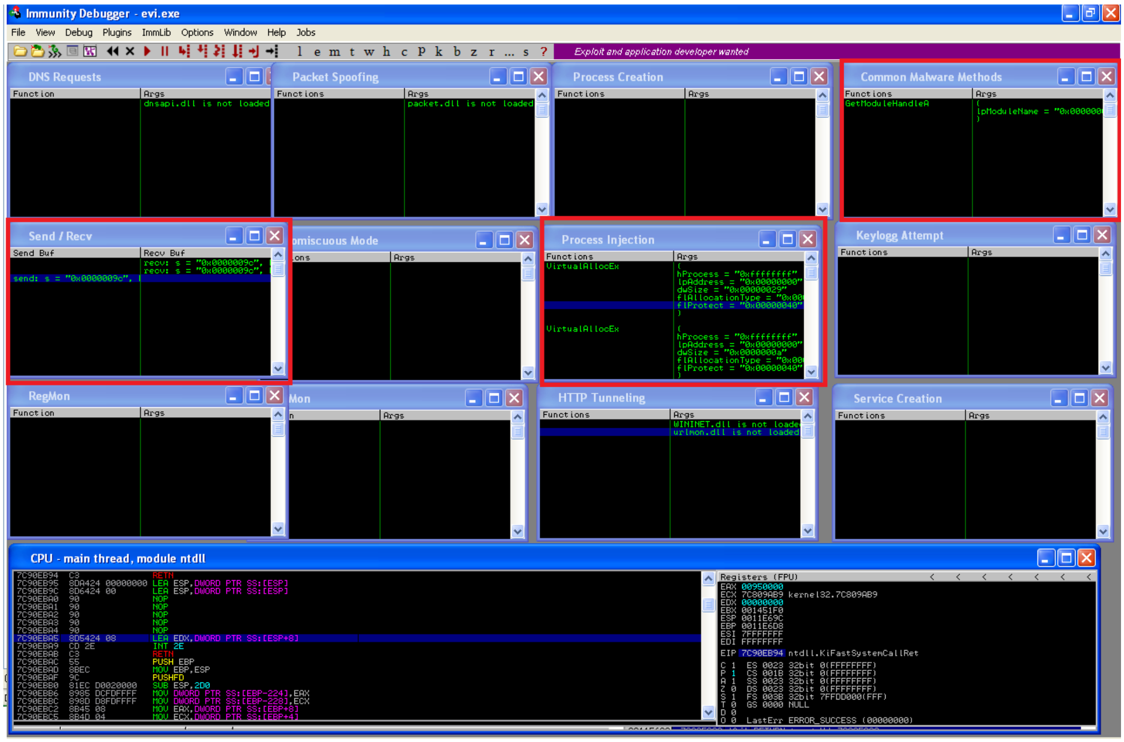
Network access, the ability to inspect the host system or read from input devices are usually disallowed or heavily restricted. In this sense, sandboxes are a specific example of virtualization.



In order to analyze the virus we must enter the "***!MalwareAnalyzer -d***" command to open all the monitoring windows:



In case the virus acts suspicious then the corresponding window will display the called API functions:



As you can see, the Malware Analyzer can easily show which API calls the virus is using, in this case the virus is using send/receive and process injection API calls. By further using the Malware Analyzer we can observe that the virus ca do harm to the computer by modifying or deleting computer data or even keylogging computer information.

3 Source Code Description

For interceptions we use a hooking method for each function as shown in the following example:

```
class RegOpenKeyExA(LogBpHook):
    def __init__(self):
        LogBpHook.__init__(self)

    def run(self,regs):
        imm = immlib.Debugger()
        regMonWin = imm.getKnowledge( "regMonWin" )

        logItems = ["RegOpenKeyExA", "("]
        regMonWin.add( regs['EIP'], logItems )

        hKey = imm.readLong( regs['ESP']+ 0x4 )
        logItems = [ "", "hKey = \"%s\"" % (hKey) ]
        regMonWin.add( regs['EIP'], logItems )

        ptr = imm.readLong( regs['ESP'] + 0x8 )
        lpSubKey = imm.readString( ptr )
        logItems = [ "", "lpSubKey = \"%s\"" % (lpSubKey) ]
        regMonWin.add( regs['EIP'], logItems )

        ulOptions = imm.readLong( regs['ESP'] + 0xC )
        logItems = [ "", "ulOptions = \"0x%08x\"" % (ulOptions) ]
        regMonWin.add( regs['EIP'], logItems )

        samDesired = imm.readLong( regs['ESP'] + 0x10 )
        logItems = [ "", "samDesired = \"0x%08x\"" % (samDesired) ]
        regMonWin.add( regs['EIP'], logItems )

        phkResult = imm.readLong( regs['ESP'] + 0x14 )
        logItems = [ "", "phkResult = \"0x%08x\"" % (phkResult) ]
        regMonWin.add( regs['EIP'], logItems )

        logItems = [ "", ")" ]
        regMonWin.add( regs['EIP'], logItems )

        logItems = [ "", "" ]
        regMonWin.add( regs['EIP'], logItems )
```

4 Conclusions

With the help of this program we can scan the danger level of the unknown applications. In the near future we wish to create an antivirus with an engine in which we develop more profoundly this type of analysis. From our point of view this kind of analysis method is more efficient because virus discovery can be made without using a signature database and it also does not need internet connection for database update.

References

[1] Florin Stoica, Laura Florentina Cacovean, Programare API, Ed. Psihomedia Sibiu, ISBN 978-973-1753-75-1, 354 pages, 2009;

[2] Peter Szor, The Art of Computer Virus Research and Defence, Addison-Wesley Professional, ISBN 978-0-321-30454-4, 744 pages, February 03 2005;

[3] Michael Ligh, Steven Adair, Blake Harstein, Matthew Richard, Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code, ISBN 978-470-61303-0, 744 pages, October 2010;

[4] Eldad Eilam, Reversing: Secrets of Reverse Engineering, ISBN 978-0-7645-7481-8, 624 pages, April 2005;

[5] Justin Seitz, Gray Hat Python: Python Programming for Hackers and Reverse Engineers, 216 pages, April 2009

ALEXE Iulian
"Lucian Blaga" University of Sibiu, Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012 Romania
E-mail: iulian.alexe@ulbsibiu.ro

DODA Gheorghe
"Lucian Blaga" University of Sibiu, Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012 Romania
E-mail: godlike_affairs@yahoo.com

BUHA Radu
"Lucian Blaga" University of Sibiu, Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012 Romania
E-mail: braduadrian@yahoo.co.uk

MIREA Felix
"Lucian Blaga" University of Sibiu, Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012 Romania
E-mail: mirea_felix@yahoo.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Security testing with advanced exploiting methods

ALEXE Iulian, VOLOSINCU Bogdan, MORARIU Stelian
Teacher Coordinators: Lector univ. dr. Mircea Iosif Neamtu,
Prof. univ. dr. Dana Simian

Abstract

The most important thing today is data security. That is why the first thing we take into consideration when we develop an application, especially when the application consists in confidential data of grave importance, security polices implementing. Our program takes care of those things by testing other applications vulnerabilities with advanced exploiting methods which find the flaws that where not programmed correctly.

1 Introduction

This article was created to review the new exploiting virus's technologies.

To create exploits and understand their action, it is necessary to know several concepts of Operating Systems such as memory organization, the structure of executable file and the phases of compilation. We will describe these principles in the lines bellow.

Compared with other types of viruses this technology is very advanced because it is rather hard for antivirus programs to detect it. With the help of exploits a lot of common applications such as Adobe Reader, Flash Player, Firefox, Internet Explorer or even media players such as VLC or Winamp ca be exploited just by opening their respective file type. After opening the exploited file the application will crash, but the exploited file can make a lot of changes in the affected computer, such as: opening a port which will allow the attacker complete access to the infected computer or it can create automatically a connection to a specific server or run a specific process, etc.
The "specific sever" mentioned above will be our application, the same application that created the exploit. This type of application vulnerability is called buffer overflow.

Basically, in computer security and programming, a buffer overflow, or buffer overrun, is an anomaly where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory.

In these kinds of situations, the data is still inserted in the memory even if it overrides the data that shouldn't. While overwriting critical data of the program, this data usually leads the program to crash. This fact alone is bad enough if we think of servers that can no longer perform their task. Worse, crushing some data, in a program that runs with super user privileges can have disastrous consequences.

Programming languages commonly associated with buffer overflows include C and C++, which provide no built-in protection against accessing or overwriting data in any part of memory and do not automatically check that data written to an array (the built-in buffer type) is within the boundaries of that array. Bounds checking can prevent buffer overflows.

1.1 Basic example

In the following example, a program has defined two data items which are adjacent in memory: an 8-byte-long string buffer, A, and a two-byte integer, B. Initially, A contains nothing but zero bytes, and B contains the number 1979. Characters are one byte wide.

variable name	A								B	
value	[null string]								1979	
hex value	00	00	00	00	00	00	00	00	07	BB

Now, the program attempts to store the null-terminated string "excessive" in the A buffer. By failing to check the length of the string, it overwrites the value of B:

variable name	A								B	
value	'e'	'x'	'c'	'e'	's'	's'	'i'	'v'	25856	
hex	65	78	63	65	73	73	69	76	65	00

Although the programmer did not intend to change B at all, B's value has now been replaced by a number formed from part of the character string. In this example, on a big-endian system that uses ASCII, "e" followed by a zero byte would become the number 25856. If B was the only other variable data item defined by the program, writing an even longer string that went past the end of B could cause an error such as a segmentation fault, terminating the process.

1.2 Methodology

Taking advantage of this program's buffer overflow weakness, we will try to take control of the victim's computer by injecting a shellcode in the overflowed buffer. Shellcode is defined as a set of instructions injected and then executed by an exploited program. Shellcode is used to directly manipulate registers and the functionality of an exploited program. We can of course write shell codes in the high level language they might not work for some cases, so assembly language is preferred for this.

We write shellcode because we want the target program to function in a manner other than what was intended by the designer. One way to manipulate the program is to force it to make a system call or syscall. A system call is how a program requests a service from an operating system's kernel that it does not normally have permission to run. System calls provide the interface between a process and the operating system. Most operations interacting with the system require permissions not available to a user level process, e.g. I/O performed with a device present on the system, or any form of communication with other processes requires the use of system calls.

2 STE Development - Ruby and Java

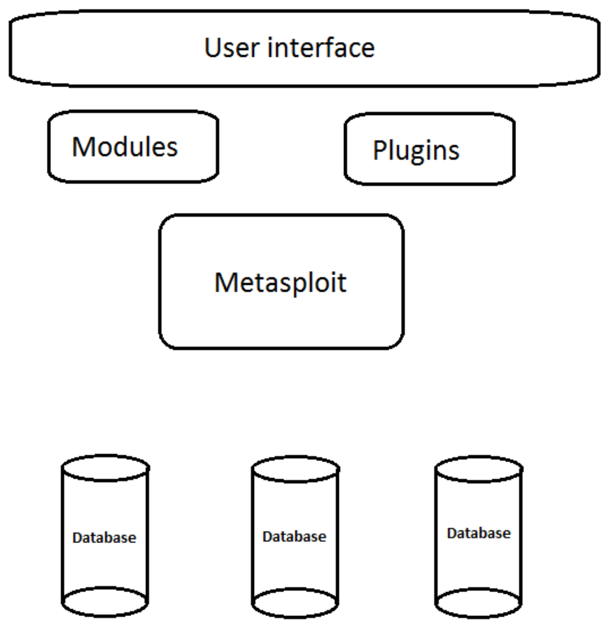
2.1 Application Introduction

The **STE** (***Security Testing and Exploit***) is a great system that beside the power of the tools for penetration testers and security researchers is that is written in Ruby which is a dynamic language and gives a lot of flexibility for extending the framework functionality with new features being written in Ruby itself or extensions in Java or C that includes calls for embedding Ruby in software, for use as a scripting language.
The most important piece of the application architecture is the Ruby Extension Library (Rex) which is designed to have no dependencies other than what comes with the default Ruby install.

2.2 Application Description

The framework is structured into different parts. The low level area being the framework core that is responsible for implementing all of the required interfaces that make possible the interaction with exploit modules, sessions, and plugins. This core library is extended by the framework base library which is designed to provide simpler wrapper routines for dealing with the framework core as well as providing utility classes for dealing with different aspects of the framework, such as serializing module state to different output formats.

Above these two layers is the UI component developed in Java language which eases the work with the framework and links the application with the database server that is the place where are stored all the vulnerabilities known at a certain point in time and that can be tested on the local system and secured.



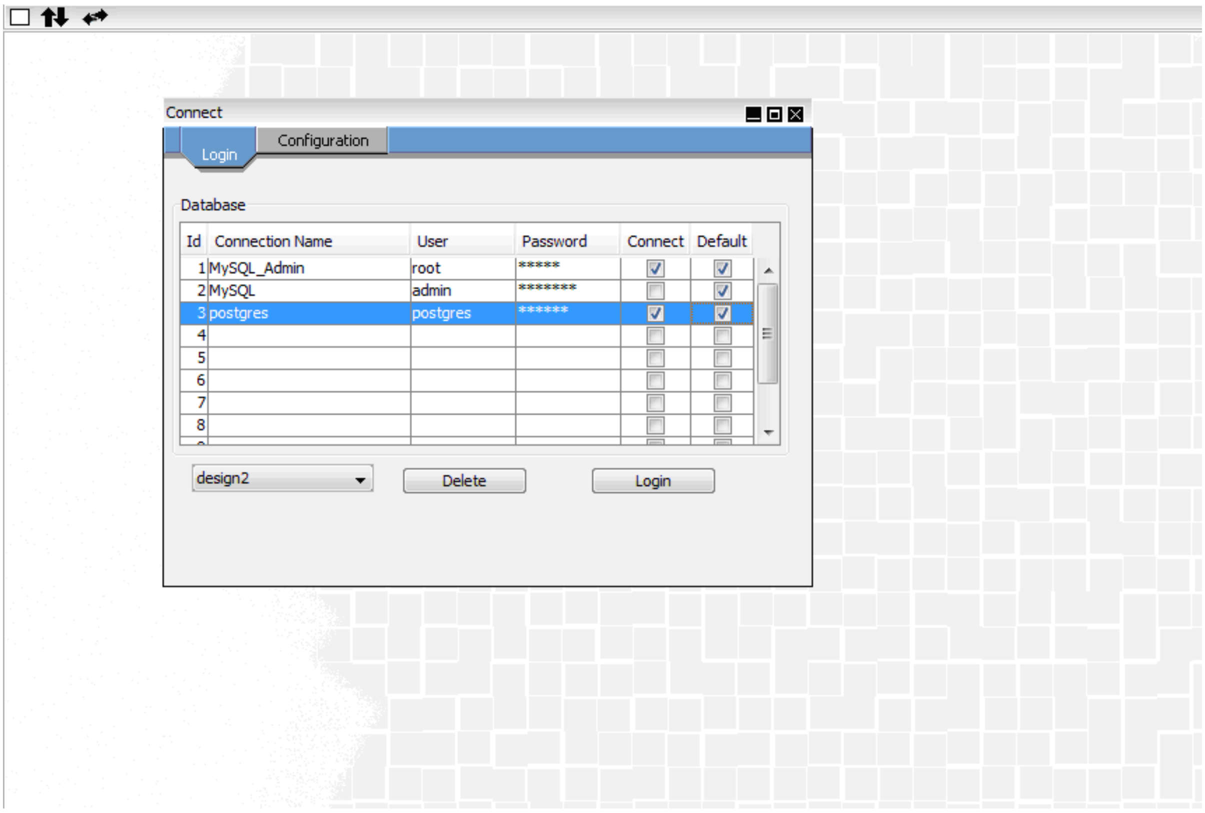
2.3 Operations Management

The operations are handled by the framework managers that are responsible for some of the basic aspects of the framework, such as module and plugin management.

Module Types:

1) ENCODER
2) EXPLOIT
3) NOP
4) AUXILIARY
5) PAYLOAD

Unlike modules, the plugins are meant to add features to the application or to change the behavior of existing aspects. Plugins have a very loose definition in terms of the scope in which they can operate. For example, a plugin could add an entirely new module type for use by the application.



2.4 Session management

The session manager is used to track sessions created from within a framework instance as the result of an exploit succeeding. The purpose of sessions is to expose features to a programmer that allow it to be interacted with. For instance, a command shell session allows programmers to send commands and read responses to those commands through a well-defined API.

2.5 Job management

Each framework instance supports running various tasks in the context of worker threads through the concept of jobs.

2.6 Utility Classes

Some classes in the framework core are intended to be used to make certain tasks simpler without being out of scope of the core aspects of the framework.

3 Conclusion

In the end every self respecting company will need this kind of solution to prevent hacking attempts by testing system vulnerabilities. That’s why we developed an easy to use java application which meets the user’s needs in every aspect. In the near future we want to implement more plugins and modules in our application.

4 Reference

[1] David Maynor, Metasploit Toolkit for Penetration Testing, Exploit Development, and Vulnerability Research, 272 pages, 2007,
[2] Johnny, LongPenetration tester's open source toolkit, 204 pages, 2005,
[3] Thomas Wilhelm, Professional Penetration Testing: Creating and Operating a Formal Hacking Lab, 504 pages, 2009
[4] James C. Foster, Mike Price, Stuart McClure, Sockets, shellcode, porting & coding: reverse engineering exploits and tool coding for security professionals, 667 pages, 2005
[5] James C. Foster, Buffer overflow attacks: detect, exploit, prevent, 497 pages, 2005
[6] Jack Koziol, The shellcoder's handbook: discovering and exploiting security holes. 620 pages, 2004

ALEXE Iulian
"Lucian Blaga" University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
Romania
E-mail: iulian.alexe@ulbsibiu.ro

VOLOSINCU Bogdan
"Lucian Blaga" University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
Romania
E-mail: bogdan_voloshincu@yahoo.com

MORARIU Stelian
"Lucian Blaga" University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
Romania
E-mail: modev_st@yahoo.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

A software implementation of a computer model and a corresponding symbolic assembler

Jovo Arezina
Teacher Coordinator: Milan Tuba

Abstract

Models of hypothetical computers are used for experiments with computer architecture. In this paper a computer model called *MTU1*, its assembly language and a software implementation called *MTU1 Emulator* are described. The *MTU1* computer model is intended to be used as an educational tool for demonstrating computer architecture principles. It does so using a carefully selected instruction set which is both small and easy to comprehend, yet able to express real world historic and current computer architecture concepts. *MTU1 Emulator* is a program which can be used to write and execute *MTU1* code. It features a two pass assembler which supports the use of symbols. It also displays useful information about CPU and memory registers and instructions being executed, with support for several numerical systems and ASCII representation. Two code examples for the *MTU1* computer model are also presented.

Keywords: Computer model, CPU emulator, Assembler

1 Introduction

Computer models are often described as “educational abstract machine”, “hypothetical computer” or “instructional model”. All these descriptions refer to a concept of an imaginary CPU defined as an instruction set and by the number and intended use of registers. They are widely used as educational tools in computer architecture courses [1].

There are many computer models in use today [2], [3]. One of the oldest and most widely known is the *Little Man Computer* by Stuart Madnick, professor at MIT. The memory is viewed as a set of mailboxes and the CPU as a little man who processes the “mail” by using input and output boxes to communicate with the memory. There are a few simulators available for it on the Internet [4].

Donald Knuth used a computer model called *MIX* in his monograph “*The Art of Computer Programming*”. It was superseded by *MMIX* in later editions of the book [5]. Software implementations called *MIXware* and *MMIXware* for both versions are made by Knuth, but a few other emulators also exist.

Computer models *NAR1* and *NAR2* by professor Nedeljko Parezanović [6] have been successfully used in *Computer architecture* courses in Serbia. The influence of the minicomputer era on these models is evident and as such they might not be the best choice for teaching modern computer architecture courses [7], [8], [9], [10], [11].

The computer model described in this paper is called *MTU1* was introduced by professor Milan Tuba in the *Computer organization and architecture* courses at the Faculty of Computer Science at Megatrend University in Belgrade, as well as Faculty of Mathematics, Computer Science Department, University of Belgrade, Serbia. This is a minimalistic model with eight instructions and three addressing modes.

The advantage of creating one's own model is the ability to precisely adjust it to the requirements of the curriculum. The model can have a mnemonic language defined to aid program coding.

Development of *MTU1 Emulator* was motivated by need to test and confirm knowledge gained during studies by being able to run *MTU1* programs and show useful information about their execution. It is fully original work as the intent was to provide a familiar user interface with as much information possible presented to the user rather than simply making an execution environment.

The main difference compared to a simulator is that it models the CPU only to a level which was determined to be useful to fulfill the original requirements. A simulator would require modeling of internal states to closely mimic a CPU implemented in hardware.

Section 2 of this paper describes the *MTU1* computer model with details about the instruction set, addressing modes and memory. Details about the assembly language, syntax and machine code can be found in Section 3. *MTU1 Emulator* is briefly described in Section 4 with focus on the different parts of the main window. Two *MTU1* program examples are given in Section 5 to highlight some of the *MTU1* and *MTU1 Emulator* features and demonstrate usage.

2 MTU1 computer model

The *MTU1* CPU is based on the von Neumann architecture and would fall in to SISD category of the Flynn taxonomy. It has one general purpose register (an accumulator) and a program counter. The instruction set is orthogonal and consist of eight instructions and three addressing modes. There are two one bit flags used by conditional instructions. CPU word is one byte in length which is also the width of the address and data lines. It uses RAM.

Only two registers are directly accessible by the user. The accumulator (ACC) holds the value which is a result of a CPU operation or is read from memory and the program counter (PC) which holds the address of the next instruction in the memory. There is a load instruction (LDA) which transfers data from memory to ACC and a store instruction (STA) which does the opposite. A halt instruction (HLT) stops the program execution.

If raised during an instruction execution a carry flag (C) indicates there was not enough space available to hold the result in the ACC. Zero flag (Z) is raised iff the ACC value is equal to zero.
A programmer can change the PC value by using jump instructions. “Jump if not carry” (JNC) alters the PC if C is low and so does “Jump if not zero” (JNZ) if Z is low. The operand of these conditional jumps represents the address of the next instruction to be executed.

Three more instructions are defined to enable writing of useful programs. The ADD instruction sums a value from memory with the value in ACC. It raises the C flag if there was carry, and lowers it otherwise. XOR performs a bitwise exclusive disjunction using the ACC and a value from memory. It always lowers the C flag. A shift left (SHL) performs a bitwise shift of the ACC value by one bit. It raises or lowers the C flag to match the value of the most significant bit in the original value.

MTU1 supports direct, immediate and indirect addressing modes. Maximum memory space it can use is 256 bytes.

All instructions supported by *MTU1* are listed in Table 1. The table also shows the possible effects on different CPU registers and flags (read or write), the number of required operands and the overall effect of the instruction.

Table 1: MTU1 instructions

INSTRUCTION	PC	ACC	C	Z	OPERANDS	EFFECT
HLT	-	-	-	-	0	halts execution
LDA	-	W	-	W	1	ACC = MEM[address]
STA	-	R	-	-	1	MEM[address] = ACC
ADD	-	R/W	W	W	1	ACC = ACC + MEM[addr.], C = 0 or 1
XOR	-	R/W	W	W	1	ACC = ACC $\oplus$ MEM[address], C = 0
SHL	-	R/W	W	W	0	ACC = ACC << 1, C = 0 or 1
JNZ	W	-	-	R	1	iff Z = 0, PC = MEM[address]
JNC	W	-	R	-	1	iff C = 0, PC = MEM[address]

3 Assembly language

Program lines consist of either one instruction or data declaration. Instruction mnemonics consist of three case insensitive letters which are chosen to immediately evoke their intended purpose in the user's mind. If required the mnemonic is followed by white space and an operand, with or without markings which determine the address mode to be used. Symbols can be used to identify consequent memory registers containing instructions or data.

The assembler in *MTU1 Emulator* is a two pass variant so there is no need to define separate code and data sections or to declare a symbol prior to its use [1]. Support for *C* style comments is also implemented. Programs consisting solely of data are also valid which enables writing in machine code.

Operands and data can be entered in decimal, binary (using a 0b prefix) or hexadecimal (using a 0x prefix) system. Addition and subtraction of numbers and symbols is supported.

For all instructions except for the two jump instructions a number sign before the operand (#X) signifies usage of the immediate addressing mode. If an operand is entered between square brackets ([X]) the indirect addressing mode is used. Operand used alone indicates direct addressing mode.

Jump instructions (JNC and JNZ) have their addressing modes "shifted to the left" so JNC X would function in immediate mode and JNC [X] in direct mode. The assembler reports an error if JNC #X is used. This is in keeping with most real world assembler implementations.

3.1 Syntax

The following rules are enforced due to CPU specifications and the way the *MTU1 Emulator* assembler is made:

- A symbol must be an uninterrupted string consisting of English alphabet letters, numbers and/or an underscore ending with a colon. It must start with a letter or an underscore and must be separated by at least one whitespace character from the next character in a program line.
- If data is labeled with a symbol it must be initialized to some value.
- Instructions that require an operand must be separated by at least one whitespace character from the operand, its arithmetic sign or addressing mode marking.
- All characters following two consecutive slash characters (//) are considered to be comments.
- Lines consisting solely of whitespace are ignored.

3.2 Machine language

Instructions are formatted in a way that makes them easily recognizable if viewed in the hexadecimal number system. The most significant nibble stores the operation code, while the least significant nibble stores the addressing mode.

HLT and SHL are the only instructions that do not require an operand and as such do not use an addressing mode. Nevertheless the least significant nibble must be zero for them to be executed properly.

Machine code for the HLT instruction is equal to zero which is the value memory registers reset to. That way the CPU is halted if execution reaches an uninitialized part of memory.

Immediate mode STA is an illegal instruction as it would produce an undefined state. Based on the implementation the operand would be ignored and the value of ACC would be copied in its place in memory.

Table 2 shows the instructions in hexadecimal format. If an operand is required the CPU expects its address to be in a register immediately following the one containing the operation.

Table 2: MTU1 machine language

INSTRUCTION	ADDRESSING MODE		
	Direct	Immediate	Indirect
HLT		00	
LDA	11	10	12
STA	21		22
ADD	31	30	32
XOR	41	40	42
SHL		50	
JNZ	61	60	62
JNC	71	70	72

4 MTU1 Emulator

Code written in the CodeBox (Fig. 1) is highlighted to emphasize different parts of a program line. Errors encountered during translation to machine code are shown in the ASM Log. Machine code is written directly to memory upon compilation. The memory contents and the symbols used in a program are shown on the right-hand side of the program window. Additional info about a memory register can be viewed in the Register Info section. The contents of CPU registers can also be viewed. The memory and CPU view can be made to present values in binary, decimal, octal, hexadecimal and ASCII. The status bar also shows useful information such as cursor position in code and CPU execution statistics.

Program execution can be done one instruction at a time or run until a halt condition or an error is reached. CPU Log shows information about every executed instruction such as the value of PC, operation and addressing mode used. The memory view indicates read and write actions with color during program execution, and the executed instruction is highlighted in the CodeBox.

A detailed help document is not currently available within the program, but a quick reference of CPU instructions in mnemonic and machine language form is provided.

There are very few options provided and are self-explanatory.

MTU1 Emulator was written using the *Microsoft Visual C++ 2008 Express* integrated development environment. *C++/CLI* was chosen as the language because it provides easy access to *.NET framework* but also opens a clear path to native *C++* and *x86* assembly language if needed in future development.

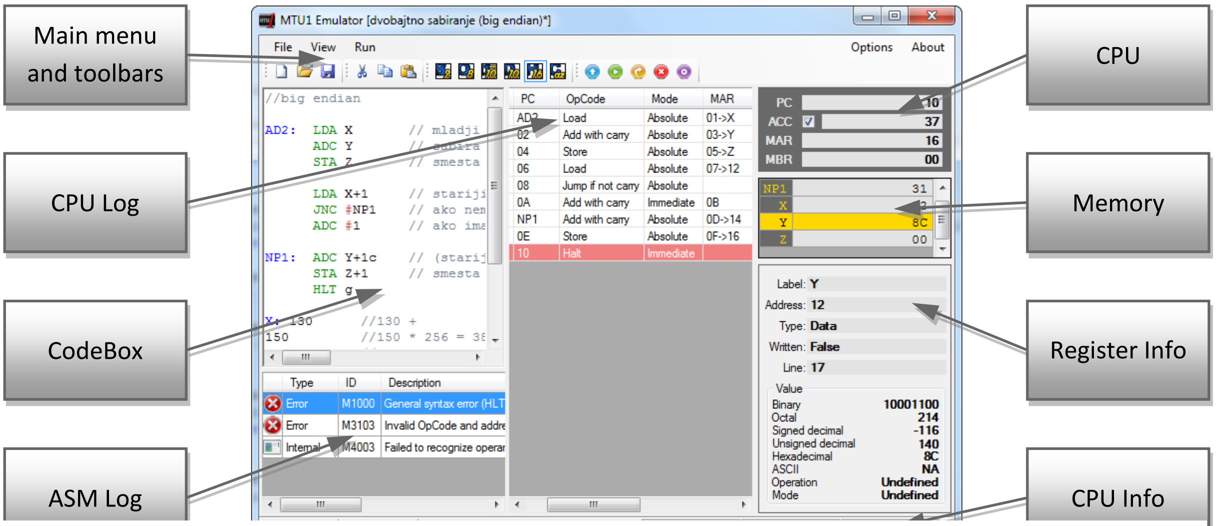


Figure 1: MTU1 Emulator main window

5 Code examples

The purpose and operation of programs is explained within the comments of the programs themselves. Two byte addition is an example of applied numeric systems knowledge. Summing the elements of an array shows the low level anatomy of a loop and pointer usage via indirect mode addressing.

5.1 Two byte addition

```
// TWO BYTE ADDITION (little endian)

AD2:  LDA X+1       // least significant byte of X
      ADD Y+1       // added to the least significant byte of Y
      STA Z+1       // least significant byte of the result is stored

      LDA X         // most significant byte of X
      JNC NP1       // if there is no carry there is no need to add one
      ADD #1        // add one if there was carry

NP1:  ADD Y         // (most significant byte in ACC) is added to the most significant byte of Y
      STA Z         // most significant byte of the result is stored
      HLT

X: 130     //130 * 256
150        //+ 150 = 33430
Y: 140     //140 * 256
160        //+ 160 = 36000
Z: 0       //15 * 256
           //+ 54 = 3894
           //+ C (2^16 = 65536) = 69430
           //33430 + 36000 = 69430 T
```

5.2 Sum of array elements

```
// SUM OF ARRAY ELEMENTS

ADN:  LDA #0
      STA S          // sum is zero at the begining

      LDA N          // array lenght
      STA i          // iterator (equal to the array lenght)

      LDA #ARR       // first element address
      STA PX         // pointer to the first element

      LDA i
      JNZ CKL
      HLT            // end if array has no elements (iterator equals zero), otherwise procede with the cycle

CKL:  LDA S
      ADD [PX]
      STA S          // add current sum with the value indicated by the pointer
      LDA PX
      ADD #1
      STA PX         // increment the pointer

      LDA i
      ADD #0xFF
      STA i          // decrement the iterator

      JNZ CKL
      HLT            // end if iterator equals zero, otherwise go to cycle

N:    10   // array lenght
i:    0    // iterator
ARR:  1    // first array element
      2
      3
      4
      5
      6
      7
      8
      9
      10
PX:   0    // pointer
S:    0    // sum
```

6 Conclusion

The *MTU1* architecture is a powerful educational tool because it is relatively easy to learn due to its small but carefully selected instruction set, yet enables demonstration of important and most widely used concepts in computer architecture. These include different addressing modes to facilitate intuitive memory manipulation, computer programming basics, the use of pointers, numerical system principles and conversion, etc.

It can be easily altered or extended so as to find the appropriate balance required in different educational scenarios. If an even more limited subset of available instructions and addressing modes is enforced during the early stages of the course the thought process behind adding CPU capabilities can be easily demonstrated and understood.

A clear connection between real world instruction sets and higher languages can also be established. It can then be used to deepen understanding of seemingly distant concepts.

The *MTU1 Emulator* program was designed to offer insight in the inner workings of the architecture to aid better understanding but also present a familiar user interface to modern computer users. In this way it differs from most other similar programs which require a learning curve and are not as informative. Most if not all currently available simulators are implemented as Java applets or programs. While this is not a flaw in itself the initial time and effort required to familiarize oneself with often exotic user interfaces may be off-putting to students.

In some aspects *MTU1 Emulator* offers more information to its users then a real assembly development environment. This is made possible by being in total control of the virtual CPU and memory and thus being able to log every change however minute and uninteresting in a real world situation but valuable in a learning process.

Upcoming versions will correct some of the known problems arising from development environment and early design choices but not fully remedied in the current version. A version with 16 instructions which is more appropriate for more advanced programming examples is already in use and will be fully implemented.

Future directions of study are *MTU1* to *x86* code conversion, implementing a computer architecture design capability, macro and pseudo higher language support, plugin support, etc.

The ultimate goal is a program which can emulate an arbitrary architecture easily described by a script language or using digital logic design. Such a program would be a useful tool for consolidating knowledge gained in several different study areas.

References

[1] Gregory S. Wolffe, William Yurcik, Hugh Osborne, Mark A. Holliday, *Teaching computer organization/architecture with limited resources using simulators*, 33rd SIGCSE technical symposium on Computer science education, Vol. 34, No.1, 2002, pp. 176-180

[2] Grant Braught, David Reed, *The knob & switch computer: A computer architecture simulator for introductory computer science*, Journal on Educational Resources in Computing (JERIC), Vol. 1, No. 4, 2001, pp. 31-45

[3] Linda Null, Julia Lobur, *MarieSim: The MARIE computer simulator*, Journal on Educational Resources in Computing (JERIC), Vol. 3, No. 2, 2003, pp. 1-29

[4] Gedupudi, R., *Simulation of Little Man Computer*, Master's Degree Project Documentation, Department of Applied Computer Science, Illinois State University, 2000

[5] Donald E. Knuth, *The Art of Computer Programming*, Vol. 1, No. 1: MMIX -- A RISC Computer for the New Millennium (Art of Computer Programming), Addison-Wesley Professional, 2005

[6] Nedeljko Parezanović, *Osnovi računarskih sistema*, Beograd, Izdavačko preduzeće "Nauka", 1992.

[7] Andrew S. Tanenbaum, *Structured Computer Organization*, Prentice Hall, 2005.

[8] Nicholas Carter, *Computer Architecture*, McGraw-Hill, 2001.

[9] M. Morris Mano, Charles Kime, *Logic and Computer Design Fundamentals*, Prentice Hall, 2007.

[10] M. Morris Mano, *Computer System Architecture*, Prentice Hall, 1992.

[11] John L. Hennessy, David A. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann, 2006.

Jovo AREZINA
Megatrend University
Faculty of Computer Science
Bulevar umetnosti 29, 11070 Belgrade
SERBIA
E-mail: jovo.arezina@gmail.com

Milan TUBA
Megatrend University
Faculty of Computer Science
Bulevar umetnosti 29, 11070 Belgrade
SERBIA
E-mail: tubamilan@ptt.rs

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Modeling of risk assessment methods during the implementation of corporative information systems at the enterprises

Ekaterina Avdeeva
Teacher Coordinator: Vladimir Chernov

Abstract

A general description of corporate information systems and features of its implementation at the enterprises are considered. Major success factors of the implementation of corporate information systems and problems and difficulties of the implementation project are described. General methods of risk assessment are analyzed.

New fuzzy models of risk assessment during the implementation of corporate information systems are offered. Fuzzy model of SWOT-analysis of riskiness assessment of the implementation project is considered. Fuzzy model of an alternative choice of corporate information systems is described.

1 Introduction

Success of any company in the conditions of market economy directly depends on how fast it can react and adapt to changes of the market environment. To major success factors in a competitive struggle, besides a well thought-out strategy and highly skilled specialists, it is necessary to carry and corporate information systems, which implementation furthers increase of the enterprise efficiency and puts it on a new level of development [6].

Corporate information systems are an effective instrument of enterprise management that makes possible not only to optimize the decision-making process and to receive reliable information about an activity of enterprise promptly in real-time, but to reduce costs at the expense of enterprise's flexibility increase and to adapt to economic and business changes.

The implementation of corporate information systems at the enterprises is a very complex and laborious process, success of which depends on different factors. A low-quality organization and realization of the implementation stages and non-observance of success factors lead to a failed project of corporate information systems implementation. Such projects usually have a lot of risks that are necessary to be identified, estimated and managed in advance [6].

Risk assessment during the implementation of corporate information systems at the enterprises is a topical issue, because a problem of risks during the implementation of such systems is insufficiently known. It is proved by not much present published works and articles in the described field that basically are descriptive (there are only recommendations about the implementation of corporate information systems). Besides at the present there is no specific methodology of risk assessment during the implementation of corporate information systems at the enterprises and the problem of risk assessment remains undecided.

2 Features of the implementation of corporate information systems at the enterprises

The implementation of corporate information systems is a very difficult process, which success directly depends on an organization and realization of the implementation project. A lot of consulting companies mark out the following major success factors of the implementation of corporate information systems at the enterprises: interest and participation of company management in the project (20 %); presence and observance of the implementation plan (19 %); clear purposes and requirements (16 %); participation of specialists from the enterprise (16 %) and others (Fig. 1).

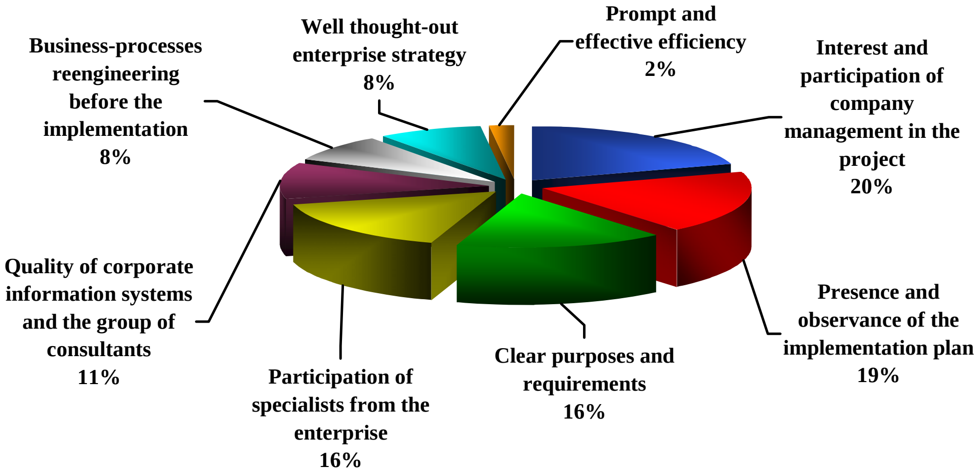


Fig. 1: Major success factors of the implementation of corporate information systems at the enterprises

A low-quality realization of the implementation stages and non-observance of success factors lead to unsuccessful implementation of corporate information systems, which basic reasons are the following: lack of management attention and consideration for the implementation project (40 %); lack of clear purposes of the implementation project (17 %); non-formalization of business-processes of the enterprise (14 %); unavailability of the enterprise to changes (12 %) and others (Fig. 2).

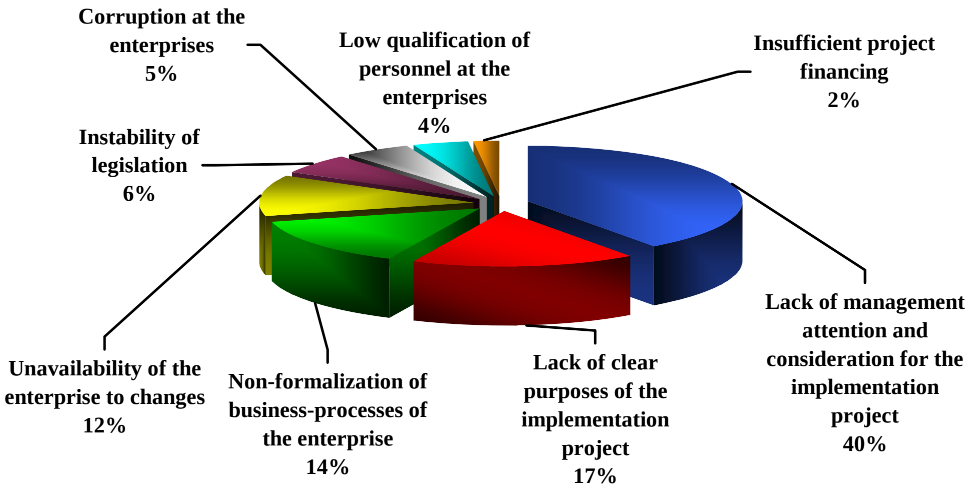


Fig. 2: Problems during the implementation of corporate information systems at the enterprises

Problems and difficulties during the implementation of corporate information systems make an essential impact on realization of the implementation project which basically comes to the end with a failure. According to statistics, 31 % of the implementation projects come to the end with a failure; 53 % of the implementation projects come to the end with the excess expenditure of a budget in 2 times; only 16 % of the implementation projects meet a schedule and a budget; in all completed projects only 61 % of assigned tasks have been realized. It entails not only with a wrong organization of the implementation project and an unavailability of the enterprise to changes, but also with an underestimation of risk factors and an importance of risk management process [4, 5].

Thus, the implementation of corporate information systems at the enterprise has a lot of risks. Therefore any implementation project should involve a risk management which purpose is not only risk identification, its estimation and decision-making on its minimization, but also an increase of the project efficiency. Timely identification of risks and its factors which arise during the implementation of corporate information systems at the enterprises and negatively influence a realization of the implementation project and its results, will allow to eliminate project lacks, thereby to lower a probability of an unsuccessful completion of the implementation project of corporate information systems [1, 3].

3 Fuzzy models of risk assessment during the implementation of corporate information systems

On the analysis of general methods of risk assessment (probabilistic method, expert judgment method, analog method, sensitivity analysis, scenario analysis, decision-tree evaluation method and Monte Carlo simulation) it has been found out that all methods are difficult to be used for the risk assessment of corporate information systems implementation, because in these methods a lot of statistical information is used. During the implementation of corporate information systems there is only up-to-date information. Therefore described methods of risk assessment leave out of account the specificity of corporate information systems implementation processes and it is necessary to create new models of risk assessment during the implementation of corporate information systems at the enterprises. These new models are being devised on basis of fuzzy sets, because fuzzy sets can process and manipulate qualitative expert evaluations [2, 3].

One of the elementary practical expert methods of risk analysis is SWOT-analysis. It is a qualitative method which is based on a comparison of opposite project characteristics. The classical procedure of SWOT-analysis, despite simplicity of its realization and visualization of its results, has an important shortcoming: use of numerical estimations. There are a lot of situations when project characteristics are easier to be estimated with qualitative evaluations, not with numbers, for example, a qualification of the implementation group: low, average, high. Besides, the traditional SWOT-analysis doesn't consider an importance and a possibility of project characteristics realization, and also a relation of a decision-maker to the possibility of their realization.

New procedure of SWOT-analysis of the corporate information systems implementation project is the following. On the basis of an optimistic position (a furl of fuzzy sets is fulfilled using "union" operation) or a pessimistic position of a decision-maker (a furl of fuzzy sets is fulfilled using "intersection" operation) a furl of Strengths, Weaknesses, Opportunities and Threats is separately fulfilled. Then a furl of positive project characteristics (Strengths and Opportunities) and negative project characteristics (weaknesses and threats) is fulfilled. If a power of fuzzy set of positive project characteristics is higher than a power of fuzzy set of negative project characteristics, such project can be realized. And on the contrary, if a power of fuzzy set of negative project characteristics is higher than a power of fuzzy set of positive project characteristics, such project is very risky and undesirable to realization. Fuzzy model of SWOT-analysis of the implementation project is not only based on fuzzy sets for processing of qualitative expert evaluations, but also considers the different relations of a decision-maker to the realization of project characteristics. Combinations of a decision-maker positions make possible to simulate different variants of development of the implementation project and to make an estimation of the project riskiness before the beginning of its realization.

One of the main stages of the implementation project of corporate information systems is the choice of system for implementation which influences a risk of unsuccessful completion of the implementation project. In a choice of corporate information systems for implementation there is an uncertainty in choice of requirements. It can lead to that during the realization of one alternative (corporate information systems) there is a redundancy of requirements and, hence, the excessive expenses. Or on the contrary, the underestimated requirements can lead to a choice of an unsuccessful alternative. Therefore it is expediently to choose corporate information systems on the basis of a necessary and possible level of conformity of alternatives to the requirements.

The best corporate information system for implementation is such alternative which a distinction of requirements importance assessments and conformity assessments of alternatives to requirements is minimum, a level of identity of an alternative to requirements is maximum, necessary and possible levels of conformity of an alternative to requirements are maximum and a subjective confidence is maximum. The considered method of an alternative choice of corporate information systems also can be realized, when estimations of importance of requirements and estimations of conformity of alternatives to requirements are qualitative.

The fuzzy model of SWOT-analysis of the implementation project of corporate information systems is simple and doesn't demand special skills and expenses. But this model of project riskiness estimation may seem too simple to the analyst. Therefore more difficult model with a detailed analysis is needed. In that case the analyst can use the model of an alternative choice of corporate information systems which will allow not only to choose the best system for implementation at the enterprises, but also to reduce the risk of unsuccessful realization of the implementation project.

At the present the research work in the described field is still continuing. It is planning to create the third model of risk assessment during the implementation of corporate information systems. This model will evaluate risks on basis of risk factors assessments.

4 Conclusions

In the aggregate developed models of risk assessment during the implementation of corporate information systems is a new specific methodology of risk assessment during the implementation project. These methods can be used at the enterprise without involvement of external consulting firms that in turn reduces cost of the implementation project and lowers probability of unsuccessful completion of the implementation of corporate information systems.

Described models of risk assessment during the implementation of corporate information systems can be used by heads of enterprises that are planning the implementation of corporate information systems, the implementation project managers and consulting firms.

In future on the basis of developed models it is planning to create software which will not only estimate risks of the implementation project but also assess efficiency of risk management.

Acknowledgement: This paper is founded from the research Grant GBO-669.

References

[1] Avdeeva E.S., Chernov V.G. Classification of risks during the implementation of corporate information systems at the enterprises. *Herald of the branch of the All-Russian Distance Institute of Finance and Economic*, IV: 171-173, 2010.

[2] Avdeeva E.S., Chernov V.G., Gradusov D. A. Review of risk assessment methods during the implementation of corporate information systems at the enterprises. *Informatics: problems, methodology, and technology*, I: 22-27, 2010.

[3] Avdeeva E.S. Risk assessment and management during the implementation of corporate information systems at the enterprises. *Information security*, VII: 65-68, 2010.
[4] Grekul V.I., Korovkina N.L., Denishenko G.N. *Management of the information systems implementation.* INTUIT, Moscow, 2009.
[5] Mazur I.I., Shapiro V.D. Olderogge N.G. *Project management.* Omega-L, Moscow, 2004.
[6] Piterkin S.V. *Practice of ERP application.* Alpina Business Books, Mosow, 2005.

EKATERINA AVDEEVA
Vladimir State University
Management and Computer Science in Technical and Economic Systems
Gorky St., 87, Vladimir
RUSSIA
avdkaterina@rambler.ru

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Face recognition using the Eigenface Algorithm

Oana Balan, Razvan Stanescu
Bogdan Ciorobea
Teacher Coordinator: Gabriela Moise

Abstract

The biometric concept of facial recognition makes human recognition a more automated, computerized process. Our purpose is to present an approach to the detection and identification of human faces and describe a working, real-time face recognition system which tracks a person's head and identifies the person by comparing characteristics of the face to those of known individuals grouped in a large database. Face recognition projects the images into a two-dimensional space, which is known as "faces space" or "feature space". The faces space is described by the "eigenfaces", which represents the eigenvectors (principal components) of a set of faces. Each face is characterized by a sum of eigenface features, and the recognition is based on the comparison of these weights to those of known individuals.

1 Introduction

The face plays an essential role in social relationships, transmitting identities and emotions. The human capacity to recognize faces is remarkable: we can distinguish between thousands of figures learnt throughout our life and we can identify familiar faces from one shot even after several years. Computational models for face recognition can contribute not only to theoretical insights, but also to practical applications for a wide variety of problems, such as criminal identification, security systems, image and film processing, human-computer interaction. As a result, this subject was in the spotlight for researchers from all over the world in the last two decades and has been recently introduced as a component for computer software mass production. On the other side, despite numerous theoretical solutions, actual techniques are still far from offering high level of success and cannot be used in sensitive domains such as check-points in airports or facial recognition based access in buildings and institutions. Developing a computational model of face recognition is not very simple, because faces are complex, multidimensional, and contain a wide range of visual stimuli.

Our purpose is to develop a computational model of face recognition that is fast, simple and accurate. The scheme is based on the informational theory approach that decomposes face images into a small set of characteristics called "eigenfaces", which represents the principal component of the initial training set of face images. Recognition is performed by classifying and comparing its position with the position of known individuals. This approach has advantages over other face recognition schemes in speed, simplicity, insensitivity to changes in the face image.

The paper is organized as follows:

The second section has two subsections: subsection Approaches of the face recognition systems makes an introduction into the main algorithms used for face recognition, while the second subsection, Principal Component Analysis, describes the Eigenfaces technique for face detection. The third chapter, The EigenSoft Application, points out the results of our implementation, while the last chapter, Conclusions, brings out the advantages of the application and why our system is better and more efficient than others.

2 Theoretical approach

2.1 Approaches of the face recognition systems

Facial recognition systems are computer programs that, based on a face recognition and detection algorithm, are able to locate, identify and compare human faces.

The concept of automated face recognition is relatively new in the computer field industry. It dates back to 1960 and has developed with many significant advances in the last years . The first semi-automated system of face recognition (early 1960s) required preliminary transformations on the photographs- the location of features such as eyes, ears, mouth or nose-followed by calculations of distances and ratios to a reference point to compare the data.

In the 1970s, Goldstein, Harmon, and Lesk introduced subjective markers such as the colour of the subject's hair and lip thickness to improve the recognition. The problem with both of these early techniques was that the measurements and locations were made manually and required a large amount of time to compute. The solution came in 1988, when Kirby and Sirovich [6] applied a new method, Principal Component Analysis, a technique based on linear algebra mathematics. This method was considered a milestone in the field of face recognition algorithms and the best solution till then , which gave significant results with less than one hundred values used for test.

Over the last ten years, face and object recognition software have become increasingly popular. Its popularity is supported by statistical and probabilistic methods to identify features of the human face (Pontil and Verry in 1998 and Jones and Viola in 2003 [7]). The most noticeable progress has been made by face and expression identification algorithms which have achieved their highest level of success because of the use of feature sets (Lowe in 2003 and Ahonen in 2004 [8]). The researchers are still trying to find a method that derives new features and performs accurate recognitions similar to human eye identification.

Face recognition systems use two main approaches for detection and identification: geometric- based on facial features such as the alignment of the nose, ears, mouth, and photometric- based on the matrix of pixels which composes the image. There are three algorithms studied and applied in face recognition literature: Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA), and Elastic Bunch Graph Matching (EBGM).

2.2 Principal Components Analysis (PCA)

PCA, or Principal Component Analysis, based on the use of eigenvalues and eigenvectors, is the technique developed by Kirby and Sirovich in 1988 [6]. PCA requires the images to be the same size and must first be normalized to line up the eyes and mouth of the subjects. The PCA approach reduces then the dimension of the data by methods of data compression and reveals the lowest and the most efficient dimensional structure of facial patterns. This reduction in dimensions removes information that is not useful and decomposes the face structure into orthogonal (uncorrelated) components known as eigenfaces. Each face image may be represented as a weighted sum (feature vector) of the eigenfaces, which are stored in a 1D array. A test image is compared with a gallery of images by measuring the distance between their feature vectors. The main advantage of this technique is that it reduces the data needed to identify the subject to $1/1000^{th}$ of the data presented.

This approach to face recognition involves the following initialization operations:

1) Acquire an initial set of face images (a training set).

2) Calculate the eigenfaces from the training set, keeping only the first M images that correspond to the highest eigenvalues. These M images define the face space. If new faces are added, the eigenfaces can be updated or recalculated.

3) Calculate the corresponding distribution in M-dimensional weight space for each known individual, by projecting their face images onto the "face space".

After the initialization the following steps are required to recognize the new face:

1) The system calculates the weight of each image based on the input image and the M eigenfaces

2) Determines if there is a face in the image, by comparing the resulted image to the face space

3) If it is a face, calculates the Euclidean distance from the original image to the nearest eigenface

3 The EigenSoft Application

Our project is called EigenSoft. EigenSoft is an application used for face recognition systems and was written entirely with Matlab Software version 2010.

Be given a database which contains images of several persons. We have to determine if a person from another images is retrived in the images from the database. We will try to present a PCA approach to this issue.

There is a well-known face database which can be downloaded from the AT&T Laboratories,Cambridge at http://www.uk.research.att.com/facedatabase.html. It contains ten different images of each of 40 distinct subjects.(Fig. 1) For some subjects, the images were taken at different times,varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details(glasses/no glasses). All the images were taken against a dark homogeneous background with th esubjects in an upright, frontal position (with tolerance for some side movement).

An experiment with a subset of the database, which only contains 4 images per each subject, has been performed to ensure how well the eigenface system can identify each individual's face. There are also 4 additional photos of a known person, added to make sure that the program works correctly.
The images from the Olivetti database contain variations of orientation, rotation (up to 20°), scale of representation (up to 20%), facial expression, lightning conditions. The dimensions of each image are 119x92 pixels, using 256 levels of gray. The format of each image is PGM (Portable GreyMap format).

Fig 1. Images from the Olivetti database [11]

3.1 The Eigenface Recognition Procedure

The Eigenface approach to face recognition is based on the Karhunen-Loeve algorithm, that is presented in the following parragraph and in Fig 2.
Suppose we have a set of observation of M variables. In our case, the M variables are the 41 subjects . For these M variables we also have N test photos, $X_1,\ldots X_N$, where X_i represents an observation of each of the M variables. In our case, M=4, because we use 4 test photos for each subject. We organize the information in a matrix where the columns are the observation vectors X_n. Each observation vector has M elements. The shape of the matrix is therefore MxN. So, we have a matrix with 41 rows and 4 columns. [9]
The Eigenface approach to face recognition involves the following steps:

1. Loading the database
2. Reading the test image
3. Calculating the empirical mean (average) for each variable. We store the averages in a M x 1 matrix
4. Calculating the covariance matrix
5. Choosing the first 100 highest eigenvectors and eigenvalues
6. Calculating the weight (cumulative energy) for each eigenvalue

7. Determinig the eigenvector which resembles the most to the given eigenvector by calculating the Euclidean distance between the given eigenface and the eigenface from the database which minimizes this distance

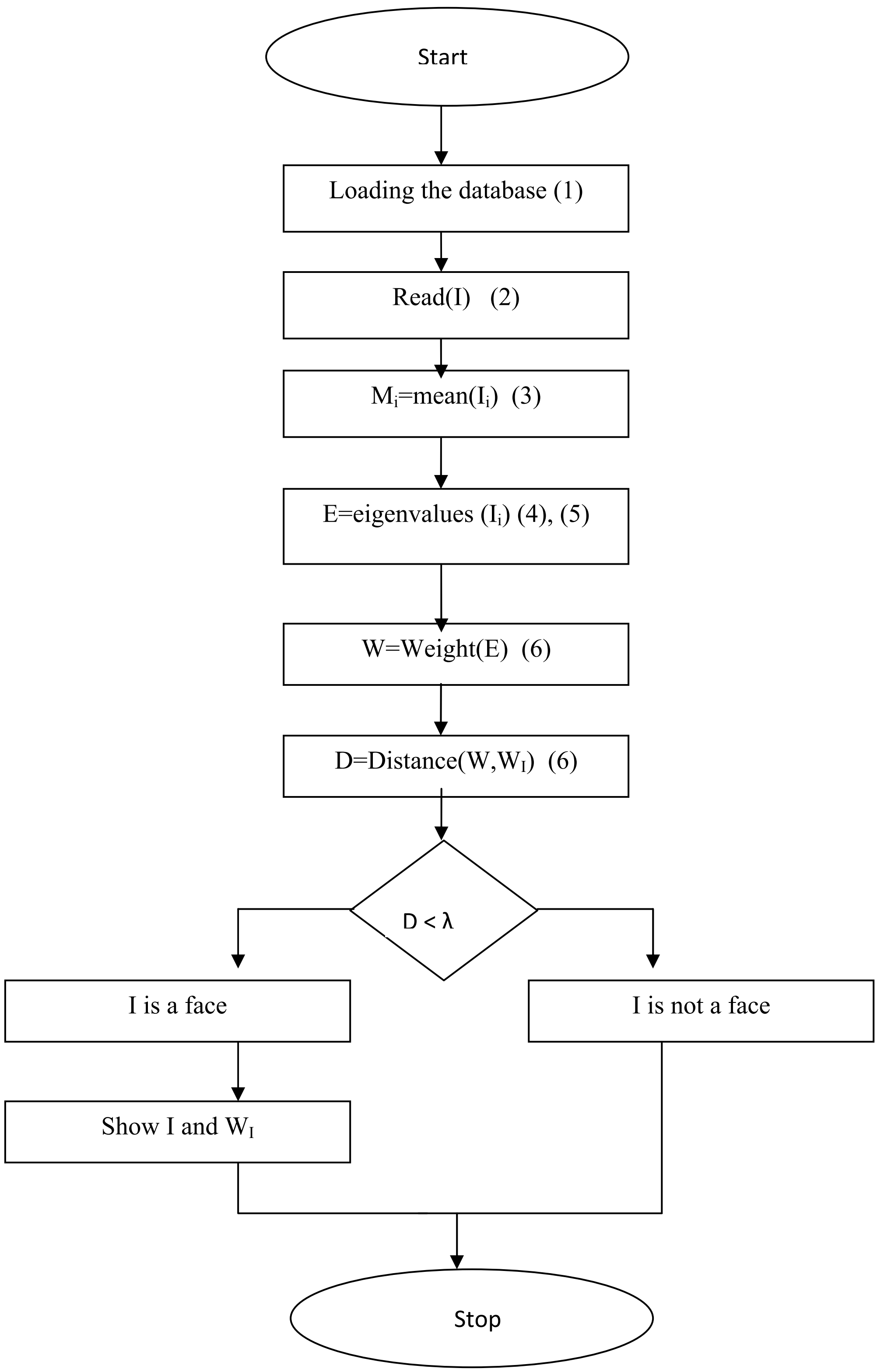


Fig 2. The Eigenface Algoritm

The graphical interface of the program is presented in Fig 3:

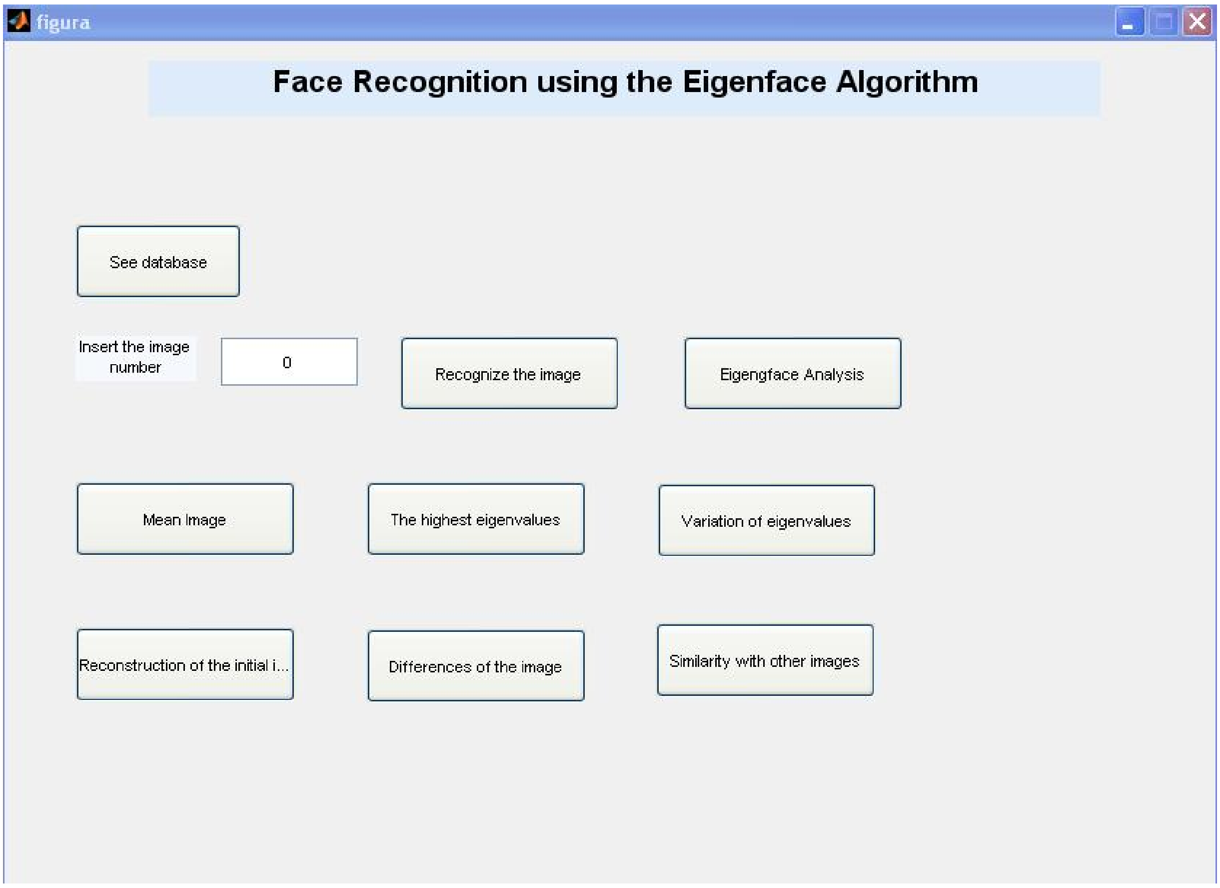


Figure 3. Graphical interface

Results obtained after running the algorithm for image number 81 (81 is the order of the image in the vector of images stored as database for our face photos) are presented in Fig 4.

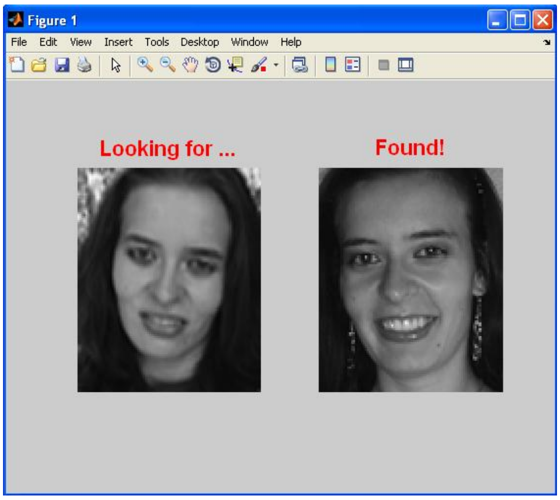


Fig. 4 Results for image number 81

The first image was taken in the year 2006, and the second in 2009. Despite the fact that the facial features have changed over time, the algoritm recognizes that in both pictures appears the same person.

The first function, Mean Image, determines the mean image, an average estimation of the characteristics of all the images in the database, as shown in Fig 5. The image is normalized (x = double(x)/255) ; *argvx* represents the mean image of all images- to obtain the mean image, was used the Matlab function *mean*.

```
nImages = k;
imsize = size(image_data);
nPixels = imsize(1)*imsize(2);   x = double(x)/255;   avrgx = mean(x')';
figure;
 imshow(reshape(avrgx, imsize)); title('Mean image');
```

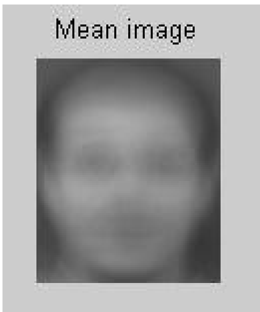


Fig 5- The mean image

The second function, The highest eigenfaces, determines the highest eigenfaces from the covariance matrix, as shown in Fig. 6. The eigenvector is ordered descending, and there were chosen the first two eigenfaces.

```
cov_mat = x'*x;
[V,D] = eig(cov_mat);
V = x*V*(abs(D))^-0.5;
subplot(1,2,1); imshow(ScaleImage(reshape(V(:,nImages  ),imsize))); title('First  eigenface');
subplot(1,2,2); imshow(ScaleImage(reshape(V(:,nImages-1),imsize)));
title('Second eigenface');
```

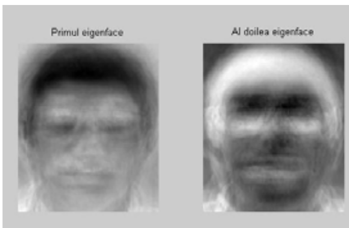


Fig 6- The highest eigenfaces

The third function, The variation of the eigenvalues, determines the variation of the eigenvalues, as shown in Fig. 7. As we can see from the plot, the eigenvectors are ordered ascending, from the lowest to the highest.

```
cov_mat = x'*x;
[V,D] = eig(cov_mat);
V = x*V*(abs(D))^-0.5;
 plot(diag(D)); title('Eigenvalues');
```

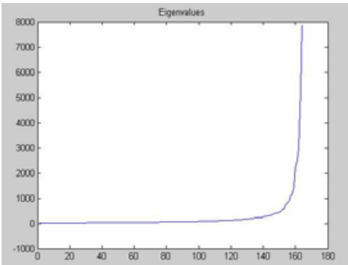


Fig 7- Variation of the eigenvalues

Another function of the application performs the reconstruction of the original image, by adding the mean of all the images to the original image, as shown in Fig 8.

```
KLCoef =  x'*V;
KLCoef(:,1:1:1)= 0;
image_index = str2num(get(handles.edit1,'String'));
reconst = V*KLCoef';
figure;
subplot(1,2,1); imshow((reshape(avrgx+reconst(:,image_index), imsize))); title('Reconstructed image’);
subplot(1,2,2); imshow((reshape(avrgx+x(:,image_index), imsize)));title('Original image’)
```

Fig.8 Reconstruction of the original image

Differences between the original images and the rest of the images are shown in Fig 9
There are calculated the Euclidean distances from the original image to each of the eigenfaces. The formula used is sqrt(sqr(X_1-X_2)+(Y_1-Y_2)), Euclid’s famous formula which calculates the distance between two points in space.

```
for i=1:1:nImages
   dist(i) = sqrt(dot(KLCoef(1,:)-KLCoef(i,:), KLCoef(1,:)-KLCoef(i,:))); end;
subplot(1,2,2); plot(dist,'.-'); title('Euclidean distances ');
```

There can also be calculated similarities between the test image and the rest of the images in the database. The differences result from the Euclidean distances and they can be seen in Fig 10.

```
for i=1:1:nImages
   dist_comp(i) = sqrt(dot(KLCoef(image_index,:)-KLCoef(i,:), KLCoef(image_index,:)-KLCoef(i,:)));
  strDist(i) = cellstr(sprintf('%2.2f\n',dist_comp(i)));
end;
[sorted, sorted_index] = sort(dist_comp);
figure;
for i=1:1:9
 subplot(3,3,i); imshow((reshape(avrgx+x(:,sorted_index(i)), imsize))); title(strDist(sorted_index(i)));
end;
```

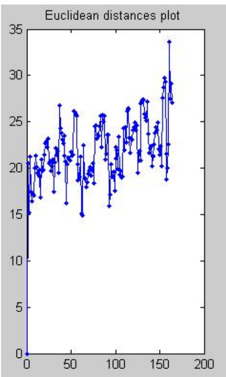


Fig. 9. Differences between

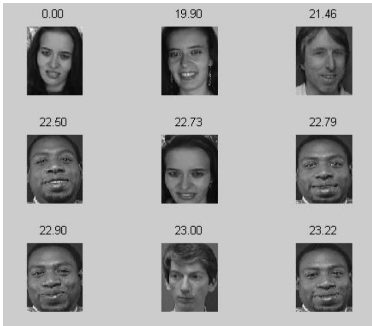


Fig. 10 Similarities between images

4 Conclusions

Biometric facial recognition has the potential to provide significant benefits to society. At the same time, the rapid growth and improvement in the technology could threaten individual privacy rights. The concern with balancing the privacy of the citizen against the government interest occurs with almost all law enforcement techniques. Current use of facial recognition by law enforcement does not appear to run afoul of existing constitutional or legal protections.

Facial recognition is by no means a perfect technology and much technical work has to be done before it becomes a truly viable tool to counter terrorism and crime. But the technology is getting better and there is no denying its tremendous potential. In the meantime, we, as a society, have time to decide how we want to use this new technology. By implementing reasonable safeguards, we can harness the power of the technology to maximize its public safety benefits while minimizing the intrusion on individual privacy.

The eigenface approach to face recognition was motivated by information theory, leading to the idea of basing face recognition on a small set of image features that best approximates the set of known face images , without requiring that they correspond to notions of facial parts and features. The eigenface approach provides a practical solution that is well fitted to the problem of face recognition. It is fast, relatively simple , and has been shown to work well in constrained environment.

Face recognition technology can be used in many areas and in many different ways. The most frequent applications are security-related, but there are also other applications concerning personal use or productivity enhancement. [12]
For instance, face recognition systems can be used to enhance security in threatened areas where it is important to identify persons based on facial characteristics- in public security, live video cameras, video footage, border control applications to identify biometric traits imprinted on ePassports or other travel documents, photo-ID documents such as drivers' licences to prevent fraud or other attacks, criminal investigation to identify suspects by comparing a given image to suspects' images stored in a database and return a match list, mobile identity check or physical access control to buildings, stadiums, office space or other locations that are intended to be made more secure by authenticating authorized persons.

EigenSoft is better than other face recognition systems because it uses the Eigenfaces algorithm, which has the following advantages:

1. The execution time of the systems: Using EigenSoft is very fast, and able to functionally operate on lots of faces in very little time.
2. Accuracy: For the system to work well, the faces need to be seen from a frontal view under similar lighting. Face recognition using EigenSoft has been shown to be quite accurate. By experimenting with the system to test it under variations of certain conditions, the following correct recognitions were found: an average of 96% with light variation, 85% with orientation variation, and 64% with size variation. However, as there has a high correlation between the training data and the recognition data. The accuracy of EigenSoft depends on many things. As it takes the pixel value as comparison for the projection, the accuracy would decrease with varying light intensity. Besides, scale and orientation of an image will affect the accuracy greatly. Preprocessing of image is required in order to achieve satisfactory result.
3. Simplicity: One of the major advantages of EigenSoft recognition approach is the ease of implementation. Furthermore, no knowledge of geometry or specific feature of the face is required; and only a small amount of work is needed regarding preprocessing for any type of face images.
4. Efficiency: EigenSoft is very efficient in processing time and storage. PCA reduces the dimension size of an image greatly in a short period of time.

5. Data compression: Data compression is achieved by the low-dimensional subspace representation. Recognition is simple and efficient compared to other matching approaches.

In the near future, we will try to integrate face recognition technology in an application for tracking faces in video streams and for analyzing and checking face features. Our software supports two dimensional intensity data, but our purpose is to extend the use of the Eigenfaces Algorithm to 3D data with additional functionality of substantial and accurate portrait characteristics determination.

References

[1] Matthew Turk, Alex Pentland, *Eigenfaces for Recognition*, Vision and Modelling Group, The Media Laboratory, Massachussets Institute of Technology,1991

[2] John D. Woodward, Jr., Christopher Horn, Julius Gatune, and ArynThoma, *Biometrics A Look at Facial Recognition*, RAND Public Safety and Justice, 2010

[3] Kyungnam Kim, *Face Recognition using Principle Component Analysis*, Department of Computer ScienceUniversity of Maryland, College Park

[4] Kale K. V., Advances *in Computer Vision and Information Technology ,* S.C. Mehrotra and R.R. Manza, I.K. International, 2008

[5] Meyers Ethan, Wolf Lior, *Using Biologically inspired features for face processing*, Springer Science+Business Media, LLC 2007

[6]Sirovich L, Kirby M., *Low-dimensional procedure for the characterization of the human faces*, Division of Applied Mathematics, Brown University, Providence

[7] Jones Michael, Viola Paul, Face *Recognition using booster local features,* International Conference on Computer Vision, 2003

[8] Ahonen T, Hadid A & Pietikäinen M*, Face recognition with local binary patterns*.,Computer Vision, ECCV 2004 Proceedings, Lecture Notes in Computer Science 3021, Springer, 469-481.

[9] http://www.biometricscatalog.org/NSTCSubcommittee/Documents/Face%20Recognition.pdf

[10] http://www.biometrics.tibs.org/

[11] http://web.mit.edu/emeyers/www/face_databases.html#orl

[12] http://www.cognitec-systems.de/Applications-for-security.18.0.html

Oana BALAN
Petroleum-Gas University of Ploiesti
Informatics Department
no. 39 Bd. Bucuresti, 100680, Ploiesti
ROMANIA
E-mail: oanab_2005@yahoo.com

Razvan STANESCU
Petroleum-Gas University of Ploiesti
Informatics Department
no. 39 Bd. Bucuresti, 100680, Ploiesti
ROMANIA
E-mail: razzvan19@yahoo.com

Bogdan CIOROBEA
Petroleum-Gas University of Ploiesti
Informatics Department
no. 39 Bd. Bucuresti, 100680, Ploiesti
ROMANIA
E-mail: bobo_crb@yahoo.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Coding Game
Programming based educational application

Adrian George BOSTAN
Teacher Coordinator: **Lect. dr. eng. Adina COCU**

Abstract

In the present paper, I will introduce an educational application in the form of a computer puzzle game, which allows the user to test his programming aptitudes in a user-friendly graphical environment. The game can be used as a tool for better understanding a particular programming language. The versatility of the application stems from the ability to easily import the structures of any programming language in the game. As a result, the user has the power to choose from a wide range of programming languages. The application comes with levels for three predefined languages: C#, C++ and Java.

The **Coding Game** bundle consists of the game along with two tools for adding, removing and modifying the content of the game. The main application has three game modes, each mode targeting a different aspect of learning a programming language. The three game modes are: *Puzzle*, *Order* and *Test*, and together, they can make the process of learning a programming language more pleasant and more fun.

The goal of this application is not to replace IDE's or programming books, but to serve as an additional tool for the user, allowing him to improve his skills and knowledge through playing a game, thus making the whole experience more enjoyable.

1. Introduction

The main purpose of educational systems is to help a particular person learn faster and more efficiently. It's common knowledge that the process of learning is difficult and it's based on the action of continually accumulating information. This process can be improved by using these educational systems in the phases of accumulating and testing knowledge.
In the last couple of years, there have been developments in educational systems based on computer games. However, the use of this kind of system is still isolated and there is a general misconception that educational games are only suitable for children. The presented application targets people who want to test their knowledge of programming. Consequently, the user can be any person interested in learning how to program.
Computer games have the ability to keep the player focused for long periods of time while entertaining him. They enhance cognitive skills, such as problem solving, decision making, organization, critical thinking, etc. This is the reason why games can be used for educational

purposes, by challenging the player to acquire the presented information, without getting him bored. The visual elements specific to computer games can make learning a pleasant experience while providing a proper level of relaxation and can motivate the player to further improve his skills.

Puzzle games are perfectly suitable for learning because they don't require the undivided attention of the player. This means that the user is not exhausted after playing the game, but actually relaxed. **Coding Game** is a puzzle game at its core, intended for testing and improving abilities related to programming. The application can help in better understanding the general structure of a program, in memorizing the syntax of a programming language and in adopting a particular programming style. By going through the levels of the game, the player observes different techniques used for achieving different things, techniques which he can acquire for future use.
The tools provided with the actual application have the role to extend the game by adding levels for the predefined programming languages (C#, C++, Java), or by adding in new ones. This feature makes **Coding Game** a universal learning tool for programming languages.

2. Application description

2.1 General description

The application and the tools were programmed in C#, using the XNA Game Library developed by Microsoft. The IDE I used for writing the application is Microsoft's Visual C# 2010 Express Edition.

Coding Game consists of the main application (the game) and two graphical tools for loading content in the game: the *Language Tool* and the *Level Tool*. The *Language Tool* is used for importing programming languages in the game, and the *Level Tool* is used for loading levels for a particular programming language and a particular game mode.
The game has three game modes: *Puzzle*, *Order* and *Test*. All the game modes are timed. The time in which a particular level is to be completed is set through the *Level Tool,* when loading new levels in the game.
Each programming language in the game has its own three game modes. So, the languages are separated from one another. Before choosing a game mode from the three mentioned above, you have to choose a programming language from the list of languages detected by the game.
Every sequence of code present in the game, depending on the programming language, has syntax colouring. The colours used to highlight a particular language are set through the *Language Tool*.

2.2 Important classes

- ScreenManager - Controls all the game states.
- GameState - Creates a basic structure for all the game states. All the game states extend this class.
- TextureManager - Static class used for loading all texture assets in the game.
- Game States:
  - Menu - The menu game state. It's the active game state when launching the application.
  - Help - Provides some information about the game modes.
  - SelectLanguage - Allows you to select a programming language to play with.
  - ChooseGame - Select a game mode: *Puzzle*, *Order*, *Test*
  - Puzzle - Loads all *Puzzle* levels and controls them, making them active one at a time.
  - Order - Loads all *Order* levels and controls them, making them active one at a time.

– Test - Loads all *Test* levels and controls them, making them active one at a time.

- Levels:
  – PuzzleLevel - a *Puzzle* level. The Puzzle class contains and controls all the *Puzzle* levels.
  – OrderLevel - an *Order* level. The Order class contains and controls all the *Order* levels.
  – TestLevel - a *Test* level. The Test class contains and controls all the *Test* levels.

2.3 The game modes

Puzzle: the first part of **Coding Game,** in which the player receives the description of a program and a set of puzzle pieces belonging to a source file. The structure of the source file is rendered on the screen, but without any code (only white space is shown). Every line of code in the original source file is divided into puzzle pieces. The player must use the puzzle pieces to recreate the original source with the working program, specified by the description. The puzzle pieces are shuffled each time, so that the main focus is to understand how and why particular pieces of code are used. A *Puzzle* level is done when the source file has been recreated using the puzzle pieces, or when the time set for the level is up.
Additional levels for this game mode can be loaded with the *Level Tool.*

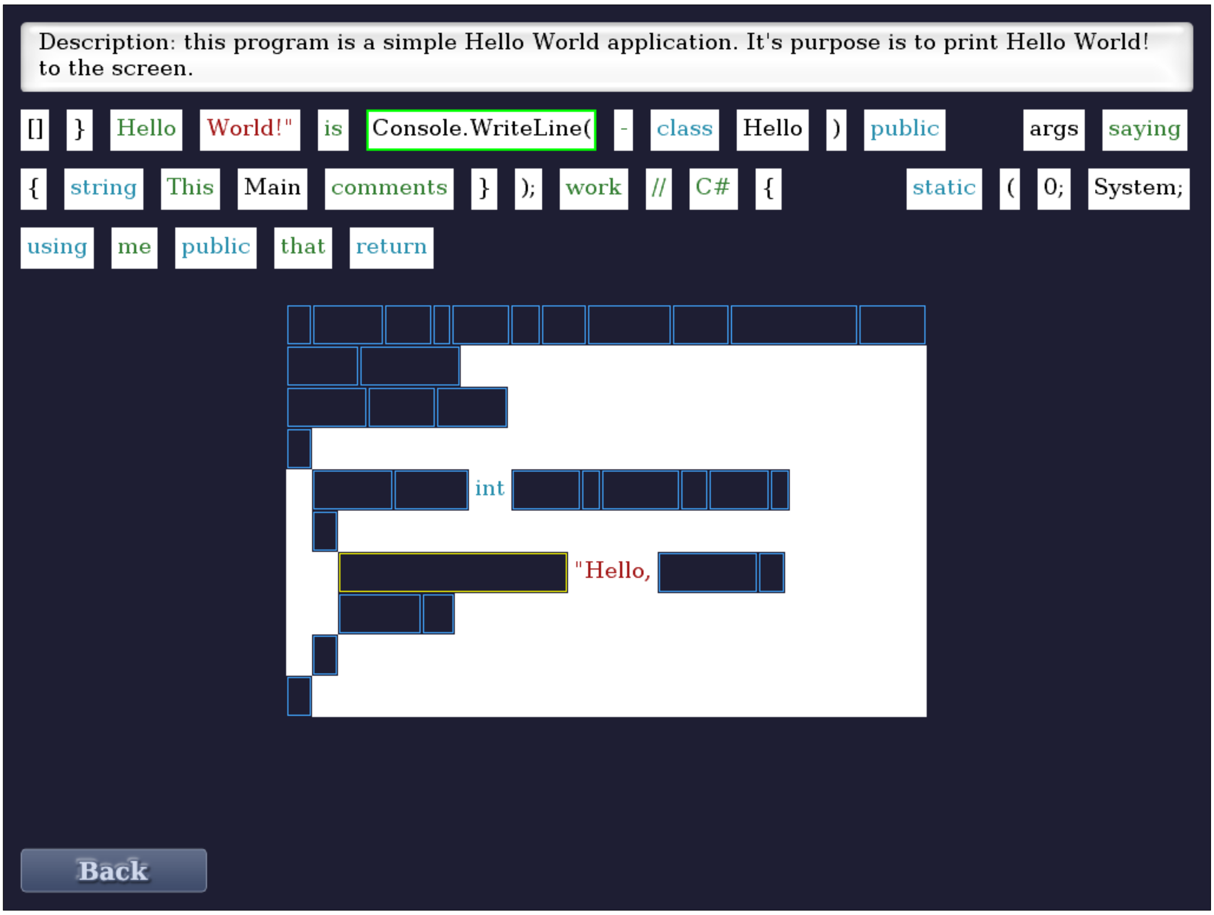


Figure 1 – A Puzzle level

Order: is the second game mode of **Coding Game** and consists of a number of code sequences that form a program. The code sequences are not in the correct order. Along with the sequences, a

description of what the program is supposed to do is provided. The player has to put the chunks of code in the correct order, to create a working program that fits the received description.
The main difference between the *Puzzle* game mode and the *Order* game mode lies in the intended learning concept. The *Puzzle* game mode is more complex, the player has to unscramble a source file by placing the puzzle pieces in the empty table which represents the source file. The *Puzzle* game mode helps with learning the keywords and their use, memorizing the structures used in a program and with adopting a proper programming style. In contrast, the *Order* game mode provides the user with a number of chunks of source code and the emphasis falls on learning the logic behind a program, the right order in which things should be done and observing the used techniques which can serve as future references for the player. An *Order* level is over when the sequences of code are in the correct order, or when the time is up.
Additional *Order* levels can be loaded with the *Level Tool*.

Figure 2 – An Order level

Test: is the last game mode of the application and it is based on a quiz. The player gets a number of questions about programming <egz: What is the result of the following sequence of code?> which he has to answer. Every question has four answers. Only one of the four answers is correct. This game mode tests the theoretical aspects of learning a programming language and challenges the player to see if he has a solid grasp of the programming concepts. A *Test* level is over when all the questions have been answered, or when the time allocated for the test has passed.
Additional *Test* levels can be loaded with the *Level Tool*.

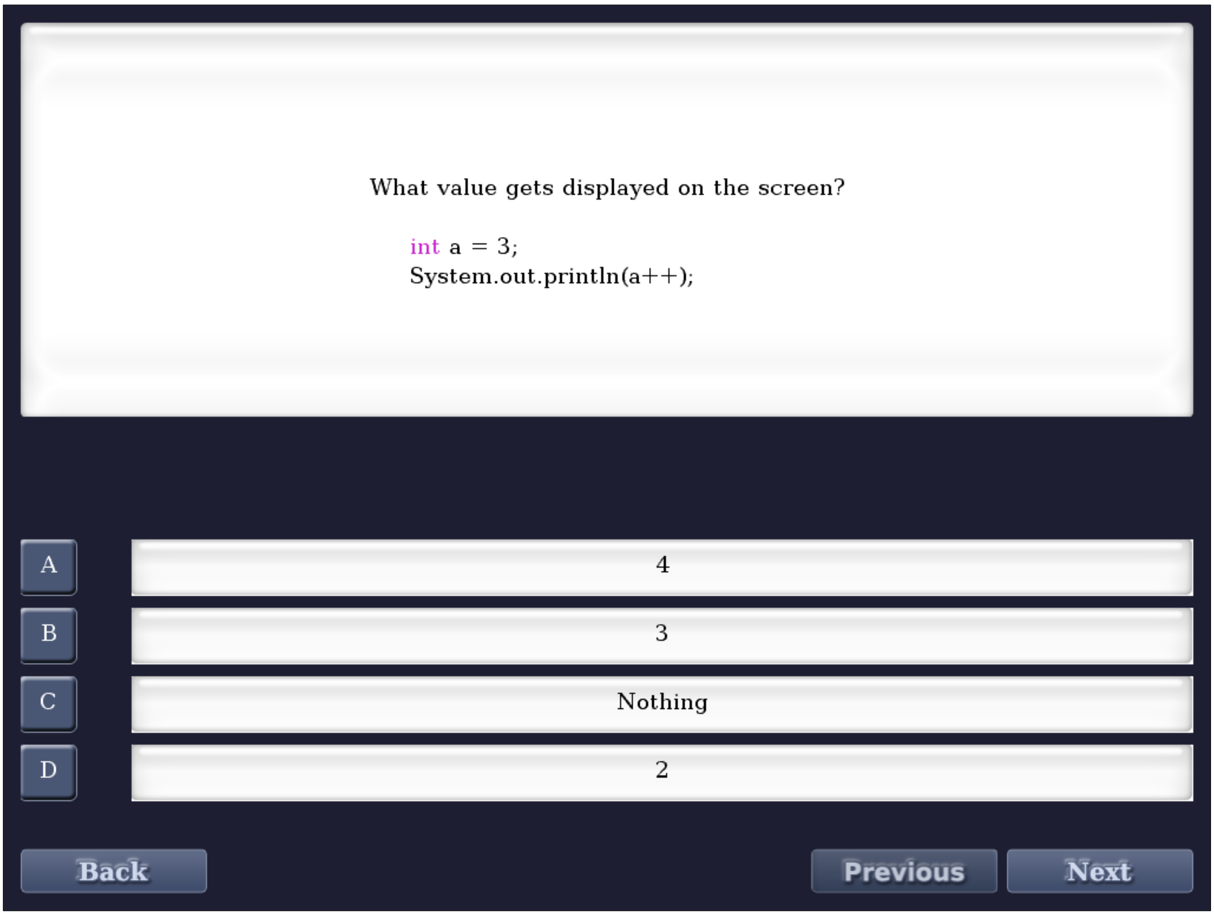


Figure 5 – The level tool

2.4 The game modes

The tools are used for adding personal content to the application. There are two tools available: the *Language Tool*, and the *Level Tool*. The tools feature an intuitive interface and are very easy to use. The tools make the application versatile, because through them, the user can load his own languages and levels for those languages.

The **Language Tool**: is used for importing languages in the game. In order to import a language, all the keywords of that language have to be provided. Also, the colours used to highlight the syntax of the loaded language have to be chosen. This tool can also remove languages from the game.

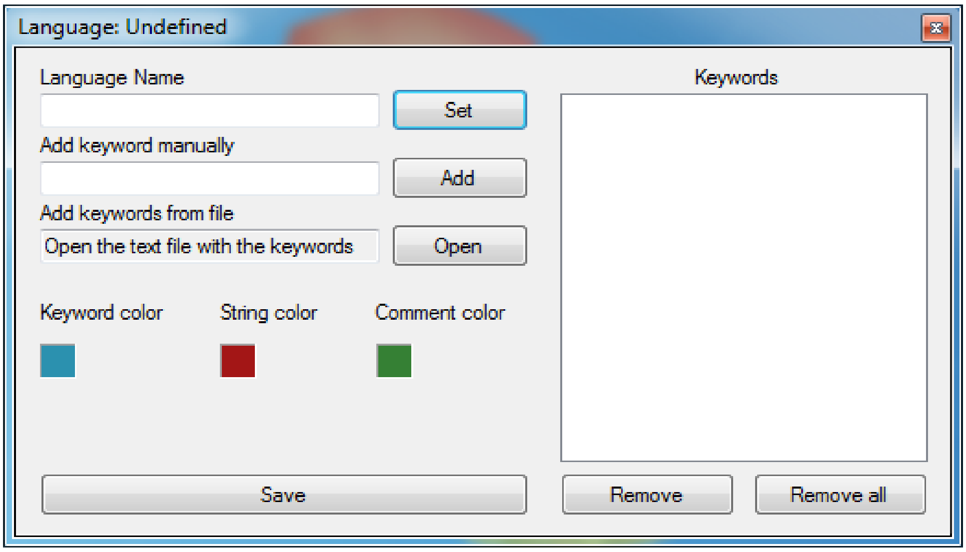


Figure 4 – The language tool

The **Level Tool**: is used for loading levels for a particular game mode (*Puzzle*, *Order*, *Test*) and for a particular programming language. The languages imported with the *Language Tool* are detected by the *Level Tool*. As each game mode is different, the process of loading a level is different for each of the game modes. Also, this tool can remove levels from the game.

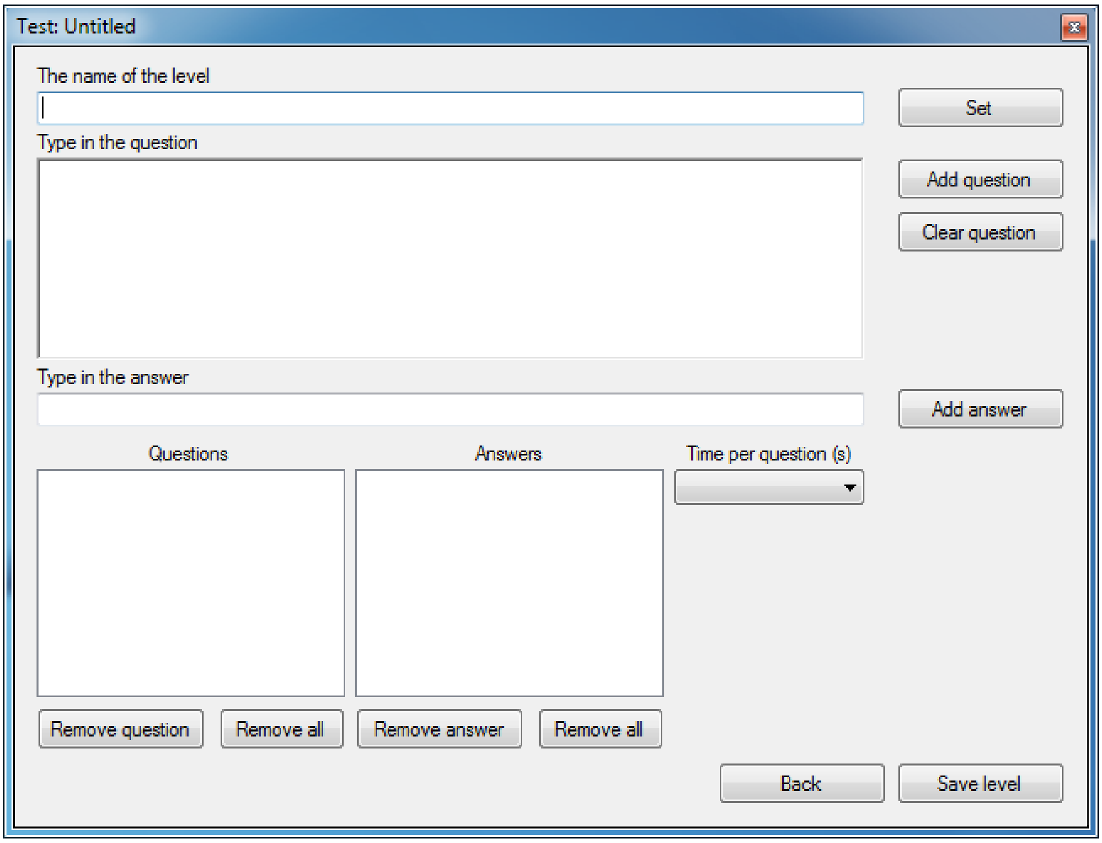


Figure 5 – The level tool

3. Conclusion

At this time, **Coding Game** only works locally. My ambition is to give the application the ability to take its resources from a server. This way, the application could even be used in schools to test students on their programming skills. The students would have access only to the executable of the application. The content of the game would be located on the school's server so that the students couldn't access it. The teachers would use the tools to create different kinds of tests, depending on the game mode, for the students. The content would be changed for every test, removing the old test, and loading a new one in the game. This kind of testing would be beneficial because it's more enjoyable for the students and would motivate them to learn.

Acknowledgements: I would like to thank dr. Adina COCU for all her patience and support.

References

[1] Professor Angela McFARLANE, *Report on the educational use of games*.
[2] dr. eng. Adina COCU**,** *Contributions to The Knowledge Representation Methods in Intelligent Tutoring Systems,* 2009.

ADRIAN GEORGE BOSTAN
Universitatea "Dunarea de Jos" Galati
Facultatea de Automatica, Calculatoare, Inginerie Electrica si Electronica
Str. Domneasca, nr. 47, Galati
ROMANIA
wait4.asecond@gmail.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Image processing and interaction elements OpenImage 0.9 Alpha

Bota Florentin
Teacher Coordinator: Prof. PhD. Dana Simian, Lect. Ralf Fabian

Abstract

The purpose of this project is to design and develop a robust application in an effort to better understand image processing and computer vision, while keeping a simple and user-friendly interface. Eclipse SDK was used as the programming environment, proving to be a real asset, but all was created using the open-source principle of Java.

OpenImage, the main application, evolved from a simple Java Panel to a uniquely shaped Image Viewer with processing tools, original interface and feedback, motivated by the will to discover and learn the high end capabilities of computers, with applications in everyday life, health-impaired, IT and even robotics.

Keywords: image processing, java, OpenImage, Computer Vision, speech

1 Introduction and description

In electrical engineering and computer science, *image processing* is, according to wikipedia definition, any form of signal processing for which the input is an *image*, such as a photograph or video frame; the output of image processing may be either an image or, a set of characteristics or parameters related to the image.

Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it.

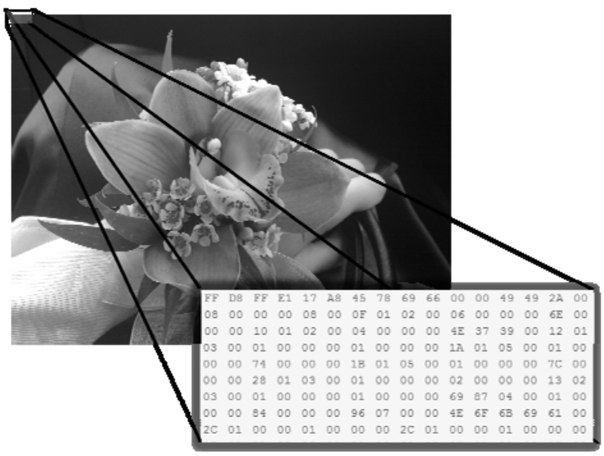


Fig. 1: Computer understanding of a picture

For a simple user, image processing consists in taking a picture, croping it, maybe tweak a little with brightness and contrasts options and so on. But the applications of the tehnology are far larger than this and goes through medical field (EKG, microscopy etc.), security, astronomy, industrial robotics, and even remote sensing by satellites.

Scientific and technological advances have led to a rising exponential curve in the way that man can now create, manipulate and save not only images, but all that means information. The camera was an important step, enabling faithful reproduction of color images and made possible the representation of moving. The introduction of television had a strong social impact as an instrumenent of information and entertainment for communities, along with newspapers, magazines and other printed forms of communication through images.

A new trend is the use of images via the PC, from online media presentations (magazines, trade shops, etc..) up to social networks (Hi5, Facebook, etc.), the visual medium of transmission is preferred due to human anatomy. To be saved, displayed and even transmitted through the Internet, digital images are represented in a way easily usable by computer systems.

In this project I tried to understand the concept of mathematical image processing algorithms underlying the images and create an easy and interactive interface. Also wanted to connect the user with the machine, enabling computer system to use the visual medium to communicate.

Eclipse development environment, with JMF's Framework, OpenCV, JNeuroph and others were used to create OpenImage application, for viewing, processing and saving images. It is worth mentioning that this application has addressed an innovative way of interaction.

2 Materials and Methods

2.1 Project motivation and objectives

The reason image processing was chosen was the fact that it provides a wide spectrum for software development and we mention here media, social networking, art, security, video games, medical analysis etc.. It is also a good opportunity to analyze various platforms, but also to better understand how computers process and use images, static or not.

Regarding the practical application, two perspectives were adressed:

1) Creating a similar „Windows Photo Viewer”, but based on the Java language, in order to benefit the cross-platform and open source oportunity.
2) Making a parallel to the frameworks used for real-time image capture and recognition of objects based on artificial intelligence or not, and also trying to implement a system in the above program. Practical problem concerns the recognition of the user's hand and using it to interact with the application.

Secondary objectives in the above example:

a. A better understanding of Java technology development and the Eclipse environment
b. Creating a GUI based on Swing and a better understanding of how BufferedImage class is used.
c. Implement a modern and user-friendly interface.
d. The design of special buttons for orders next / previous.
e. Display of images by keeping the correct proportions regardless of JFrame's size and implementation of a zoom option.
f. Creating a slide-show system to allow viewing panoramic images, with the ability to select the speed of succession.
g. Implementation of filters applicable to various images to better understand the way they operate (blurr, contrast, negative, sharpen, channel selection, histogram, threshold, etc.).
h. Testing various CV type platforms (Computer Vision), Media Framework, and neural networks to establish a method of recognition of the human hand, with clear reference to OpenCV, JMF, and others like Neuroph.

i. Providing this application under the GNU licence for those who want to use the software and those seeking a better understanding on image processing in Java, by uploading it on the SourceForge server, server used to obtain many of the tools used in this project.

2.2 Image processing in Java 2D

Java.awt.image package provides a pair of interfaces that define operations on BufferedImage and Raster objects: BufferedImageOp and RasterOp.

The most important classes that implement these interfaces are: AffineTransformOp, BandCombineOp, ConvolveOp, LookupOp, RescaleOp. These classes can be used for geometrical transformation, blur, sharpen, contrast enhancement, image binarization and color change.

Example 1: *Edge detection using Java 2D*

```
//
float[] elements = { 0.0f, -1.0f, 0.0f,
-1.0f, 4.f, -1.0f,
0.0f, -1.0f, 0.0f, };
//read the bi buffered image
BufferedImage bimg = new
BufferedImage(bw,bh,BufferedImage.TYPE_INT_RGB);
Kernel kernel = new Kernel(3, 3, elements);
//a kernel is used in order to specify how a pixel is affected
//by the surounding neighbours
ConvolveOp cop = new ConvolveOp(kernel,
ConvolveOp.EDGE_NO_OP, null);
cop.filter(bi,bimg);
```

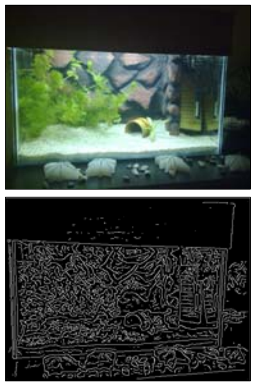


Fig. 2: Edge Detection în Java

Fig. 3: *Simple Blur*

The following code fragment (Example 2) illustrates how to use one of the classes of image processing ConvolveOp. In this example, each pixel of the image source is mediated equally by the eight surrounding pixels.

SimpleBlur variable in the previous example contains a new instance of ConvolveOp implementing a blur operation on a BufferedImage or Raster. Assuming sourceImage and destImage are two instances of BufferedImage, when called filter method, which is the main method of class ConvolveOp, this method assigns values for each pixel in the destination image by averaging the corresponding pixel in the source image with the eight surrounding pixels.

By modifying the convolution kernel you can perform other types of convolutions, such as blurring (Gaussian blur, radial blur, motion blur), sharpening, smoothing operations, etc..

Example 2: *Using ConvolveOp for Blur*

```
//
float weight = 1.0f/9.0f;
float[] elements = new float[9]; // create a 2D surface
// fill it with equal elements
for (i = 0; i < 9; i++) {
elements[i] = weight;
}
// create a Kernel
private Kernel myKernel = new Kernel(3, 3, elements);
public ConvolveOp simpleBlur = new ConvolveOp(myKernel);
```

```
// sourceImage and destImage are BufferedImage instances
simpleBlur.filter(sourceImage, destImage) // blur
```

The problem is to determine a computer system to act similar to a human and make normal mistakes like an optical illusion. Sometimes is harder to make a system susceptible to error, especially if you are trying to mimic a human being.
How is it possible for a static image to appear animated?

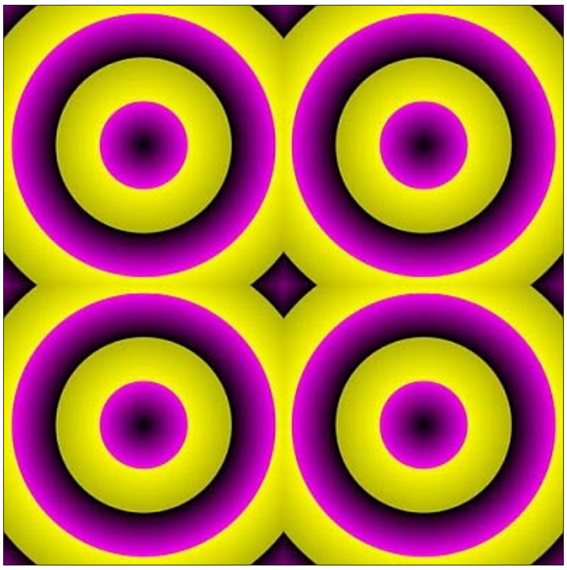


Fig. 4: *Optical Illusion*
Source:
http://unrealitymag.com/index.php/2010/01/25/ten-amazing-optical-illusions/

2.3 OpenCV

OpenCV is an open source computer vision library originally developed by Intel. It is free for commercial and research use under a BSD license. The library is cross-platform, and runs on Mac OS X, Windows and Linux. It focuses mainly towards real-time image processing and example applications of the OpenCV library are: Human-Computer Interaction (HCI); Object Identification, Segmentation and Recognition; Face Recognition; Gesture Recognition; Motion Tracking, Ego Motion, Motion Understanding; Structure From Motion (SFM); Stereo and Multi-Camera Calibration and Depth Computation; Mobile Robotics.

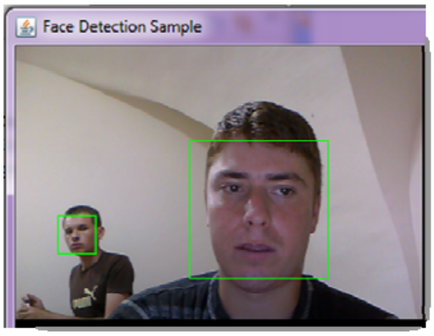


Fig. 5: *Face detection using OpenCV*

OpenCV is designed to provide the basic tools needed to solve image processing problems. Even when not appropriate, the basic components of the library are quite complex to allow development of a complete solution to a problem. It comes with a default OpenCV face detection method based on HaarCascadeTraining as seen in Figure 5.

3. OpenImage

OpenImage is an attempt to better understand image processing at both basic and complex mode on pattern recognition. As the name suggests, the application deals with the imagery, but not only, using Java programming language and Eclipse IDE on graphics API. The program is an image viewer and editor built using the AWT (Abstract Windowing Toolkit) and Swing, with the OpenCV framework.

The program has a graphical user interface (GUI), which is a term widely understood to refer to all types of visual communication between a program and its users. This graphical interface includes graphical objects (buttons, menus, checkboxes), behind which *Event Listeners* make interactions with the application's users.

The user interface will open a default picture of two white diagonal lines. The user can then open an image of his choice, and can apply the following filters: blur, brighten, edge detect, negative, rotate, then it can be saved on the hard disk. The image files can be saved with different extensions: jpeg, jpg, bmp, png, gif.

Also, a Process button is available, where the user can try some of the implemented filters, image processing tools like Threshold or Blur, or even artistic effects like Mirror or Chrome.

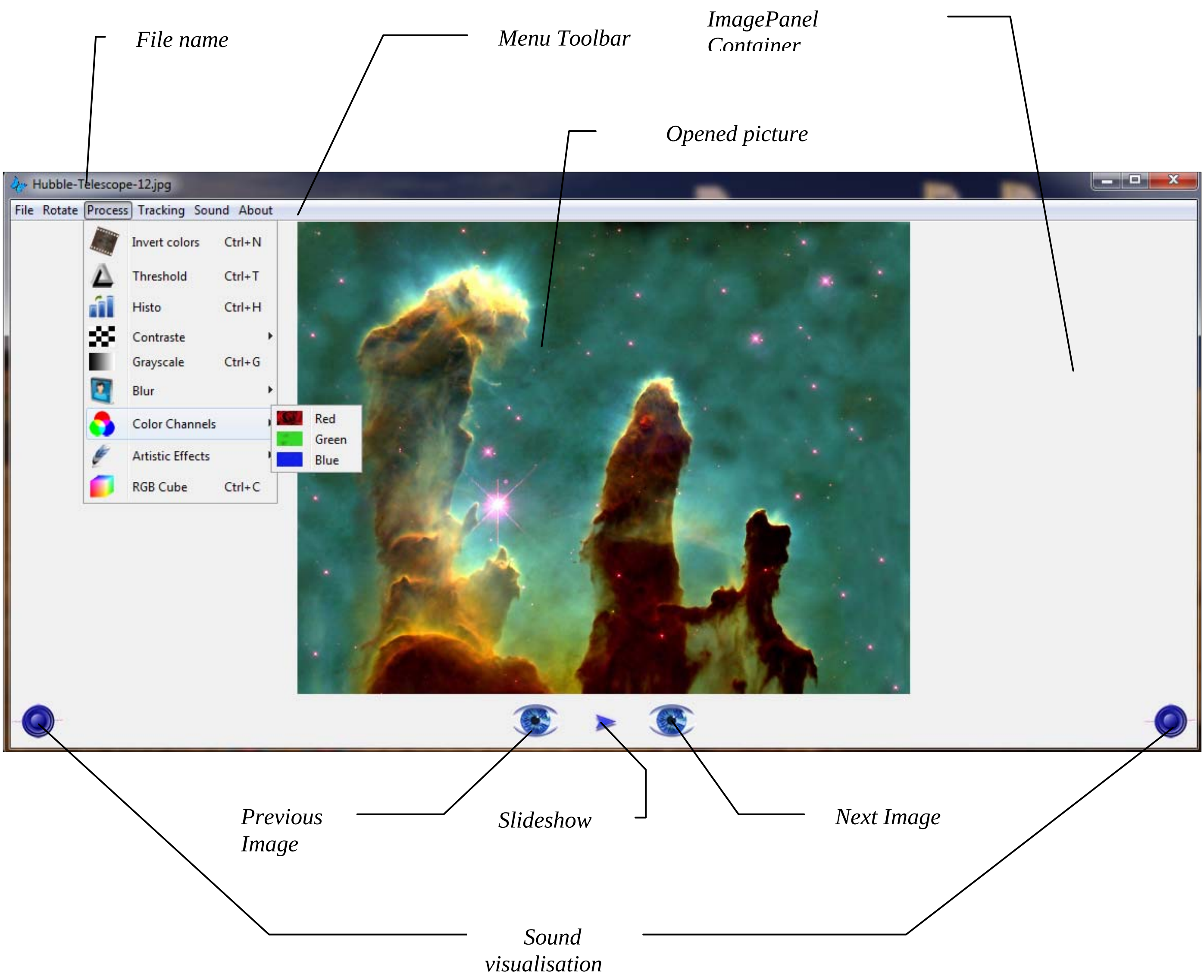


Fig. 6*: Basic OpenImage preview*
Picture source: Hubble Space Telescope-NASA

3.1 Tracking option

The menu is intended to implement a tracking control method application through visual interaction with the computing system. Basically, a way to allow the user to change the current image with a simple movement of the hand, without using other peripherals. To capture images, a webcam is recommended. The following paragraphs will follow the stages taken in this direction.

I originally started the idea of contour detection and their recognition using neural interfaces. (Neuroph core framework). Initial problems were the capture images in real time using a webcam in Java, but were solved by using JMF (Java Media Framework).

Once captured images, we applied in this order: contrast filters, blur, and grayscale transformation using Canny edge detection, segmentation and then to eliminate background.
The results were remarkable, as seen in the figures below.

Fig. 7: *Edge Detection and HSV hand tracking*

The problem was learning the neural network to recognize the shape of a hand. I tried this with Neuroph, but the results were scarce, because the networks could only work with small images (24x24) and needed a database large enough to be able to train.

The following approach with notable results is image segmentation according to skin color pixels. However, although more reliable, this method depends on light conditions and background to provide accurate results.

The figure below is observed the use of JMF for segmentation pixel (in RGB) and false results of the procedure.

Example 3: *RGB skin color segmentation*

```
//
public boolean isSkin(int color){
boolean isSkinTone = false;
int red = (color & 0x00ff0000) >> 16;
int green = (color & 0x0000ff00) >> 8;
int blue = color & 0x000000ff;
int max, min;
//
if ((red > green) && (red > blue)) {
  max = red;
} else if ((green > red) && (green > blue)) {
        max = green;
        } else {
            max = blue;
        }
```

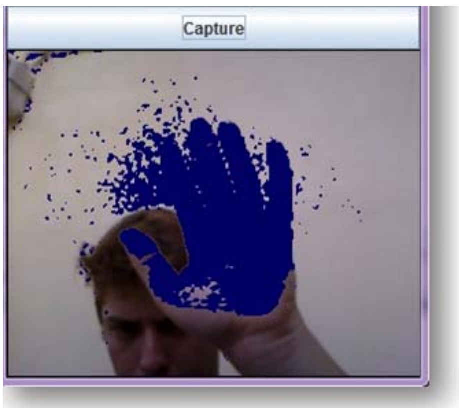


Fig. 8: *RGB skin color select*

```
if ((red < green) && (red < blue)) {
      min = red;
} else if ((green < red) && (green < blue)) {
      min = green;
} else {
      min = blue;
if ((red > 95) && (green > 40) && (blue > 20) && ((max-min) > 15) && (Math.abs((red-green)) > 15) &&
(red > green) && (red > blue))
            {
                  isSkinTone = true;
            }
            return isSkinTone;
 }
```

Since in this case the results were not satisfactory, we turned our attention to OpenCV. Using methods provided by this library we have reached remarkable results. Of note is that applications created using OpenCV are mainly written in C and C++ using Eclipse CDT environment.

As you can see, results weren't perfect, so another Haarcascade was needed. Creating one takes a few days and a large pozitive and negative image database, but with the right instruments and algorithm, interaction without a device becomes possible.

The figures above clearly proving OpenCV library capabilities, we tried another implementation of hand detection using HSV format; the results are acceptable under natural light only.

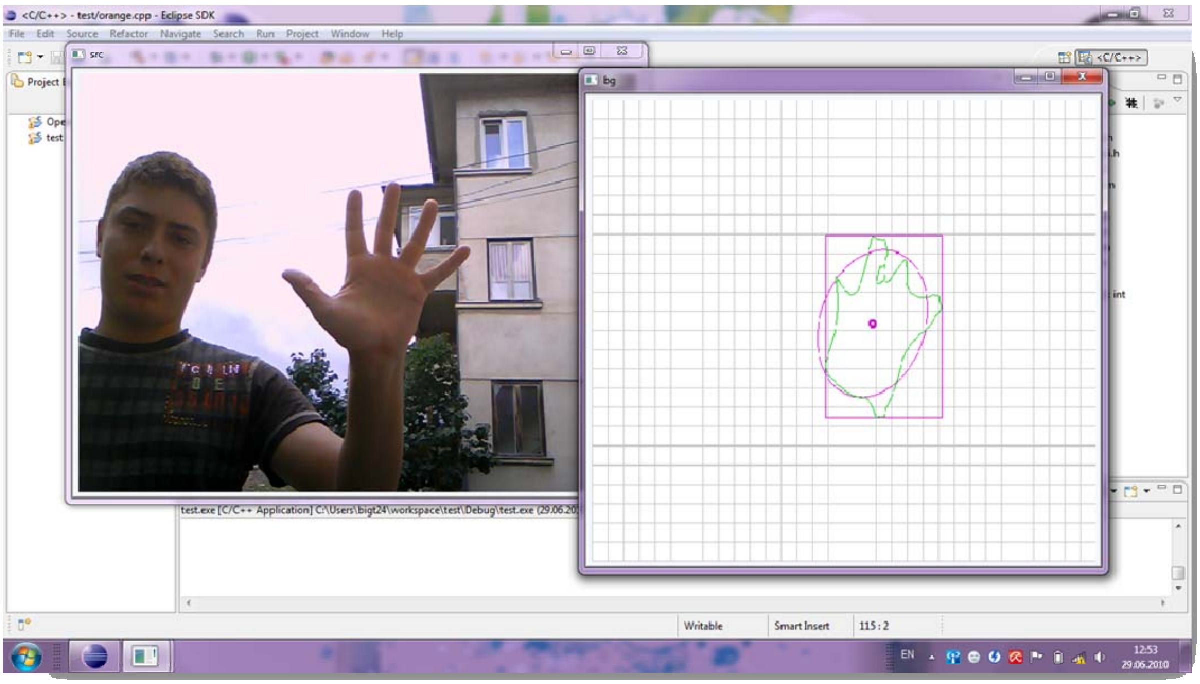


Fig. 9: *Using HSV skin color segmentation in an algorithm based on OpenCV library*
Algorithm Source: Andol X. Li [7]

Example 4: *Openimage hand detection extract*

```
while (t != null && cv != null) {
            try {
                  // JOptionPane.showMessageDialog(getContentPane(),
                  // "Software or hardware problem detected!", "Error", 0);
```

```
t.sleep(FRAME_RATE);
cv.read();
if (ml > 9998)
    ml = 0;
ml++;
MemoryImageSource mis = new MemoryImageSource(cv.width,
        cv.height, cv.pixels(), 0, cv.width);
frame = createImage(mis);
squares = cv.detect(1.2f, 2, OpenCV.HAAR_FIND_BIGGEST_OBJECT,
        20, 20);
int n = squares.length;
if (n > 0) {
    int i = 0, k = 0, max = squares[0].width
            + squares[0].height;
    for (Rectangle rect : squares) {
        if ((rect.width + rect.height) > max) {
            max = rect.width + rect.height;
            k = i;
            // parent.print(1);
        }
        i++;}
    memory[ml] = (squares[k].x + squares[k].width) / 2;
    int cresc = 0;
    int s;
    if (ml > 10)
        if ((memory[ml - 10] - memory[ml] > 80)
                || (memory[ml - 10] - memory[ml] < -80)) {
            cresc = 0;
            for (int p = ml - 10; p < ml - 1; p++) {
                s = memory[ml] - memory[ml + 1];
                if (s > 7) {
                cresc++;
                } else if (s < -7) {
                    cresc--;
                }
            }
            if (cresc <= -5&&(memory[ml - 10]!=0 && memory[ml]!=0)) {
                if(parent.soundSwitch==true)
                {
                    Thread a1 = new VoiceOI("Next", "text", parent);
                a1.start();
                }
                                        Thread.sleep(100);
                parent.previousImage();
                // parent.print(1);
                // t.sleep(1500);
                ml = 0;
                cresc = 0;
            }
/*******************************************************************************************************************/
```

Fig. 10: *Hand detection using OpenCV Object Detection*

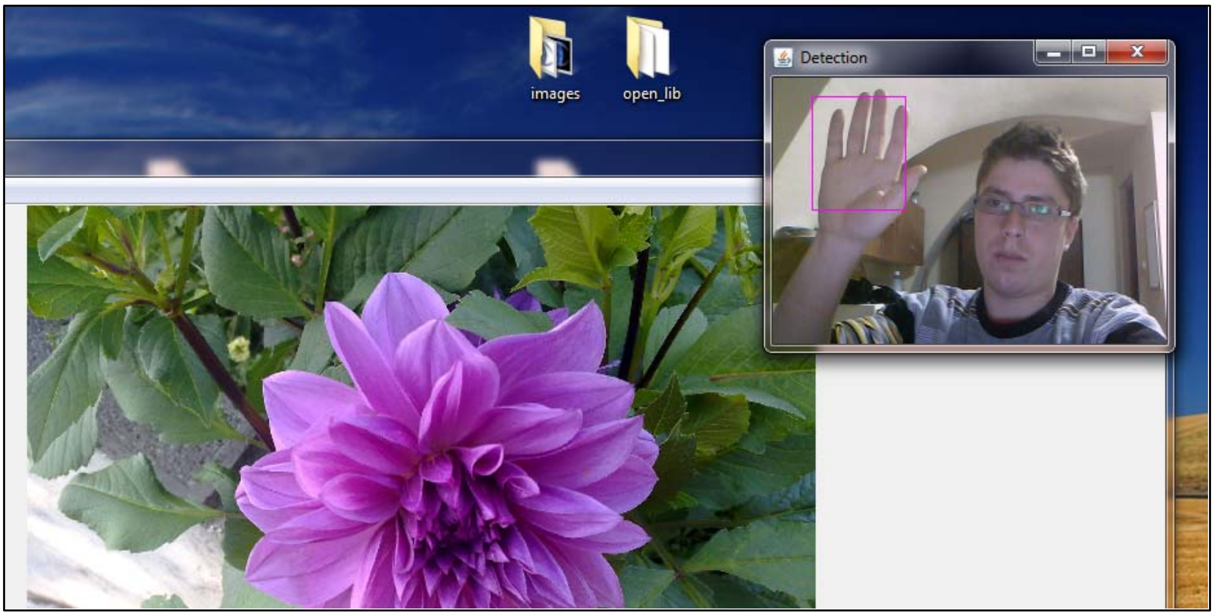


Fig. 11: *Final hand tracking used in OpenImage*

4. Conclusions and future development

The 'OpenImage' project presented in this paper and conducted as part of practice during my *Bachelor-Thesis* is the most significant and original part of the work characterized by design, research and programming.

Today, the colors and color images are widely used in presentations, to illustrate textbooks and advertising, marketing, trade and commerce, television, in everything involving the Internet and Internet communications, etc.. Basically, today's society is dominated by the color and color images.

Also, the control of applications through gestures is an evolving field with uses in video games, medical software, robotics, virtual reality, simulators and so on.

Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a richer bridge between machines and humans than primitive text user interfaces or evenGUIs (graphical user interfaces), which still limit the majority of input to keyboard and mouse.

As a last minute update, in this program, a Text to Speech Synthetiser was implemented, in order to further connect the user with the application.
This project can also be folowed from: https://sourceforge.net/projects/openimage/

As we can see, most objectives have been met, but future developments follow:

- Implementation of an intelligent zoom
- The introduction of language packs
- Implement a system for recognition of several gestures
- The ability to run other media (sound, video etc.).
- Implementation of interaction in the gaming area
- Voice recognition

References

[1] Articles related to "*Java 2D Image Processing*": http://www.javaworld.com/, visited Dec. 2010
[2] Bruce Eckel. (E-Book*) Thinking in Java (2nd Ed). pdf*
[3] Bruce Eckel. (E-Book) *Thinking in Java. pdf*
[4] Cristian Frăsinaru (E-Book), *Practical Course Java.pdf*
[5] Daniel Liang, (E-Book) Prentice Hall, *Introduction to Java Programming*
[6] Eclipse Foundation, http://www.eclipse.org/
[7] Hands detection demonstration: http://www.andol.info/hci/1116.htm, retrieved March 2011
[8] Java Neural Network Neuroph: http://neuroph.sourceforge.net/, visited March 2011
[9] Java Tutorials and Demos: www.java2s.com/Java2D.html, visited Dec. 2010
[10] John C. Russ, (E-Book) *The Hand book Image Processing, Third Edition.pdf.*
[11] Official Oracle Site: http://www.oracle.com/us/technologies/java/index.html
[12] OReilly, *Java 2D Graphics*, O'Reilly Media, May 1999, ISBN 978-1-56592-484-0
[13] OReilly, *Learning OpenCV*, O'Reilly Media, 2008, ISBN: 978-0-596-51613-0
[14] Ralf Fabian, *Practical Graphics Course*, "Lucian Blaga" University, Sibiu
[15] Sams, (E-Book*) Data Structures and algorithms in Java.pdf*
[16] Stefan Tanasa*, Java from 0 to expert,* Polirom Iasi, 2008
[17] The HandVu Vision-based hand gesture interface: http://www.movesinstitute.org/~kolsch/HandVu/HandVu.html, retrieved February 2011.
[18] Wikipedia the free encyclopedia, http://en.wikipedia.org/wiki/OpenCV, visited Jan. 2011
[19] WK Pratt, *Digital Image Processing*, *2nd Edition*, John Wiley & Sons, New York, 1991.
[20] Bota Florentin, *Data concealment using digital images*, National Students Conference ICDD, Sibiu, 2009

Bota Florentin
"Lucian Blaga" University
Faculty of Sciences, Informatics
Sibiu, 5-7, Ioan Ratiu str.
Romania
botaflorentin@yahoo.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Disconnected Recordsets and their Implementation in Java

Vasile Braileanu
Teacher Coordinator: Adina Cocu

Abstract

This paper describes the disconnected recordset approach for database access. Also, it describes a Java library written by author from scratch, who implements some of the interfaces needed for disconnected recorsdsets in Java. Similar technology is available for almost all databases. Although a database is designed to be connected all time with the user, producing the data which user needs to consume, sometimes is most efficient to disconnect temporarily the user data from database and use data disconnected from database. Later on, data changes can be reflected in database implementing specific synchronization schemas in database server part.

Introduction

Today's businesses and activities are strong tied with computer databases, which is one of the most studied and used concept in computer technologies. From their beginning as flat text files up to modern databases, they all share the same need: to store, analyze and modify in shortest time all information stored. Major software companies have developed their own databases, like Microsoft, Oracle, IBM to mention a few. Free non commercial databases are available too. Also, linked with databases, there were developed a lot of software libraries for administering and accessing the databases in different programming languages and operating systems.

In this paper an aspect of database synchronization and a possible solution is described, used sometimes to solve some common issues. The same concept, of data disconnected from source is used in human history to access other information, like newspapers, books, sound and video records. Sometimes this method tends to be slow, but the concept exists and it is used because of his simplicity. On the other part, on a day to day basis there are heavily used systems that works connected with the user, like telephony, radio and television.

One of the important aspects of data producing and consuming is the connection type between data producer and consumer. Considering the time of connection, a consumer can access the database permanently (asynchronously) when needed but also it can “disconnect” from the data source and access the information “offline”. Later on, the information updated in the disconnected database can be synchronized with the main database. There are scenarios when this later approach is feasible and needed.

In section one, Database Synchronization describes the concept of data synchronization and some of the problems which can occur on using a permanent asynchronous connection between database and user.

Section two, Implementing the solution is reflects the steps which conducted to the solution implemented in Java, and the recordset storage file format.

Section three explains the implementation technologies used, the script used in server side and similar implementations.

Section four is the case study of a database who cannot be accessed remotely all the time.

Section five, Performance and Issues highline some of the gains and the loses of the disconnected recordsets.

1 Database Synchronization

Databases are one of the most important objects used everyday. Even if we don't use databases directly, they are everywhere. Today, any real world application uses at least one form of data persistence. The Relational Database Management System (RDBMS) is the most used persistence storage mechanism and uses SQL for query and data manipulation.

In current database implementations, a connection must be established with the database server. Exceptions are the embedded databases that are linked directly with the applications and are installed with them. Also, to use the database we send SQL or native commands to the database asynchronously. That means we need the connection to manipulate the database.

In practice, sometimes there is a need to read, update and insert a new record even if a link with the database cannot be established.

On the other part, anyone who uses databases will probably have to work with a particular recordset over a period of time. However, it would not be practical or desirable to maintain an open connection to the database all time, because this would use server connection resources.

Here are few simple examples:

- The database at an ISP cannot be accessed remotely because of the database access rights. But, the database server can be accessed from web pages inside web server. That means a database connection can be made locally in server. The database can be accessed on regular basis writing server side software that accesses the database. The server side part can receive the commands and execute, later sending results back to the caller. This can be easily done sending GET and POST messages to a web page that is backed server side.
- At a supermarket sales point, there are cash registers. Cash registers are connected with a database all the time. What happens if the server is offline? Cash registers cannot send their data in this scenario. We need a cached record for the payments at the cash register. This cached record can be stored in cash register computer then updated in database when the server is online.
- In a remote area, a sales point is connected through a wireless connection to the Internet and from there to the database. In case the wireless connection is not available, the database cannot be updated. The records are stored locally until the wireless connection will be available in order to update the database.
- Accessing the database with many connections, especially when we need to query it will slow down the database server. In some applications will be more efficient to store locally the dataset needed. For example a localized application, where all the strings for a language must be fetched from database, the translation time will be considerably reduced if there is only a database query which maps the strings for a particular language.

- Finally, creating a database in a small device, maybe mobile phone with a RDBMS will be cumbersome (and maybe impossible) if the database is small and there is no disk space in the device.

All these scenarios are somehow common in real world. If we want to share information, let say an article, we can print it. From printed form, it can be read by many but when concurrency is high the information cannot be accessed easily.

That is exactly the same as database server bandwidth available. Of course, today databases are state of the art programs, as we expect from them, but the above scenarios are not uncommon. We can share the bandwidth but at one point the database server can crash because of many connections (or at least will slow down). Some of the database operations can be done in client side if we want to.

2 Implementing the Solution

Considering the scenarios described above, we can imagine a result set stored locally in a file. Also, because this result set must be accessed through a connection, it will be useful to be represented in a form of a real recordset identical with the database recordset so there is no need to implement the whole database access programs again.

Because in Java is implemented JDBC, it will be useful to access the recordset in identical form with the other database recordsets, thus creating a recordset from this file in same manner.

In Java, the user can take advantage of already available Java packages to create recordsets disconnected form the main database. There are also other databases that offer natively disconnected recordsets.

2.1 Introducing the disconnected recordset notion

These specific recordsets are, as the name suggests, recordsets that have been disconnected temporarily from the server, thus allowing the user to work off-line and move freely between records.

If a disconnected recordset is created with write permissions, the user can also add, modify or delete records. These changes will be cached locally, mainly in files and they will not affect the main database. Later on a connection can be re-established to the database, which can then be updated with the changes.

One possible problem which can arise is the possibility of records conflict with other recordsets, online or offline. That could happen also in non transactional databases.

One requirement of disconnected recordsets is that they must be maintained by the database client, rather than the database server.

2.2 File Format

Data can be stored in files in any format, but XML is one of the most heavily used format today and many of the actual technologies uses it to store information. In practice, XML will be a common choice. Today, many databases can export tables and other data as XML files.

Considering XML penetration in network transmissions (even Apache helicopters are using XML streams for data communications between helicopter and ground), XML it is a smart choice.

Being portable and simple, XML is used both in files and streams. The disadvantage of space is somehow tempered by the simplicity of format and human readability.

2.3 Format Manipulation

Because XML is a very used format and the algorithms for manipulating it are very simple, today there are libraries for XML manipulation in all programming languages.

In Java we have multiple choices for parsing and generation. A standard library in Java, which is included in JDK is SAX. SAX model is based on events generation, like other implementations in other languages. In implementation described in this paper, SAX was preferred because the structure of the XML was very simple and because is faster than DOM. Also, DOM uses a lot of memory which can be a limitation for storing data organized as tables like we need in a recordset. DOM uses a tree for supporting the XML content and that tree resides in memory. Using SAX accessing the information is faster, reading the whole XML file in one step then preserving it in memory. For simple and small tables this will be a simple task.

DOM, on the other part is more useful when we need a complete structure reflecting the elements inside XML.

2.4 Programming Language

What programming language is the most used today? The simple answer is Java. There are a lot of reasons why many programmers chose Java. To mention a few, is portable, offers a superior speed than scripting languages being compiled for JVM, forces exception treatment and has access protection to code. A lot of code was written in Java, both for desktop and server and there is a tremendous quantity of documentation and samples. Also, the JDK is free to use, being shipped with many free available IDEs.

A natural choice, Java was chosen to implement a disconnected recordset with XML input and output.

3 Implementation

Disconnected recordsets are not new. Few years ago, Sun Microsystems included in their SQL libraries for Java some implementations that support them. At the top of them is a class, WebRowSetImpl who implements WebRowSet interface. It is available from JDBC 3.0, which was introduced in JDK 5.0. Someone who uses this class will notice that WebRowSetImpl is proprietary and it could be removed from next versions. That not happened yet but happened in the future. WebRowSet implementations can offer serialization if needed, so they can be sent through networks or stored on disk.

We could also choose ADO 2.0 and stick with Microsoft technology to create disconnected recordsets or any other available technology if we want to.

As a sample code, a simple WebRowSet implementation was written, who contains base functionality for the recordset, including common data insert, update and delete, input from and output to XML. Due to the complexity of the code the current implementation does not store the data streams and national strings or blob and clob fields. Future implementations could solve this. Also, for a better functionality it will be useful to write specific implementations for databases. Some of the databases use specific date format which cannot be easily detected.

A simple PHP script was also written for MySQL database to provide a very simple way to receive and send database commands and data through GET and POST methods, generating simple XML pages conforming to Java WebRowSet XML schema.

4 Case Study - A Database Server cannot be accessed remotely

As a case study, it was considered a MySQL database who cannot be accessed remotely because the lack of rights. The database was located in a remote server who can be accessed only server side. It is the standard case of web server serving pages at a web services provider.

For testing purposes, a small Java program sent an SQL command through POST, querying the database and returning the recordset. Server side script, written in PHP, returned in HTTP response the WebRowSet XML schema (webrowset.xsd) conformant XML. WebRowSet filled the records and modified some of them, saving another XML file. This last recordset, stored in XML can be accessed like other recordsets from other databases, even to be sent back to the server if there is a service that can analyze and modify the real table in database.

Testing procedure:

1. Sample query (sent to PHP script on remote server): ”select * from tranzactii”
2. XML recordset has been sent back to Java program (according to standard schema)
3. WebRowSet implementation written in Java executed some operations on recordset and returned the new XML as text file.

Commands executed:

```
WebRowSetVB wrs = new WebRowSetVB();//implemented disconnected recordset
Reader buffer = new BufferedReader(reader);
wrs.readXml(buffer);

//test delete
wrs.absolute(2);
wrs.deleteRow();

//test insert
wrs.moveToInsertRow();
wrs.updateInt(1, 10000001);
wrs.updateString(3, "test..........test");
wrs.insertRow();

//test modify
wrs.absolute(1);
wrs.updateInt(1, 20);
wrs.updateString(3, "modify test");
wrs.updateRow();

//write result
FileWriter fstream = new FileWriter("test.xml");
BufferedWriter out = new BufferedWriter(fstream);
out.printtln(wrs.getXMLString());
out.close();
```

XML response after recordset commands:

```
...
<modifyRow>
      <columnValue>1</columnValue>
      <updateValue>20</updateValue>
      <columnValue>1</columnValue>
      <columnValue>2536 3040 3020 1034</columnValue>
      <updateValue>modify test</updateValue>
      <columnValue>1</columnValue>
      <columnValue>Mon Dec 20 06:48:00 EET 2010</columnValue>
      <columnValue>0.6</columnValue>
</modifyRow>
<deleteRow>
      <columnValue>2</columnValue>
      <columnValue>1</columnValue>
```

```
        &lt;columnValue&gt;2536 3040 3020 1034&lt;/columnValue&gt;
        &lt;columnValue&gt;1&lt;/columnValue&gt;
        &lt;columnValue&gt;Mon Dec 20 06:50:49 EET 2010&lt;/columnValue&gt;
        &lt;columnValue&gt;0.6&lt;/columnValue&gt;
&lt;/deleteRow&gt;
...
&lt;insertRow&gt;
        &lt;columnValue&gt;10000001&lt;/columnValue&gt;
        &lt;columnValue&gt;&lt;/columnValue&gt;
        &lt;columnValue&gt;test..........test&lt;/columnValue&gt;
        &lt;columnValue&gt;&lt;/columnValue&gt;
        &lt;columnValue&gt;&lt;/columnValue&gt;
        &lt;columnValue&gt;&lt;/columnValue&gt;
&lt;/insertRow&gt;
...
```

5 Performance and Issues

5.1. Benefits

- Considering the application area and memory involved, implemented WebRowSet contains a vector of records in memory. All data is stored in memory, which makes sense because the recordset is not large.
- XML format is very simple to manipulate, consisting of text fields.
- It uses standard schema and other applications can use the XML format generated. Using a standard schema facilitates the transfer between generated format and other formats.
- Less database traffic, only the selected information is sent through network.

5.2 Liabilities

- XML file format is large and this can be an important factor on distributed applications. Data sent through network can be sent also in a binary form or at least compressed on server and decompressed on client and vice versa.
- As we expect, storing in memory all values from recordset limits the application area, and the supplementary data coming from field properties and result set properties (metadata) will slow down the application. Because the memory in client is not as large as the server's, and usually the server only caches a small chunk of recordsets or none the application area is limited to those applications which stores in memory only a small part of the database. Similar server caching schemas can be used to reduce the memory used.
- Accessing data is slower than in database server. Database servers are powerful computers comparing with client machines, data is in binary format and is cached inside server memory.

Conclusion

Described implementation is similar in functionality with WebRowSetImpl from Sun, implementing the same interface. The common XML content used as output and input format simplifies the using of the disconnected recordsets. Because disconnected recordsets need to be synchronized on the server part and the implementation depends on the database, implementation use some PHP server side code for connecting a MySQL database. Also a mapping between MySQL data types and JDBC data types was created to permit the creation of the recordset. In the future versions the implementation could be completed with data streams to permit the storage of objects in serialized form.

Acknowledgement: I wish to thank Lect. Dr. Eng. Adina COCU for her patience and guidance in writing this paper and the programming code associated with it. Also I wish to thank her for his Java course which we taken in 2010 at the University “Dunarea de Jos”. I wish to thank for programming course to Asist. Drd. Eng. Diana STEFANESCU (C/C++), and Associate Prof. Dr. Emilia PECHEANU (Linux).

References

[1] Java API
[2] Deepak Vohra, *Oracle Web RowSet,* http://www.packtpub.com, 2009
[3] Infinite Software Solutions Inc., *Tutorials – Using disconnected recordsets,* http://www.devguru.com
[4] Sharad Acharya, *Making the Most of JDBC with WebRowSet,* http://onjava.com, 2006
[5] McObject LLC, *McObject’s eXtremeDB High Availability Embedded DatabaseGains Role In Boeing AH- 64D Apache Longbow Helicopter,* http://www.in-memory-database.net, 2004

VASILE BRAILEANU
Low Danube University
Faculty of Computer Science
111 Domneasca St. , Galati
ROMANIA
E-mail: vasilebraileanu@yahoo.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Securing stored passwords using cryptographic techniques

Bucur Bogdan Ion
Teacher Coordinator: Lect. PhD. Gabriela Moise

Abstract

The necessity of information encryption exists almost since the beginning of human kind. People always felt the need of protecting important information from falling into the wrong hands and this need has increased in these modern times more than ever.

The technique of cryptography is used in various areas of computer science and in Internet applications. One of its uses is successfully storing a password in such a way that even if the site is hacked and the information in the database is leaked, the hacker will not obtain a clean list of passwords and it would be extremely difficult, if not close to impossible, for him to obtain the original password of a user. The methods used have their advantages and ways of discouraging hacking, but they also have their flaws.

In this paper these methods will be presented, as well as why some of them are not completely safe, and a way to increase their efficiency will be suggested and thoroughly explained.

1 Introduction

Data encryption is not a new technique, the necessity of text encryption being around since the oldest times. By securing the data sent or stored, its confidentiality is guaranteed up to a certain point and any unauthorized attempt to discover restricted information is discouraged.

Having its fundaments in math, cryptography has easily found a role in the computer science world, by having practical applications in a great number of areas. One of these applications it's how to store in the safest way possible a user's password so that an eventual breach of security in the site's code or its database would not reveal to the attacker the user's passwords. This is essential not only because a hacker might discover the admin password of a webmaster and take control of the site, but also because it could find out the passwords of all the users, as well as information about them. By using this knowledge and by taking advantage of the fact that most people often use unique passwords for the sites they frequent [1], a single hacked site can lead to the compromise of most of the accounts that someone can have on the Internet, if not all, from e-mail accounts that can be used to send spam mail to Internet Banking or online payment sites, that can cause a serious loss of money.

2 Methods of securing user passwords

2.1 Encrypting a password

Currently, the most popular method of storing a password in a safe way is encrypting it using certain algorithms [2], some of the most used being MD5, SHA-1 and recently SHA-2. The first of them, MD5 (Message-Digest algorithm 5) it's often found integrated in various site scripts used on the Internet (for example MyBB, vBulletin and SMF for forum scripts, Joomla, PHP-Fusion and Drupal for portals and Wordpress for blogs).

MD5 was designed and created by Ron Rivest in 1991, in order to replace MD4 and can process messages of a variable length into a string of characters with a fixed length of 128 bits (16 byte). In order to accomplish this, the algorithm divides the original message into chunks of 512 bits. In the eventuality that the length of the message isn't divisible to 512, the message is padded by adding a single bit 1 followed by as many zeroes as it's necessary to bring the length of a message up to 64 bits fewer than a multiple of 512. These last 64 bits are filled up with a 64 bits integer representing the length of the original message in bits.

SHA-1 and SHA-2 (Secure Hash Algorithms) are similar algorithms developed by the American Agency NSA (National Security Agency) and they use the same principle, having instead a 160 bits output variable and not a 128 bits one, being especially used in security applications.

This kind of algorithms always generates the same unique string of characters for a certain input data. For example, the string *abc123* encrypted using MD5 will always return *e99a18c428cb38d5f260853678922e03* or *6367c48dd193d56ea7b0baad25b19455e529f5ee* if encrypted with SHA-1 [3]. Therefore, in the database are stored the hashes (encrypted passwords) and when a user types his password, it's being encrypted and the obtained hash is compared to the one in the database. If these two values match, this mean the correct password was used. In the eventuality that someone gets access to this list, including here the site's webmaster too, it won't be possible to find out the passwords used by the users unless you apply cryptanalysis techniques on the hashes.

Although it's still being used by a significant percentage of sites, starting with the year 1996 there have been found weaknesses in the MD5 encrypting algorithm [4], leading to its replacement in many applications with the algorithm SHA-2. However, these discoveries do not affect that much the probability that a password encrypted with MD5 will be easily cracked, instead they refer to the possibility to discover different input data that will return the same output after being encrypted [5]. The main weakness of this password securing technique, either it's encrypted with MD5, SHA-1, SHA-2 or any other similar algorithm, it's represented by the so-called *Rainbow Tables* [6]. The Rainbow tables are used to store hashes and contain string of characters (possible passwords), as well as their hashes. Such a table is being generated by using the following method: starting from a certain plaintext, its hash is calculated. A reduction function is the applied to the hash, afterwards it's also transformed to plaintext and then hashed again.

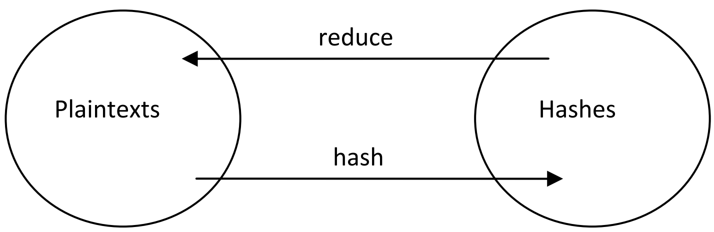


Fig. 1: generating a Rainbow table

Let's suppose for example that we want to generate a Rainbow table for the numeric passwords with a length of 6 characters, encrypted using MD5. By starting from *123456* we will obtain a first hash of *e10adc3949ba59abbe56e057f20f883e*. A reduction function is then applied to this hash, function that may vary according to the purpose of each table. In this case, such a function might pick the first 6 numbers from the resulting hash: *103949*. This string of numbers obtained from the hash is also encrypted and the process can be repeated as many times as we want, allowing the "storage" of even millions of hashes in a single chain.

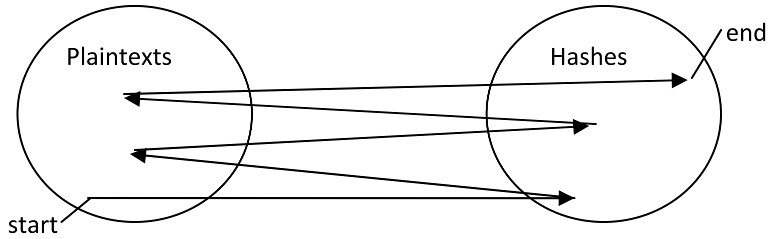


Fig. 2: generating a hash chain

Having a hash whose original plaintext we want to find out, we proceed as follows: the hash is compared to the ones in the table. If there's no match, the hash is reduced (using the reduction function), transformed to plaintext and then hashed (encrypted) and the searching process resumes. When the hash matches the final hash in a chain, that chain will contain the plaintext and from there on is as simple as recreating the chain: you get that chain's starting plaintext, and start hashing and reducing it, until you come to the known hash along with its secret plaintext.

Therefore, if the hash from a hacked database is found in such a rainbow table, the original password is automatically found out. Although the rainbow tables are hard to generate because of the time and storage space needed, the process of cracking a password becomes extremely easy and it resumes to a simple return of a value from a table.

In order to prevent this we can add a *salt* to the password. In cryptography, the term *salt* it's used to define a random string of bits that are added to a password before encryption. This way the password's length and complexity can be increased and therefore decreasing the probability that a rainbow table will contain its hash. For example, many php scripts use the following method to generate a hash:

```
function salt_password($password, $salt)
{
   return md5(md5($salt).$password);
}
```

This method increases the degree of security in more than one way:

- the password becomes longer and more unpredictable. Usually, a rainbow table is generated either based on a password dictionary (a list of words, terms and strings with a high probability of being used as passwords), either by automatically generating as many combination of character strings as possible, values which are then encrypted. So, if there is a great probability for a rainbow table to contain the hash for a password like *abc123*, we can't say the same thing about a password like *abc123e99a18c428cb38d5f260853678922e03*;
- even if the original plaintext corresponding to a hash is revealed, it doesn't correspond to the password used by the user, but to another hash that must also be cracked;
- in order to discover the original password, the knowledge of the password's hash is not enough, you must also know the salt used for its generation.

2.2 The weakness of storing passwords as hashes

The method described earlier may seem powerful due to the fact that just by adding a salt, combined with encrypting again the password with the same algorithm, it greatly increases its efficiency. Also, creating a rainbow table is not an easy task and a simple modification to the function that adds the salt to the password can make the hash virtually impossible to crack. There are however some issues that can compromise this method.

A user's MD5 hash can usually be found out in two ways: you can either "steal" the cookies from his computer, cookies that often contain the password's hash in order to auto-login on certain sites, either you get somehow access to a site's database. The first method is somewhat harmless on a large scale because rarely someone bothers to find out a single password's hash, especially since there are better and quicker ways to do that. The real problem is when a site's database is compromised, in particularly the user table that contains data like e-mails, usernames, password hashes and the salt (or salts) used.

There are many ways to obtain access to a database, such a task being much easier to accomplish than to obtain access to the files on the server. Either the access to PhpMyAdmin it's restricted accordingly, either the webmasters saves on the server a backup of the database in predictable places (for example, some php scripts offer this feature, by saving the backups in standard directories with standard names and therefore saving the file *database_name_current_date.sql* in the folder *backup*) or the data sent by GET or POST is not verified and you can use a SQL injection. The conclusion would be that there is no site with a 100% safe database.

Also, since a great percent of the sites are based on public scripts, it's enough to discover a vulnerability in a script in order to compromise a couple of hundreds, thousands or even millions of sites. For example, the Joomla platform has been downloaded over 21 million times and the 3.0 version of Wordpress over 32 million times, so we can safely assume that there are certainly a couple of million of sites that use the same script and the same password encryption techniques.

Of course, this doesn't mean that the passwords stored by such a site can be easily cracked. Let's take a look at the way MyBB (My Bulletin Board – a well-known php script used for forums) secures a user's password [7], method also used by other forum scripts:

```
function salt_password($password, $salt)
{
   return md5(md5($salt).$password);
}

$pass = $mybb->input['password'];

$md5pass = md5($pass);
$salt = generate_salt();

$salted_pass = salt_password($md5pass, $salt);
```

Therefore, the hash in the database corresponds to the hash of the original password encrypted with MD5, to which it's added the salt's hash. At a first sight, it seems like a string of characters that wouldn't mean anything to anyone and that no one would think of including it in a rainbow table. However, this method is flawed.

A database contains the salt initially used in order to obtain the same hash each time, based on the original password. By knowing the salt and therefore its encrypted value and by taking advantage of the fact that it is easy to find out how the hash stored in the database is generated, due to the fact that this scripts are public and are meant for the webmasters without advanced programming skills, this meaning a very low chance that they will modify the script files, one can attempt to generate a rainbow table specially designed to decrypt a database obtained trough hacking techniques. While it is true that such a task is not an easy one, the benefits would consist in a list

of at least a couple of thousands of e-mail and password sets, making it highly probable that some people will make the necessary efforts. If we take a look at the site freerainbowtables.com, a site that uses distributed computing techniques, allowing anyone to install a software that will make its pc working for that site in its “spare time”, to generate hashes.
This site has rainbow tables for MD5, NTLM, LM and SHA-1, tables that can be downloaded by anyone, with a decryption success rate of almost 60% [8]. If we take this into account, as well as the arguments presented before, finding out a password encrypted even more than one time doesn’t seem impossible anymore.
It can be observed that the main flaw of this method is the fact that a site stores all the data necessary to generate an unique hash based on a given password and if this data is leaked, in theory, you can discover the password by applying the same encryption process again. In order to fix this, a very simple modification can be done to the login forms, a modification that would remove this weakness.

2.3 A new way to improve the security of the stored passwords

The idea of adding a new string of characters to a password before or after the encryption is a very good one but its flaw is that once you find out those characters added, the possibility of cracking it greatly increases.
If the salt wouldn’t be stored in the database too and would be typed by the user along with the password, this issue would be solved, the modifications done to the original password being harder to detect. However, such a method would be very stressful for the user. No one wants to basically type down two passwords, especially since the salt must be quite long and look random in order to have a real effect against the rainbow tables. Also, such a salt would be hard to remember. Anyone can remember a combination between the birth date and address for a password, but who could remember a salt like *a4kj8ilb7*? More than that, who would be willing to type it separately after the password. The method I propose could solve this problem by introducing another input field in the login panel, in which would be introduced a value that the user already knows, is unique and typing wouldn’t take long.
For example, when a user makes an account on a site, he could be asked to choose a security password, similar to the way he asked to choose one for the password recovery process. What is his lucky number? In which town was he born? What are the initials of his name?
In the case that a user would not like this extra question, he could opt out, rendering the hash list even more useless for a hacker since he won’t even know what users use security questions or not. And for the site’s login system it would be extremely easy to check if a user sent an extra answer along with his password when he logins (php *isset* command).

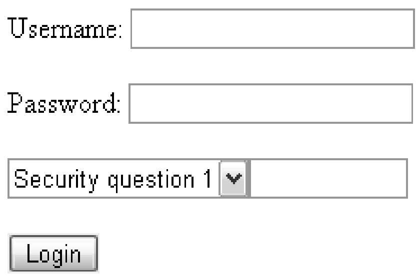


Fig. 3: proposed login system

In figure 3 such a login form is simulated. A user who didn’t choose to use the security question can just leave the field blank when he logins and the hash will be generated using a standard method. Otherwise, the hash will be generated taking into account the new variable.

Based on this new variable, the password could be modified and encrypted or unique salts could be generated for each user, having as a result a hash that could only be reduced to a combination of characters for whose cracking would be necessary the variable typed by the user, something much harder to obtain since it's not stored anywhere. Such a list of encrypted password would be absolutely useless for a hacker and finding out the original password would turn out to be practically impossible for a third party. To accomplish such a feature, a hacker would have to research each user in order to somehow deduce the value that he chose. Not only that, he won't even know if a user uses a security question or not.

This method might seems simple but what matters is that such a modification would be easy to implement and would not have the main disadvantage that using an encryption algorithm has, namely standardization, the fact that the algorithm and the hashing techniques are used on a large scale and the way it works can be found out and analyzed by anyone. When the accent is put not on breaking an algorithm, a problem whose solving would facilitate the obtaining of a whole set of passwords, but on cracking each password separately, by getting the data directly from the user, the negative effects are reduced since almost no one would try that hard for a single password.

Also, we must not neglect the fact that this way a user would not be stressed or tired. There can be developed even more complex ways of securing a password who could guarantee the safety of a password both in theory and practice, but no one wants to get trough an overly complicated login process. In other words, what makes this solution ingenious is its simplicity.

3 Conclusion

Using mathematic and programming techniques, one can create hundreds of encryption algorithms based on character or bit permutation, adding new elements and so on, whose complexity would make an encrypted password impossible to crack, but they will never be as safe and fast as a password known only by the user and based on elements that only he knows.

The methods that are currently used by the various sites on the Internet are based on very powerful encryption algorithms, that are hard to break, but not impossible. The great majority of webmasters feels safe because of the fact that no one could generate a rainbow table that would make the usage of hashing useless, but this is not as certain as it was a couple of years ago.

By implementing the simple yet effective method suggested in this article, passwords would be hashed in a much safer way without causing trouble for the user. On the opposite, it could even generate a feeling of a much stronger site customization for the user by allowing him to feel more protected and important, giving him the chance to choose his own security question, an effect that has been observed in more than one occasion on similar situations. This would also be a novelty for the login system, implying a unique way of hashing passwords for each user, based on information only he knows, a way that could not be reproduced by a hacker and therefore eliminating the weakness of a hash against a rainbow table.

The research will be continued by developing an encryption algorithm for passwords, based on this method, that will be able to generate at each step not a random string of characters, but a new string resembling a password so that even if a hacker could get pass the hash and obtain the plaintext, he could not be able to differentiate the original password by those generated by the algorithm. I also plan to implement this method in the sites I own, in order to increase their security.

To sum up, this article presents a new way to increase the safety of the passwords stored by sites, using a new method that is easy to implement and could add a new short security question to the current login forms, in the same time increasing drastically the efficiency of making the stored password hashes impossible to crack.

References

[1] R. Morris, K. Thompson. *Password Security – A Case History*.
[2] C. Wille. *Storing passwords – done right.,* 2004.
[3] http://askcheck.com/hash?hash=abc123
[4] X. Wang and H. Yu, *How to Break MD5 and OtherHash Functions*, 2009.
[5] J. Black, M. Cochran and T. Highland. *A study of MD5 attacks: insights and improvements*. 2006.
[6] *How Rainbow Tables Work*, http://kestas.kuliukas.com/RainbowTables/
[7] *MyBB's Password Encryption Method*, http://community.mybb.com/archive/index.php/thread-75972-3.html
[8] *Rainbow Tables FAQ*, http://www.freerainbowtables.com/faq/

BUCUR BOGDAN ION
Petroleum-Gas University of Ploiesti
Faculty of Letters and Sciences
Advanced Information Processing Technology Master
Ploieşti, Prahova
ROMANIA
E-mail: b.bucur@yahoo.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Rational implementation of performance testing for sites

Ksenia Burykh
Teacher Coordinator: Oleksandr Dorokhov

Abstract

Any program product can not be released without executing performance testing. This kind of testing is quite difficult, but its importance very high. We can suggest, which load pattern a local application will stand. But speaking about web sites (e.g. sites of scientific journals), it is difficult to imagine what load will withstand the web application, because it is in free access in the Internet. In such situations performance testing becomes imperative. As usual, performance testing executes on databases, which contain a lot of data (for example, 1 million of different kinds of objects with complicated indexing). So the test engineer faces at least two problems: First - how to generate so much data to fill in tested database in the short term. And second (the main one) – how to reset database to initial state after executing each load test, because there is a great probability, that the results, received from each subsequent test will be worse, then the previous. In this article I'll show, how to solve this problem with the smallest loss of resources and time using Microsoft Visual Studio for testing ASP.NET web application with deployed on MS SQL Server database.

1 Introduction

Very often the final step in the product quality control is the performance testing. There are three types of performance testing: load, stress (but usually load and stress testing are merged into a single type) and, directly, performance testing.
Performance of an application is very important for a multi-user application. Performance is not only the speed of execution; it includes the load and concurrency aspects. Performance Test is an answer for the following questions:

How can we ensure that our new application will support the expected user load?
How to avoid the issues which will arise only in real load conditions?
How to find the response time?
How to plan the capacity of the servers?

Usually, performance testing is carried out in three iterations. On first iteration the basic parameters (e.g. response time, speed, data transfer rate, bandwidth and others) are measured on tested system 'AS-IS'. On the second iteration the same parameters are measured on tested system with made improvements (for example, some code modifications: reindexing, stored procedures' upgrading). On the third iteration the received results on first two steps are analysed and compared. In my opinion, the generation of large number of initial test data (for example, 1 million of different kinds of objects) should be performed only once on first iteration, because the creation of such great number of test data in database takes a lot of time, and perform this action

twice (on first and second iteration) would have been irrationally wasting time. That’s why the only way – to reset database to initial state before the second iteration.
If the performance testing goals are to check the system behavior under heavy loads, than I recommend to reset database to initial state after executing of each load test, if you want to get adequate and truthful results.
According to the programming language of the tested system, the performance testing goals and project’s budget, the corresponding testing tool should be selected (The main such specialized applications are: Microsoft Visual Studio Testing Tools, LoadRunner, Silk, IBM Rational Performance Tester, LoadStorm, StressTester, TestComplete, TestMaker, WebPerformance Load Tester, etc.).
At the moment one of the most common development technologies is ASP.NET (ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites, web applications and web services [1]).
So, if the tested system (web site) is based on this technology – the best testing tool will be ‘Microsoft Visual Studio Testing Tools’ (Visual Studio Test edition or Visual Studio 2010 Ultimate provides the support for test automation), because both application are using the same platform, provided by Microsoft [2]. Also, load test, created in Visual Studio, can be used not only for measuring performance key indicators, but for generating the great value of test data in database either.

2 Performance tests creation and execution in Visual Studio Testing Tools

Performance testing and debugging is a major focus of Visual Studio 2010 Ultimate. Web testing and load testing have been supported in Visual Studio Team System since the 2005 release, but Visual Studio 2010 offers major improvements.
Firstly, the new Test Project should be created: Test projects are language (C#, VB) specific and created by selecting the corresponding “Test Project” template from the Add New Project dialog box.
The Test Project template will create an empty unit test by default. This file can safely be deleted. If you want to customize what is included in new test projects change the settings in Test Tools/Test Project section of the Visual Studio options.
Secondly, web performance test should be created: Web performance tests use manually written or recorded scripts to verify that specific functionality of a Web application is working and performing correctly.
For instance, we can record a test script to ensure that a user can log in to the application. Web performance tests are also composable so over time we can build a library of reusable scripts to combine in different ways to test different, but related activities.
There are several ways to create Web performance tests:

- Recording browser interactions;
- Manually building tests using the Web Test Editor;
- Writing code to create a coded Web performance test.

By far the easiest method is recording browser interactions. Recorded scripts can be customized easily. They can also be used to generate coded Web performance tests. It is generally recommended to start with a recorded test therefore we will not be discussing coded Web performance tests here.
After the web test was recorded, we should modify it: for example, a data source (some table from database) can be connected to web test, after we can change some appropriate recorded parameter values to values from DB, that will make the current test more universal.

Than the load test can be created. Load tests verify that an application will perform under stress by repeatedly executing a series of tests and aggregating the results for reporting and analysis. Load tests are often constructed from Web performance test scripts but most other test types are also allowed.
In the Load test QA engineer can specify, what web tests will be executed, how many times selected web tests will executed, how many virtual users will reform actions of web tests (fig.1).

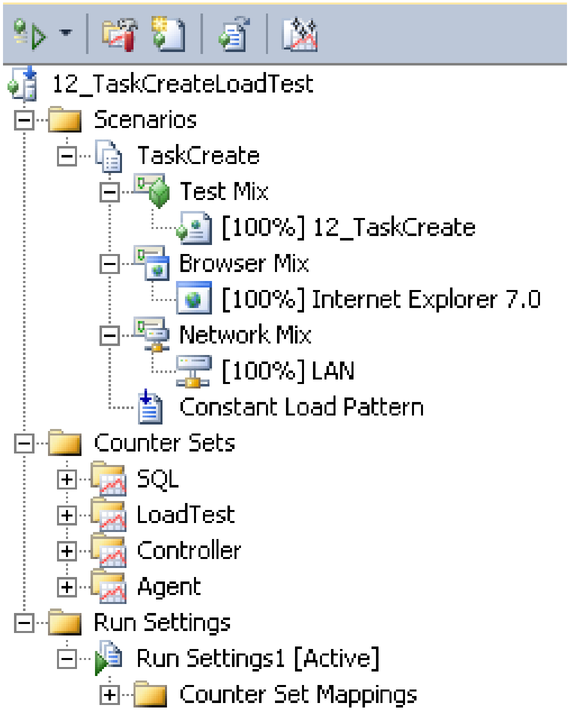


Fig.1. View of Load Test's properties

Usually, the tests scenario of load test contains several web tests (one web test executes as the main test and another – as the background one).
This approach allows to load more the tested site and, respectively, to find more problems or even critical defects.
In the load test settings we can also specify, from which browsers will the virtual requests sent, or values of what counters will be saved (For example, such counter sets are important: Memory, Network Interface, Process, SQL Server: General Statistics, SQL Server: Access Methods, ASP.NET Applications and so on).
When the load test created, we can run it. After the load test finishing, its results should be analyzed, as during the course of a test run Visual Studio is collecting performance data according to the defined performance counter sets. The Load Test Monitor is used to view and analyze the collected performance data.
By default the Load Test Monitor displays the results in graph form (fig.2). The graph view provides a way to see performance data over time. This is useful for quickly identifying possible issues. To really get into the details though we need to switch over to the tabular view by clicking the "Tables" button.

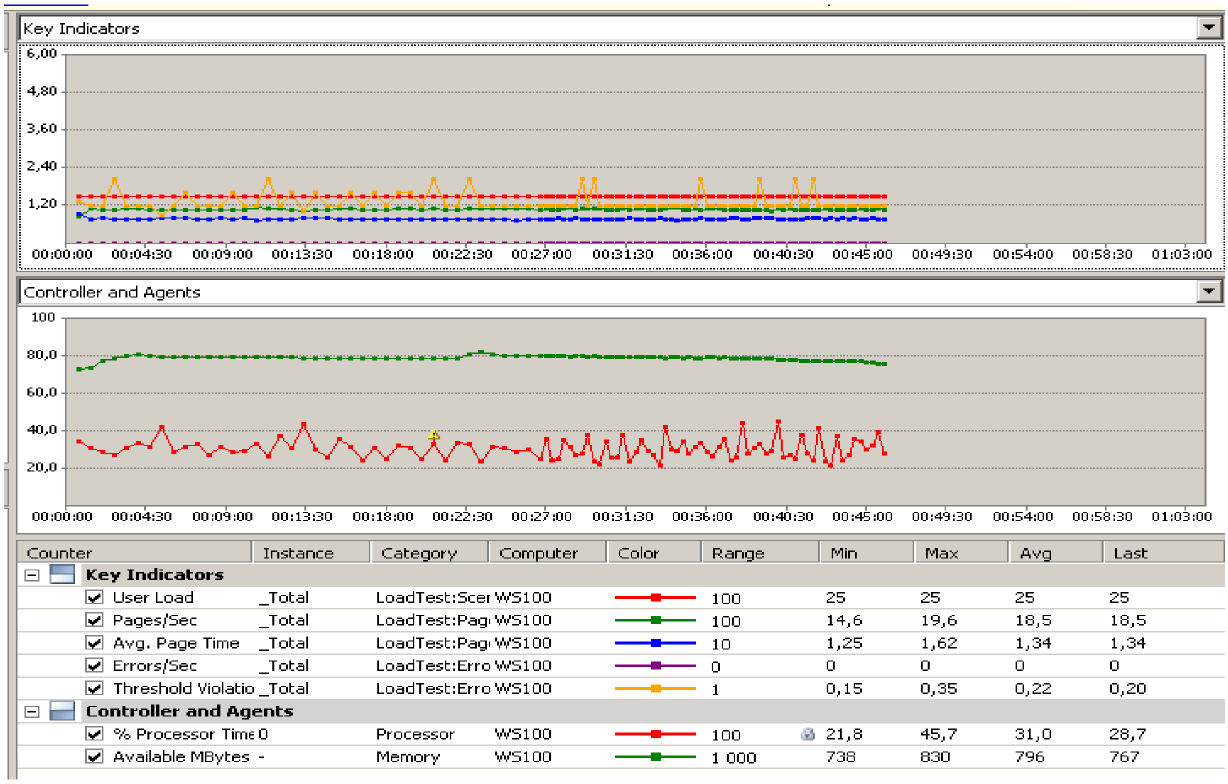


Fig.2. Graphic representation of Load Test's results

The tabular view provides a select box listing several tables:

Tests displays summary information for each test in the load test.

Errors displays details about any errors encountered during the test run.

Pages lists all pages that were accessed during a test run along with summary data such as average response time.

Requests includes details for all HTTP requests issued during a test run.

Thresholds displays information about each threshold violation encountered.

Transactions provides information about Web performance transactions or unit test timers.

Also, in the Visual Studio Testing tools we can export the results of load test (report) to Excel using 'Create Excel Report' option. It creates an excel report with multiple sheets representing average response time, runs, page time, etc.

We can create two types of reports (Trend – create an excel report with the trends in the selected run result. Comparison – create an excel report by comparing two or more run results).

3 How to optimize the performance testing

3.1. Test data generation

There are different ways of initial test data generation. For example: we can use the SQL script, which uses some existent stored procedure for creation of needed object. But using this method, only 1000-2000 objects can be created during one script execution (As my practice shows, unfortunately, when Management Studio executes a time-consuming script, the application "hangs").

I recommend using the load tests, created in Visual Studio for creation of test data. There are several reasons in favor of such method:

As usual, during performance testing the object creation, reading, editing, search operations are tested. That’s why the load test for creation some object has been already created and we can just modify it (not waste time for writing new SQL script).

Load test can create 100 000 objects easily. And this type of test can be executed at night to save working time during the day.

The test data, created via load test are more close to real objects (created manually by users), that gives an ability to simulate more real situation.

3.1. How to reset database initial state

Before the executing performance testing, we have to prepare the database, which has already contained large number of text data. As I’ve said earlier, usually, performance testing is carried out in three iterations: first – testing of the system without improvement, second – testing of the system with improvement and third – comparing of results.

So, before the first iteration, the full backup of system database should be created. This backup is the initial database state.

After the performing of first testing iteration the system database contains data, which were created or edited by load tests. We can’t use this system database on the second iteration, because it also contains “extra data” (To get correct results, the load tests on both iterations should be executed on the same environment and in the same conditions. Even small value of “extra data” can have a bad influence to the final result). That’s why we have to reset database to initial state.

On the second iteration the system database should be restored from backup, which was created earlier. Than the system can be upgraded with code improvements.

Mostly, code improvements contains reindexing and stored procedures modifications. Thus, there can be some problems with updating of already existent data. I give the examples of solving such problems below (the following actions should be performed):

All table indexes in database should be rebuild (not reorganize, but rebuild). The next SQL script should be executed:

```
USE [DB_name]
GO
EXEC sp_MSforeachtable @command1="print '?' DBCC DBREINDEX ('?', ' ', 0)"
GO
```

Than all table statistics should be updated for all tables in database. The following SQL script should be executed:

```
USE [DB_name]
GO
EXEC sp_updatestats
GO
```

The next step will be reorganization: reorganize data pages in the data file, but do not shrink the files if they have free space. In Management Studio, right click database and choose Task/Shrink/Files. In this dialog you can reorganize data pages, if there is free space in the database you should not release it. Or the use the script:

```
USE [database]
GO
DBCC SHRINKFILE (N'database_dat' , 4342)
GO
```

Only after performing this steps the performance testing ob second iteration can be started.

4 Conclusion

Performance testing can help to test any system (web site) under realistic conditions to identify usability problems and to compare measures such as success rate, task time and user satisfaction with requirements. But is hard to organize correctly the process of such testing. Only when all small details were thought at the planning stage, the desired result can be obtained. As the example, I had suggested how to optimize the performance testing for ASP.NET web application with deployed on MS SQL Server database, which will help to save time, receive adequate and truthful results and to find and fix problems (defects) before the release of program product. Also, I'm planning to investigate, how to optimize the load testing of web sites developed using JAVA programming language, because such sites have fundamentally different organization of the database from ASP.NET sites, thus, there will be other problems, which will face test engineer.

References

[1] *ASP.NET* - http://en.wikipedia.org/wiki/ASP.NET.
[2] *Microsoft Visual Studio 2010* (officcial site) - http://www.microsoft.com/visualstudio/en-gb/products/2010-editions/ultimate.

Ksenia Burykh
Kharkiv National University of Ecomomics
Faculty of Economic Informatics
61001, Kharkiv, pr. Lenina, 9a
Ukraine
E-mail: mykitana@mail.ru

CRM dedicated database architecture

Francesco Capuzzo
Teacher Coordinator: **Prof. univ. dr. Dana Simian**

Abstract

A CRM (Customer Relationship Management) software keeps trace of people, companies, events, actions in time and much more. Using a classic database approach, the information would be dispersed in dozens of tables. The present work shows how to take a smarter approach that needs less tables and that provides a flexible tracing system for all elements of the application developed on it, and much more. The ideas shoed in this paper are completely original.

1 Introduction

A conventional relational database, when designed for a CRM, is normally formed by tables containing contacts, leads, potentials, events, tasks, campaigns, calls and so on. This kind of design does not allow the user to use the application for doing what is not supposed to do. For example, keeping trace of a telephone calls campaign in which every person contacted is asked 10 questions, and the consequent answers, feedbacks and reactions, is a quite challenging job that a classic database design cannot accomplish. And an e-mail campaign, if not expected in the design of the application, is not allowed to be introduced. Instead, the database architecture needs to be flexible, small and functional. At the same time, a proper architecture can provide a huge number of metadata.
Before thinking at a new architecture, I have studied and worked with the Zoho CRM [1] and the SilverLake CRM [2] by Esox Informatica. Both are good products. The first is an online solution, accessible upon registration. The second is a product being developed in the present months. For both, the available modules are static and does not allow the user to create others.

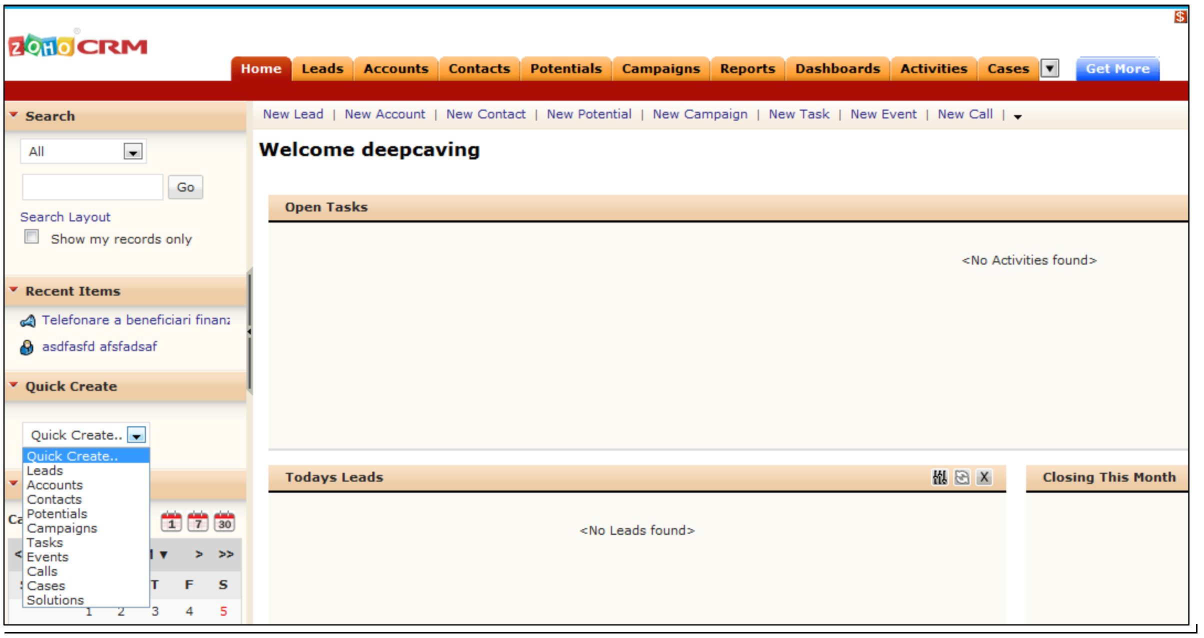


Fig.1: a screenshot of Zoho CRM. Down on the left side of the picture a dropdown list is showing the possible elements to be created by the user. Those and no more.

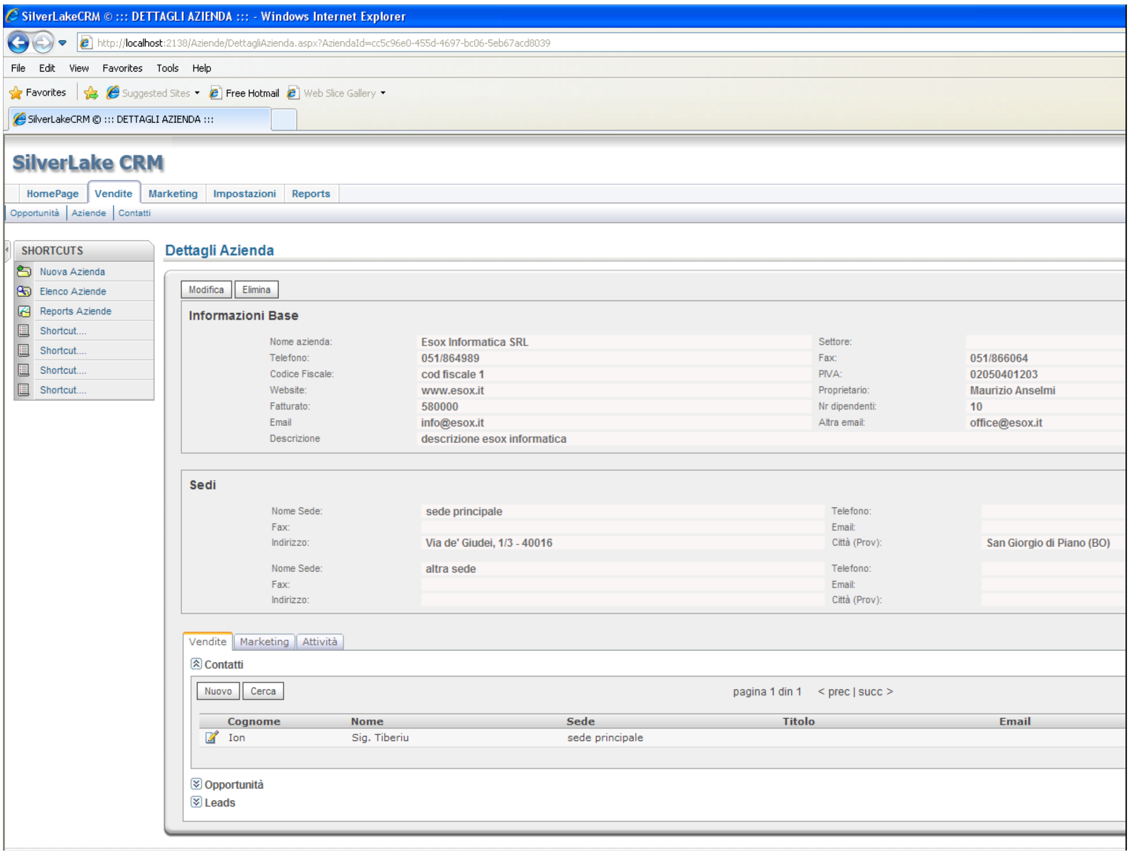


Fig.2: a screenshot of SilverLake CRM by Esox Informatica. This project is yet under construction.

2 Problem formulation

In the formulation of the problem, I have recognized two main goals:

1. Creating flexibility.
2. Being able to keep trace of each element of a project, and from it walk across the whole structure

2.1 Create flexibility

Creating flexibility means that users can create their own structures in the application. These structures can be as a simple chain in “one to one” format, or “one to many”, or “many to many”. Users can create actions of any type, sub-actions and different elements.

2.2 Being able to keep trace of each element of a project

As most actions are reactions to facts, these elements are joined by a cause-effect link. Keeping trace of these links means that users can find out all the information about every element present in the CRM application: knowing where it comes from, what follows etc. They can walk across the links horizontally and vertically.

3 Problem solution

In order to obtain the results described in the problem formulation, we first have to reformulate the table structure, going through the following steps:
The tables Leads, Contacts, Potentials will be merged, and the difference among the contents will be described in an additional field.
The Campaigns table will become Projects. This new table will be the main container for every big project recognized as a “main need” of the company.
The Activities, Tasks and Events tables will be reformulated in two new tables: Tasks and Actions. Actions will be the sublevel table of Projects, and Tasks will be the sublevel table of Actions.
In order to keep trace of all the levels involved in a project, vertically and horizontally, each record will keep record of the upper table, the same way a linked list works.

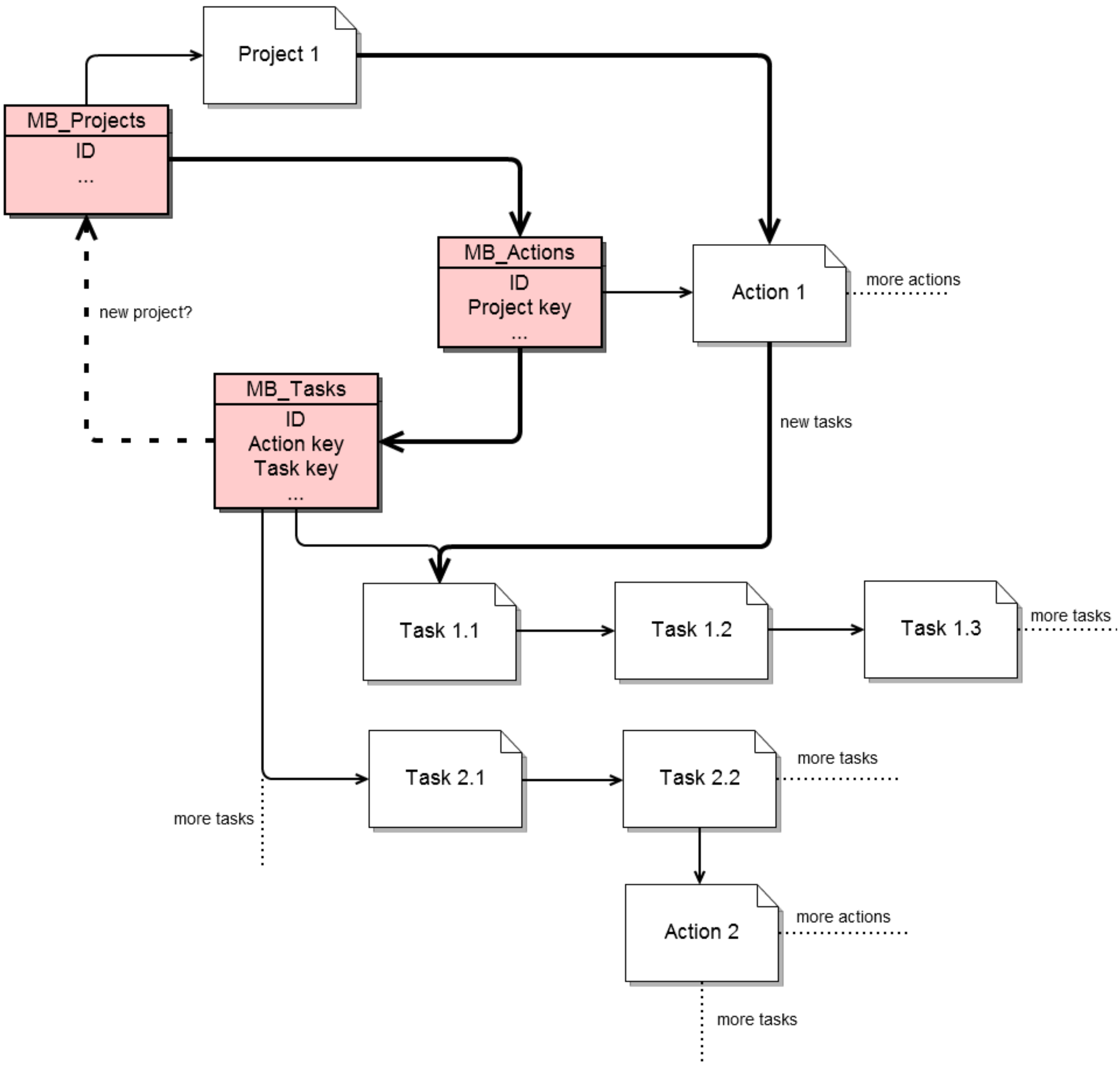


Fig.3: table structure. The base structure is composed by 3 tables: Projects, Actions and Tasks.

Starting from one project, Actions are the macro activities that are expected to achieve the main goals. Goals are expected to be described in a dedicated column of all the three base tables: Projects, Actions and Tasks. There is not a limit to the number of projects, actions and tasks. Actions can generate Tasks. Tasks can generate actions. Tasks are micro activities. Tasks can generate e new project, because they can discover a new need for the company.

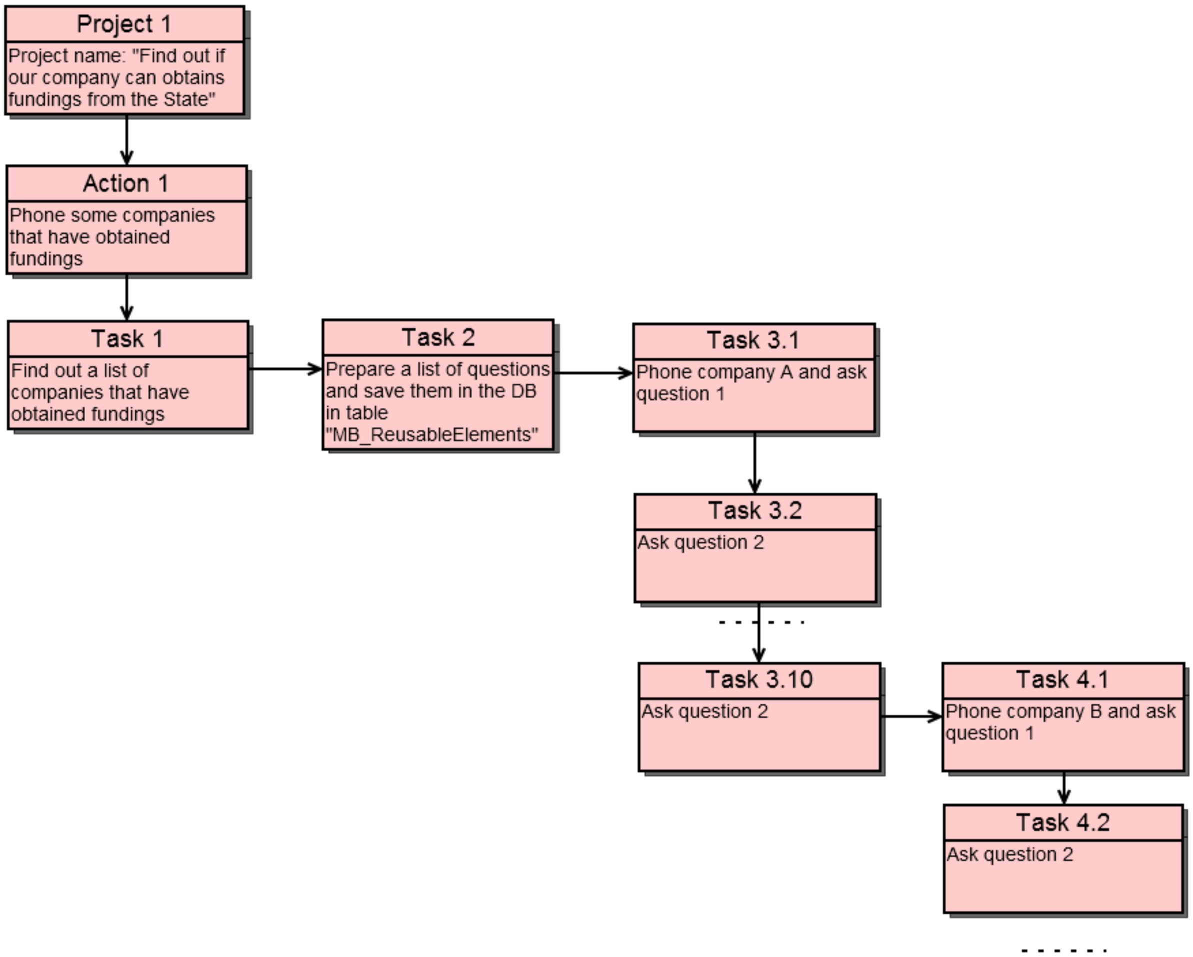


Fig.4: an example of a simple chain of actions and events. Answers to questions and their feedbacks are not included in the picture.

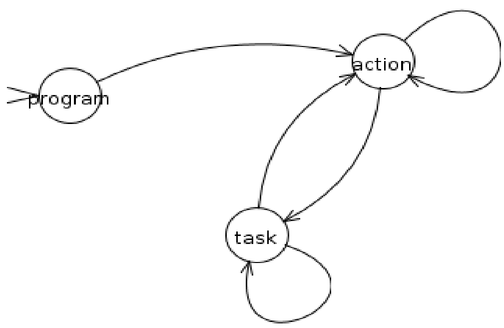


Fig.4: information flux among tables

4 Using the context as a powerful feature

As mentioned before, questions and answers are saved in separate tables. Reusable elements are not grouped in a specified group, and they are not given a group name. In every CRM, everything depends on time. Everything happens at a certain time or in a range of time, after something else has happened or before something is expected to happen. The architecture presented in this paper is extremely useful and easy to use if the application is designed to consider contexts instead of simple elements. In this way, a list of actions to be taken at a certain time will not be entered a table regardless the context, as a named group to refer to when needed; instead it will be necessary to enter it in the proper table being the user deep in the right context of the application. Normally a project has a tree structure; the user will easily go through the right branch of the tree using a simple graphic tool and will enter the data while being in the right place. The application will keep in “mind” the context, time and other metadata of these inputs. Finding out information will be easy, having the possibility to search in a tight context, a wider or wide context, search for date, user and other criteria. Using context will be an extremely powerful tool for searching and for developing business intelligence features.

5 Conclusion and future development

The presented CRM database architecture delivers an outstanding flexibility, a simpler database structure, powerful search features and business intelligence capabilities. It can be the milestone for a new generation of CRM applications. I will use the presented architecture to build an online CRM service that will provide all the features showed in this paper, and more. The user will have the possibility to trace every action/task in it's context and will be able to link actions/tasks between different contexts. A datamining tool will look for hiddend links that could potentially offer the user new perspectives and ideas for making his/her activities more profitable.

References

[1] http://www.zoho.com/crm/
[2] http://esoxgroup.eu/

Francesco CAPUZZO
Lucian Blaga University
Sciences Faculty
Informatics profile
Sibiu – ROMANIA

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Automated Number Plate Recognition in Android Mobile Devices

Ivan Danov
Teacher Coordinator: Milena Lazarova

Abstract

The paper presents an automated number plate recognition software for Android operating system mobile devices. The architecture of the software consists of two modules and is presented as well as the information processing that is divided into several steps. Each step relies on the results from the previous one so the final result is dependent on the work of all the steps in the process.

1 Introduction

The importance of mobile systems programming has emerged over the recent years as a new domain in software development [1]. Since the mobile handheld devices have limited resources – less processing power and storage capacity than large desktop computers, there are some constraints on what can be accomplished on such devices. The application of algorithms with high resource usage on a mobile devices environment is an exciting and challenging task.

The growing penetration of mobile devices in today's life and their hardware capabilities are demanding more services to be implemented for handheld devices. Since the hardware limits are pushed forward now is the right time to redesign some high resource usage applications and put them in a mobile context. Some of these applications are implementing computer vision tasks which are extremely useful when they are mobile. Using mobile devices for computer vision tasks is supported by the fact that almost all of the recently built devices have integrated camera, GPS sensor, accelerometer, Internet access via GPRS or 3G networks. That is more than enough to enable talented developers to provide users with a whole bunch of innovative services.

There are number of computer vision software used on a regular PC-based systems. They could implement face recognition systems, augmented reality systems, optical character recognition systems, image restoration systems, video tracking systems etc [2]. Some of the application areas of these systems could benefit and have potentially better usage on a mobile device.

Automated number plate recognition (ANPR) is a mass surveillance method that uses optical character recognition on images to read the license plates on vehicles [3]. ANPR technology is used widely in a variety of contexts, including both the public and private sectors. These include parking enforcement, traffic management, special vehicle access mode and congestion charging.

Usually the ANPR systems require either expensive computer platforms or are grounded to a single place – they are not mobile systems.

This paper focuses on specific computer vision problem – automated number plate recognition systems in mobile environment. The successful usage of these systems on a regular PC-based configurations could be starting point for the development of similar systems for mobile devices. As mobile applications these systems could be used as a base for parking applications, parking permission checkers, software for temporary or mobile special vehicle access checkpoints and so on.

The paper presents a software system for automated number plate recognition for Android mobile devices. The architecture of the system as well as the methods and algorithms used at each of the processing stages are presented together with some experimental results.

2 Automated Number Plate Recognition

In recent years Automatic Number Plate Recognition systems (ANPR) have become a reliable and affordable state-of-the-art technique that can be used in a variety of situations: parking applications (directly sending parking SMS for the scanned number plate or checking the parking for expired parking permissions), marketing solutions (GPS-based games), authenticity proof for vehicles.

Automated number plate recognition systems consist of two components: a camera that is used to capture an image of either static or passing vehicles and a software that recognizes number plates. The recognition software is based on utilization of algorithms that could be regarded as composed of several image processing and recognition stages.

The first step is shooting the image and the second one is the image processing and finding regions of interest. Several region of interest need to be selected since the number plate might not be detected in the first hit. These regions of interest have to be selected by a simple and descriptive set of geometric and/or graphical rules. The third step is the segmentation of the region (the number plate). After that the fourth step is classifying the segments – either neural networks or other pattern classification method can be used. That step is of crucial importance for the plate recognition system and requires the extraction of the most discriminative image features. After classifying the segmented images of the individual characters the fifth step is the syntax analysis, where mistakes from the previous steps can be found and fixed. The final step is a decision to proceed with the next selected region of interest or to stop processing and save and/or display the result from the whole process.

3 Automated Number Plate Recognition in Android Mobile Devices

As the mobile market is growing and the mobile hardware is getting cheaper and more powerful, the use of mobile devices for different tasks looks very promising. In fact today's standard hardware of a middle to high class mobile device is quite enough for high resource usage tasks – 1 GHz processor (Qualcomm Snapdragon, Samsung Hummingbird and the dual core NVIDIA Tegra 2), more than 256MB of RAM and microSD cards up to 32GB. With a proper handling of all these resource, really good results can be achieved for solving computer vision tasks such as automated number plate recognition.

Mobile devices have a wide variety of operating systems – both open source and proprietary. In order to provide wide usage of given mobile device application, the top 5 operating systems by market share should be considered. The choice amongst Android, iOS, Symbian, Windows and Blackberry's RIM[4] focuses on Android since it is an open source OS and a port of a very popular and widely used open source computer vision library – Bill McCord's OpenCV port for Android[5] and JavaCV[6] can be used.

ANPR application described in the paper is composed by two modules (fig. 1). The first module is the Android module, written in Java using the Android SDK. In this module the application could take advantage of the hardware sensors of the mobile device such as camera, GPS sensor, rotation sensor etc. The second module is written in C using the Android NDK and OpenCV. That is possible because of the Java ability to take advantage of native code using the JNI technology. The application is built using Eclipse IDE for Java with installed CDT and ADT. GCC is used as a C compiler. Other technologies that were used are JDK 6, Android SDK, a port of OpenCV for Android (in this case Bill McCord's port of OpenCV 1.1) and the latest version of Android NDK in order to take the advantage to build JNI modules.[7, 8]

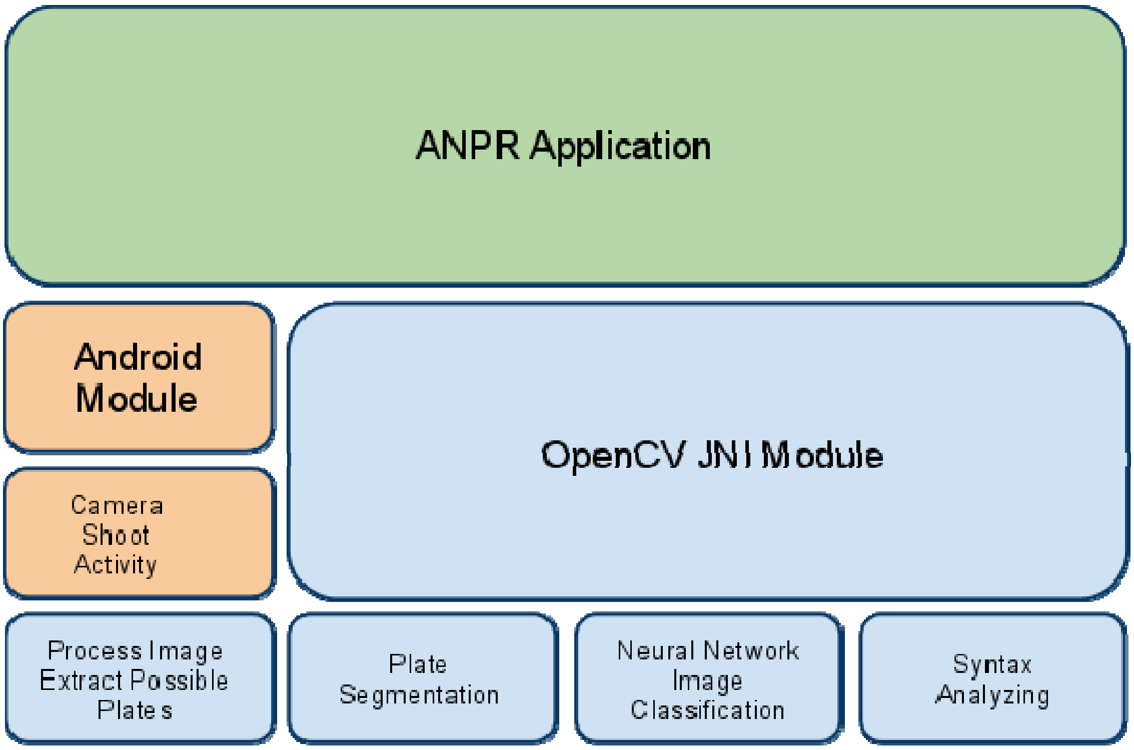


Fig. 1. Architecture of the ANPR application for Android mobile device

The Android module is specific for the application and in the simplest case it is aimed at shooting a picture and passing it to the second module for further processing. The JNI module is responsible for finding several supposed number plates which could be used for analyzing and extracting data from them. The algorithm is composed of several predefined steps: choosing ROI, segmenting, classifying, syntax analysis. Each step has it's own tasks and rely on the data provided by the previous step.

3.1 Shooting

The shooting of the image [7, 9] has to be done by the OS function calls. In Android every process is an Activity derived class with at least an overridden onCreate method. In this method every object and callbacks to be used should be defined such as Camera object setup, ContentView of the activity setup (which will be of type SurfaceView). After taking the picture and transforming it into an array by the BitmapFactory, it is passed to the JNI module for creating the OpenCV IplImage object. After that the process is passed to the second phase, which is implemented in the JNI module.

Fig. 2. Image captured by camera of a mobile device

3.2 ROI

Finding the region of interest (or several regions of interest) is decomposed into several subtasks [10]. After having access to the image, it should be transformed into a grayscale image using OpenCV functions. First edge detection using Canny algorithms and image thresholding are applied by cvCanny and cvThreshold functions and the image is converted into a binary contour image. Applying the OpenCV function cvFindContours finds all the contours in the image and analyzes them all for specific physical features. In the ANPR system finding the minimal bounding box with cvMinAreaRect2 and finding the contour perimeter with cvContourPerimeter are all the features required to analyze more that 2000 contours and select only 1 to 3 of them to be passed at the recognition step.

The simple rules to select the promising ROIs are: the contour should be with perimeter above 300 pixels (image has to be clear enough for analyzing); it should be made of at least 4 points (to make sure it has the chance to be a rectangle); the long side of the minimal bounding box should be between 4 and 6 times longer than the short side. Example result is shown on fig. 3. One of the disadvantages of the utilized ROI selection method is that not the shape of the contour, but only it's bounding box is analyzed. On the other hand the advantage is that the analysis of the bounding boxes and the ROI selection is extremely fast. If the selection results are poor the contour moments can be calculated and their orientation can be followed in order to filter the noise out of the preselected ROIs. After getting the final number of ROIs (usually no more than 3) each of them is passed to the second phase.

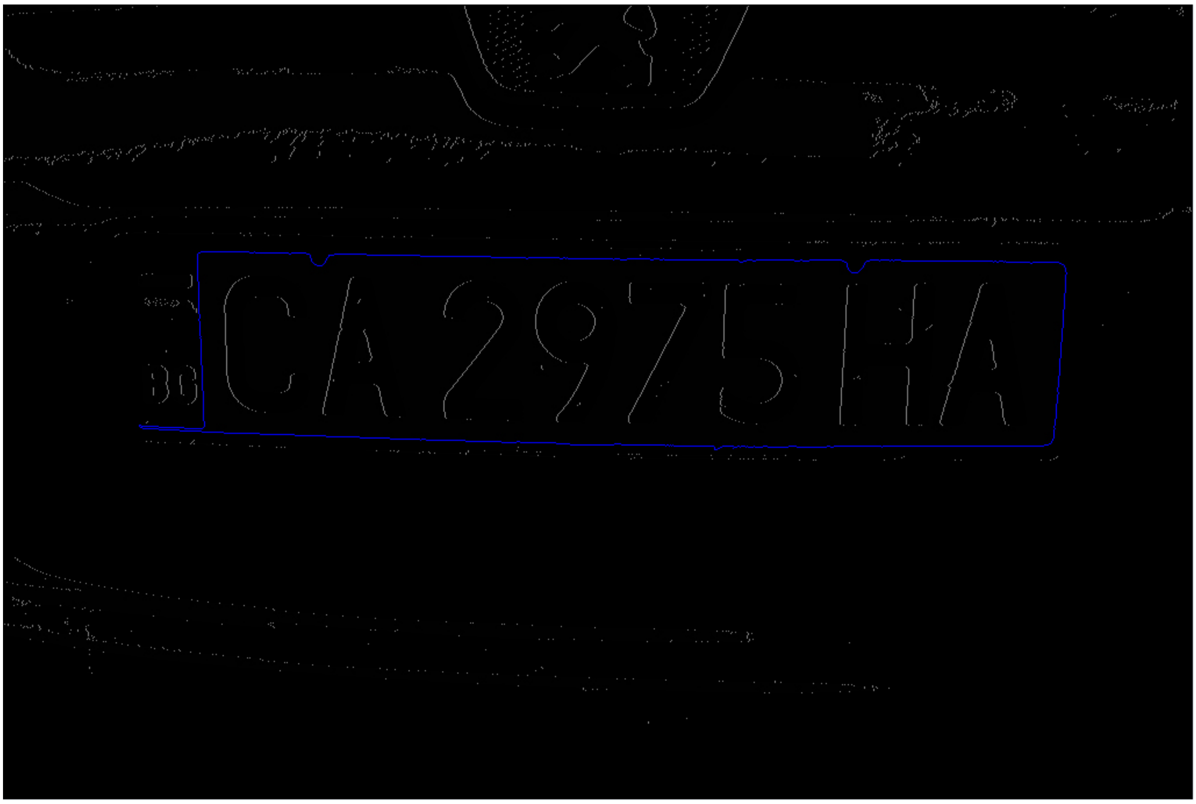


Fig. 3. The only selected contour as ROI from more 2000 contours.

3.3 Segmenting

There are number of different algorithms for segmenting of the image [11]. Here a simple but effective approach is used. After cutting the selected ROI and rotating it to be horizontal, a shrinking is applied at the image, so most of the noise pixels that are not necessary for further analysis of the image are cleared (fig. 4).

Fig. 4. Selected region cropped from the whole image

After the preselected region is presented as a single image, it is thresholded (if it's a full color image) using the appropriate OpenCV function. Finally a binary image is required in order to be able to count black and white pixels. In order to cut at this step as much of the data as possible that are not necessary for the nest stages of character recognition the following method is used. First a horizontal projection of the image is created. Example result is given on fig. 5 where the green line shows the position of the line of interest calculated as follows:

$$vli = avg + (width - avg) / 3 \qquad (1)$$

where *vli* is the distance of the line of interest from left image border, *width* is the width of image, *avg* is the arithmetic mean value of the count of white pixels on each row. After the position of the line of interest is determined, the peaks of the white pixels are found (the blue lines on fig. 5) and the image is cropped at these y positions.

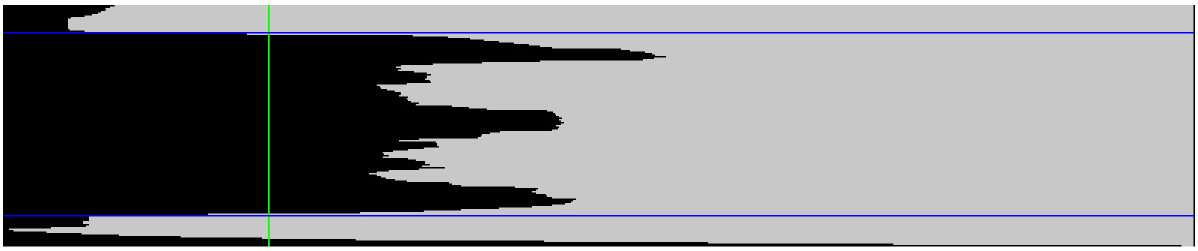


Fig. 5. Horizontal projection of the image after thresholding with all imporant lines marked

Fig. 6. The image after horizontal cropping

The resulting image (fig. 6) is now clear from unnecessary pixels above and below the number plate. The next task is to segment that image into smaller images comprising separated characters of the car number. Similar approach as the above is applied based on cropping the image using vertical projection and detection of a horizontal line of interest with position from top to bottom of the image calculated as follows:

$$hli = (height - avg) / 3 \qquad (2)$$

where *hli* is the distance of the line of interest from top image border, *height* is the height of the image, *avg* is the arithmetic mean value of the count of white pixels on each column. After the position of the line of interest is found, the start and the end of the peak values of the white pixels can be easily determined (the blue lines on fig. 7) and the lines for cutting are decided accordingly (the red lines on fig. 7). The segmenting of the car plate image using this approach provides good results for most of the cases – in the example on fig. 7 only the first and the last segments do not correspond to characters, but they are to be filtered at the next processing stages of classifying and syntax analysis.

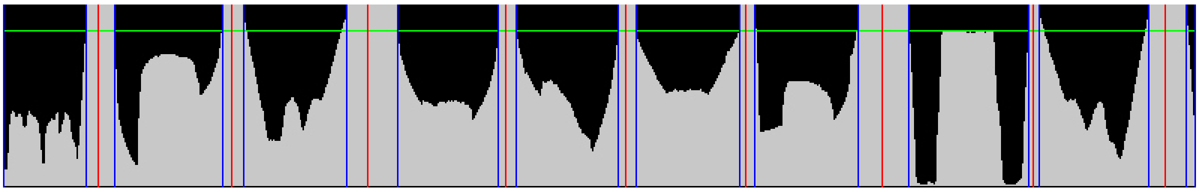


Fig. 7. Vertical projection of the image with all important lines marked

3.4 Classification

The classification of the segmented images of the car plate is based on neural network approach [10, 12, 13]. The neural network is first trained with sample images of the characters to be recognized and then the trained network is used for recognition.

Before the classification stage certain features from each characters image have to be extracted. The images should be normalized to the size of the images used for neural network training, for example downscaled to 20x30 pixels. If the image has different proportions, it has to be centered to a white 20x30 pixels image.

There are different methods for feature extraction that can be applied before the neural network recognition phase. Having in mind the recognition application is aimed at working on a mobile device with limited memory, the feature extraction should minimize the size of the training dataset. When working with large training datasets which are crucial for better results of the classification and mobile devices with limited heap size reducing the data for representation of the processed images is very important requirement for the selection of the appropriate feature extraction approach.

The straightforward presentation of 20x30 pixels image requires 600 pixels per character to be stored in memory. Although the pixels of the processed images are still too much for the standard limit of 8MB or 16MB heap size, a modified to 32MB heap size Android platform can be used in this case.

Another approach for feature extraction is used in the described ANPR system: it is based on extraction of descriptions of the images to be classified based on the analysis of simple binary textured patterns. An area of four pixels is selected with a set of all available pixel patterns for that area. From the set of patterns four are removed – four whites, four blacks and the two chesstable-like patterns. Thus a set of 12 patterns is formed and the frequency of these patterns in each of the processed images is calculated (fig. 8).

The presented pattern frequency is not enough description to differentiate between the characters and still dissimilar images can have similar pattern distribution. Therefore the image is further divided into four regions and the frequency of the patterns is counted separately for each region, thus more exact description of the analyzed image is derived. In this case using the described approach the letter ‘A’ for example will have different slopes from the left side and the right side and having an image with similar slopes, but placed to the wrong side, will not lead to an assumption that the image is an ‘A’. Using the described feature extraction only 48 values (12 patterns by 4 regions) have to be stored for each image thus greatly reducing the required memory.

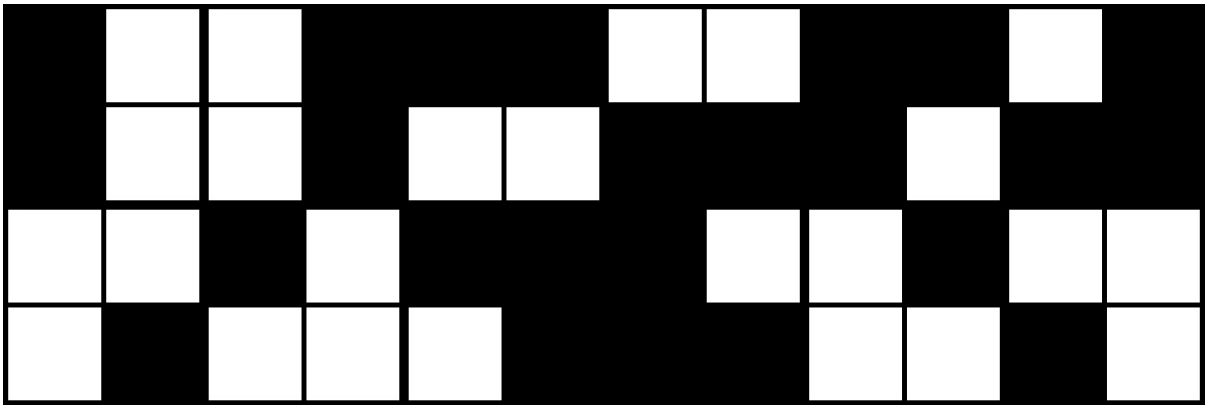


Fig. 8. The set of 12 patterns used for feature extraction

After the feature extraction stage is accomplished the classification of characters in the segmented images is based on the obtained results. For the classification a multi-layer perceptron neural network (MLP NN) is used which is a feed forward artificial neural network, utilizing a supervised learning method called backpropagation[14]. After initial training of the neural network once, the results for the network weights are saved and are loaded each time recognition takes place in the ANPR system. The functionality for the neural network classifier is based on OpenCV.

3.5 Syntax analysis

The result of the neural network classification of given image is a letter with a matching coefficient. Based on the results non-character artifacts are filtered. The recognized letter can also be passed to syntax analysis. Syntax analysis stage is built on top of the rules for the official number plate syntax, proposed by the laws of the country where the application is operated. If the recognized letter doesn't fit into the number plate syntax it is marked as unreliable. In this case the next match is used or the character is skipped assuming it's a non-character image.

The final stage is the option how to proceed with the ANPR processing: to repeat any of the above steps in case the results are not satisfactory, to repeat the whole process with different ROI, to repeat the recognition process with another image. The results of the processing and recognition can be shown, can be printed or can be sent as message if the application requires it.

4 Conclusion

The paper presented an automated number plate recognition system for Android mobile devices. The image processing and recognition stages of the described ANPR system were selected and implemented with the purposes of working on a limited resources device. The system is developed as a two module application and is implemented in Java and C languages.

The described ANPR application can be run on mobile devices that meet the following requirements: Android 2.1or later, CPU 500MHz or more, RAM 128MB or more, Dalvik Virtual Machine heap size 32MB or more, Storage for programs 10MB or more

The system can serve as part of parking applications, parking permission checker applications, mobile stations for special vehicle access mode software and any automated number plate recognition systems where system mobility is required.

Future extension of the system would include international syntax rules for number plate recognition, advanced plate selection rules including different shapes and different types of number plates, better feature extraction algorithm and well trained neural network. It would include better memory management of the heap size usage, so it could be decreased to 16MB or even 8MB.

References

[1] T. Mikonnen, *Programming Mobile Devices: An Introduction for Practitioners*, Wiley, 2007
[2] Wikipedia, *Computer Vision*, http://en.wikipedia.org/wiki/Computer_vision
[3] Wikipedia, *Automatic Number Plate Recognition*, http://en.wikipedia.org/wiki/Automatic_number_plate_recognition
[4] Wikipedia, *Mobile Operating Systems*, http://en.wikipedia.org/wiki/Mobile_operating_system
[5] Bill McCord, *OpenCV-Android*, http://github.com/billmccord/OpenCVAndroid
[6] Samuel Audet, *JavaCV*, http://code.google.com/p/javacv/
[7] Google, *Android Developers*, http://developer.android.com/
[8] The Eclipse Foundation, *Eclipse*, http://www.eclipse.org/
[9] Zixuan Wang, *OpenCV in Android*, http://www.stanford.edu/~zxwang/android_opencv.html
[10] O'Reilly, Gary Bradski and Adrian Kaehler, *Learning OpenCV*, 2008
[11] Ondrej Martinsky, *B.Sc Thesis, Algorithmic and Mathematical Principles of Automatic Number Plate Recognition System*, 2007
[12] Andrew Blais and David Mertz, *An introduction to neural networks*, http://www.ibm.com/developerworks/library/l-neural/
[13] Wikipedia, *Artificial Neural Network*, http://en.wikipedia.org/wiki/Artificial_neural_network/
[14] Wikipedia, *Multilayer perceptron*, http://en.wikipedia.org/wiki/Multilayer_perceptron

IVAN DANOV
Technical University of Sofia
Faculty of Computer Systems and Control
8 St. Kliment Ohridski blvd, Sofia 1000
BULGARIA
E-mail: me@danov.eu

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Simulation modeling of buyer's service in retail pharmacies

Mykhaylo Dorokhov
Coordinator: Prof. Dr.Sc. Ludmila Malyaretz

Abstract

In the article processes of organization of drugs and goods of medical purpose sale in a pharmacies are described and analyzed. On the basis of field research and their elaboration in Statistica environment types and parameters of statistic timeframes of customers' visits, duration of their service by assistants, probability of appearance of certain types of clients are determined. The existent possibilities of the organization of retail sale of medicine in drugstores are determined and structurized. A number of computer models of the system of mass service of customers in the drugstores by means of a package of discrete simulation modelling ExtendSim 7 LT are developed. However the possibility of taking into account the different number of counters at the drugstore, the number of consultants and doctors, possibility of appearance of priority clients and breaks are ensured. A number of calculative experiments are conducted, their results are processed and presented e.g. the appropriate quantity of staff according to characteristics of the concrete drugstore. The practical implementation assured that the model is adequate and it's ability to be used.

1 Introduction

Under the condition of the financial crises the competition, fight for the market in all spheres of commerce and produce arises. This also applicable to the Ukrainian pharmacy market e.g. the system of retail sale of goods of medical purpose. However an essential part of the improvement of pharmaceutical service in drugstores is the appropriate organization of selling the drugs, medicines and goods of medical purpose.

Selling customers medicine is a process of mass service with peculiar to it basic and specific elements which reflect the features of organization of selling pharmaceutical production [1-4].

Computer modeling of such a system will lead to the minimization of the time of consumers' service and queue, optimization the work of pharmaceuticals, and etc.

So, the aim of the research is the system of mass medicine service in the drugstores [5-7]. Also the development of the discrete simulation model in the computer modeling tool ExtendSim7 LT [8, 9], the analysis of the results and recommendation of medicine selling is supposed.

2 Identification of Statistical Parameters

In order to identify the initial parameters of the model the pilot surveillance were conducted. A number of details were fixed: the time when the customer enters the drugstore, the number of counters, the duration of service by the pharmaceuticals, the amount of the purchase.

Also into the account were taken the return of the customers to the subsequent return of the customers to the counter, the case when the customer left the queue because of its length and etc.

The analysis of initial parameters was conducted in the Statistica program.
For the time lines of the customers' entrance, the duration of staying in the queue, the amount of the purchase the hypothesis of normal, exponential and extreme dispersion were proved (some corresponding examples for similar distributions are shown on Fig.1, 2).

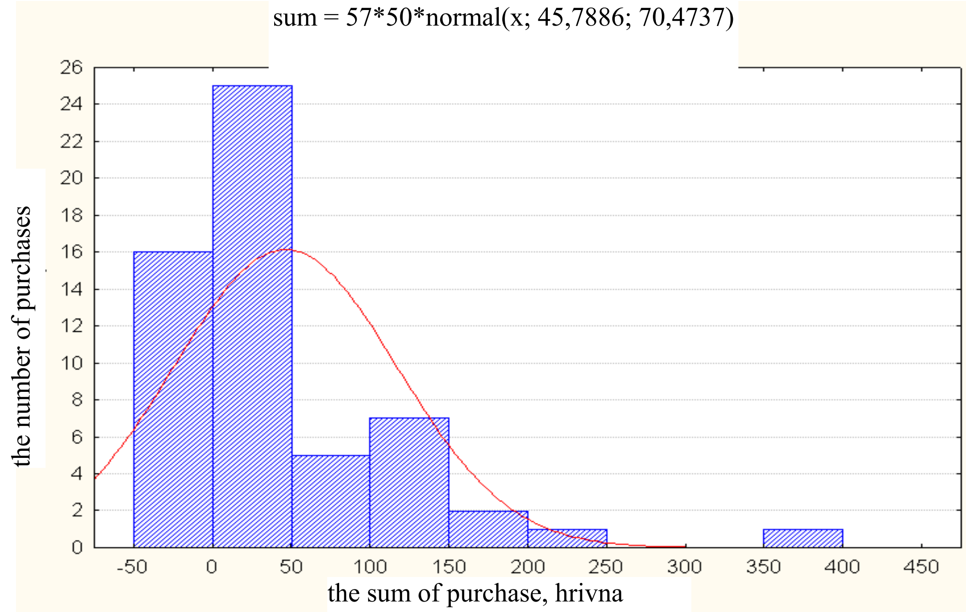


Fig. 1. The dispersion for amount of purchase (hrivna, similar to normal distribution).

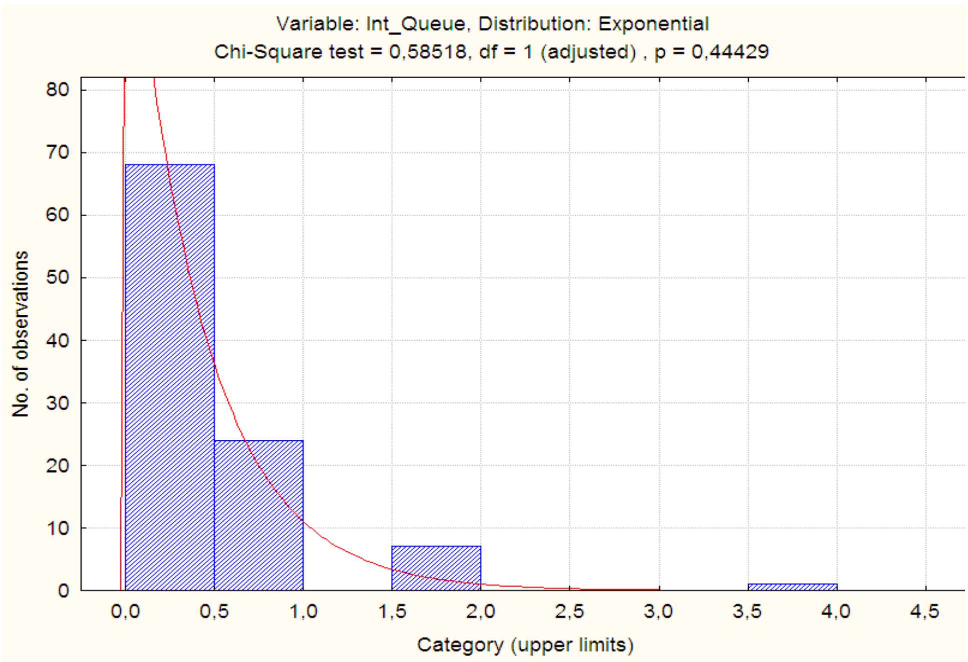


Fig. 2. The dispersion for duration of staying in queue (minutes, similar to exponential distribution).

3 Construction of Discrete Simulation Models

The first computer simulation model, shown on Fig.3, has been developed with the use of ExendSim7LT discrete simulation software.
This model includes following numbered items (blocks):

- №1 - input model parameters;

- №2 - customers' entry to the drugstore;
- №3 - joint queue;
- №4 till №6 – service windows (drug's sale);
- №7 - drugstore exit and statistics summary collection;
- №8 till №10 – time of service for each window;
- №11 till №14 - service times representation of the results for multi-time runs;
- №15 and №16 - represents the number of work items that have exceeded their waiting limit;
- №17 - analyses the queue;
- №18 - counts the work items that have left the drugstore at once.

Additional units represent other features of the model:

- №19 till №21 - count repeated service;
- №22 till №24 - the formation of the joint queue and clients' return to the common entry queue.

Blocks №25 till №27 reflect the repeated service of different customers at each cash desk.
This first, base model allows user to change the next entering parameters (in brackets measure units are written): time measure units (minutes); drugstores working time (hours); frequency of customer's appearance, its probability character depending on different distributions; quantity of servicing windows (one, two or three windows); duration of one customer servicing in some window (distribution, individual for each window, with corresponding servicing time).
In further models presence of a doctor in the drugstore, priority customers, different cash desks in case of epidemics and etc must be taken into the account.
It needs for the ultimate imitational modeling for different groups of drugstores in according to models that reflect the practical organization of selling drugs in its.

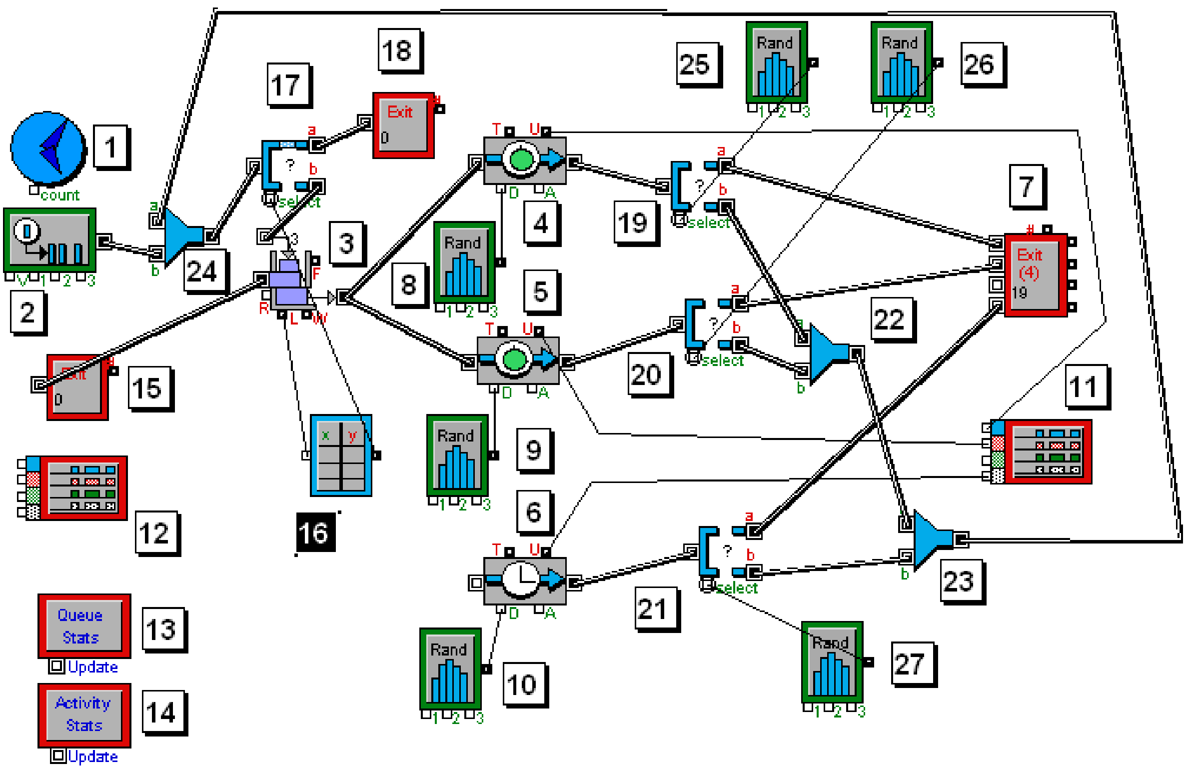


Fig. 3 – The model with three servicing devices, analyses of the length of the queue and the time of waiting, returning of the customers to the common entrance queue for repeated service.

Not losing a generality, we shall consider further a drugstore with two service windows. The first model assumed the common queue to all windows. A case of presence of separate queues is presented on Fig.4.

The client goes to a window with smaller queue. If queues are identical, the buyer goes in any of its. If the total number of buyers in both queues exceeds admissible, the new client leaves a drugstore. This model contains next additional blocks:

- №28, 30 and №32 – choice of clients way to smaller queue, where the number of clients is less;
- №29 and №31 - blocks for comparison of queues lengths;
- №33 and №34 – separate queues before first and second service windows;
- №35 – summary of common number of clients in both queues;
- №36 and №37 – blocks, which provide clients' leaving from drugstore if common queue length is unacceptable.

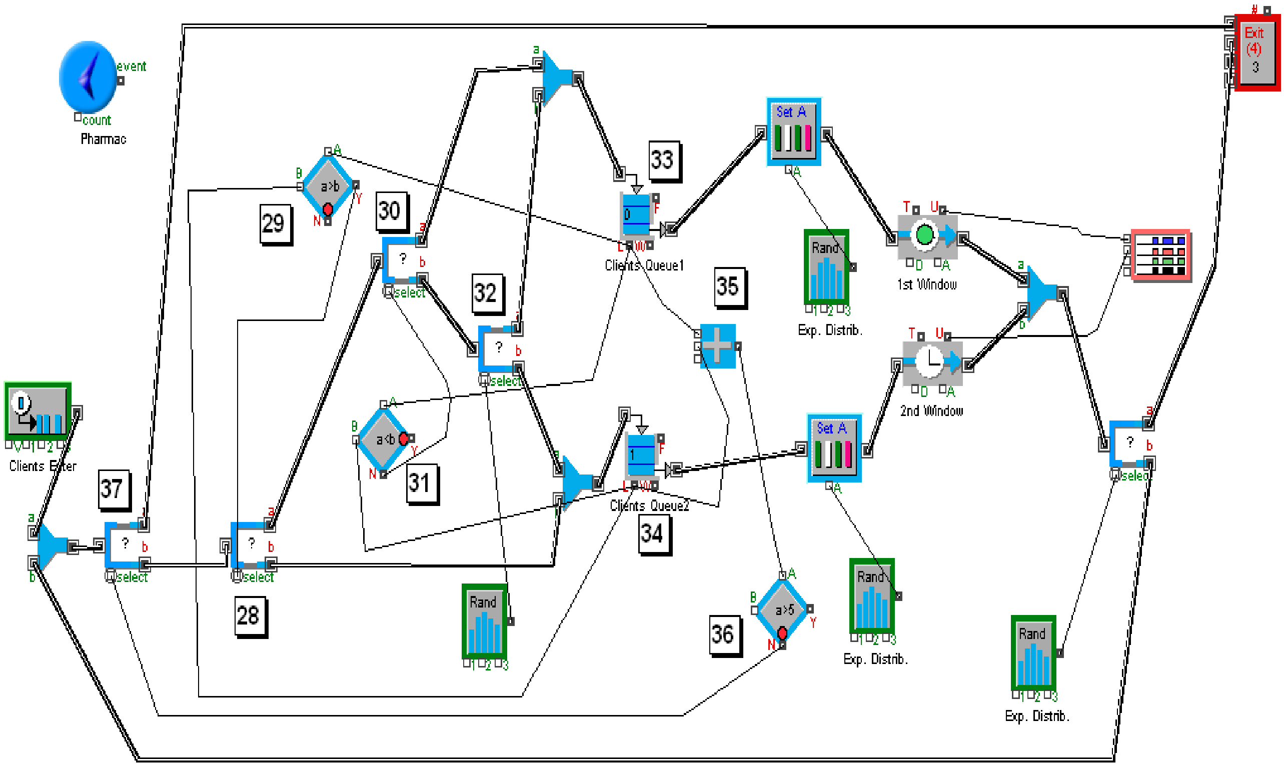


Fig. 4 – The model with two service windows and separate queues to them,

Next step for models is presented on Fig.5.

In this case the buyer after the choice of drugs in service window (at the pharmacist) pays its cost in a separate cash window, and then again comes back in the common turn for reception of the paid purchase at the pharmacist.

For this purposes corresponding group of block added to finish part of previous model:

- №38 and №39 - fixing (removal) of the attribute, which testifying (confirm) to already taken place service of the buyer in cash window (payment of drugs by him);
- №40 and №41 - check of conditions (if payment is needed then client go to cash window, else drugs are already taken by buyer and he go to drugstore exit;
- №42 - formation of the common queue from two service windows before cash window;
- №43 - queue before cash window;
- №44 and №45 – service time parameters for cash window;
- №46 - cash (pay) window;
- №47 - block for set to client attribute, which confirm his payment in cash window and the subsequent clients transition to beginning of model (common input and formation of queues for both service windows). transition to beginning of model (common input and formation of queues for both service windows).

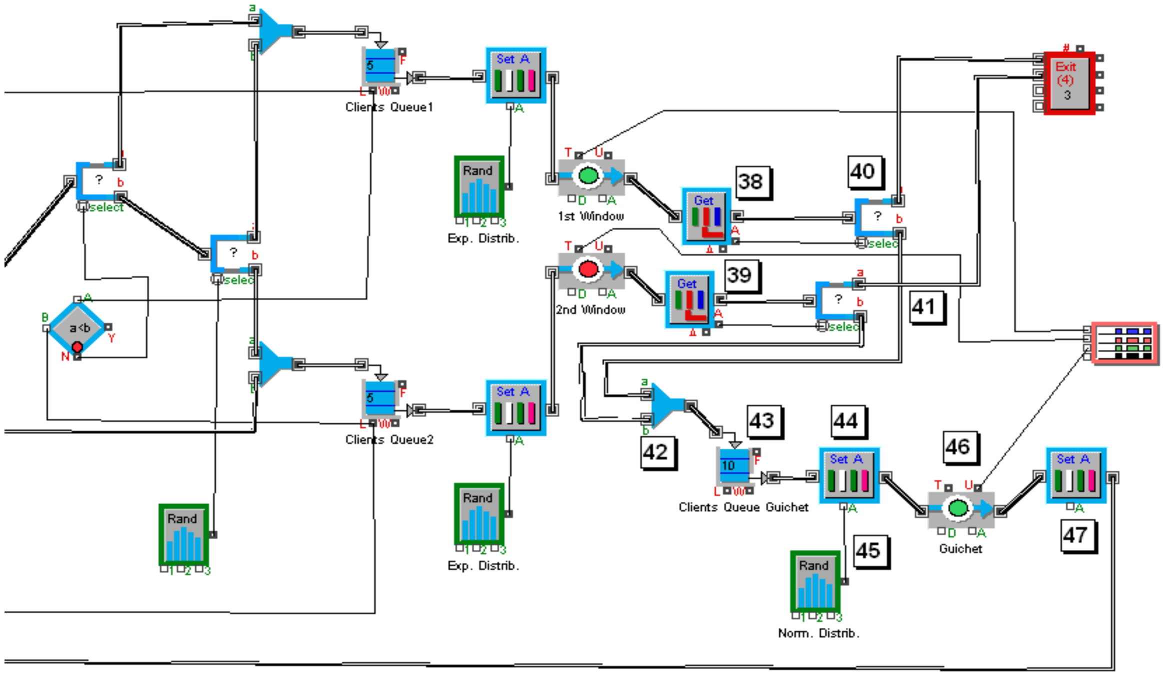


Fig. 5 – The model with two service windows and separate cash (pay) window.

4 Results of Simulation

Finally, let's consider results of numerical simulation for the model showed on Fig.3. Entrance parameters for distributions of customer's appearance and one client servicing terms in each window are shown in Table 1.

Table 1. Customers Appearance And Servicing Distributions, Min

Time	Customer entrance	Servicing time		
		Window №1	Window №2	Window №3
Minimum	1	2	3	5
Most probable	2	10	6	8
Maximum	5	12	9	12

The imitation took place for the cases of working only one, two or all three windows in the same time. Modeling results, in particular queue characteristics – length and waiting time, windows loading for 50 model imitations with 95% confidential interval are shown in Table 2.

Results of some imitations may be received in the graphical view using the special model blocks.

By using modeling results analysis we can make conclusion that drugs selling in three windows is the most optimal (also from the point of view of customers servicing quality and pharmacists loading). If this number of windows, total queue to all of them on the average is absent and doesn't exceed 4 customers.

Waiting time in the queue is on the average 1.5 minutes and doesn't exceed 7 minutes. Such a service is entirely comprehensible for the customers.

From the other side, the pharmacists are loaded by the work for 90-95% of working hours, what is quite high, practically maximal parameter of their use efficiency. Though, reduction of windows quantity would provide very long queues and big waiting time for customers. Windows quantity increase would provide stagnation of drugstore stuff.

Table 2. Modelling Results with 95% Reliability (Duration of Simulation Time is 600 Minutes)

Servicing parameters	Only window №1 works	Windows №1 and №2 works	All three windows works
Average queue length	76	24	0
Maximal queue length	151	49	4
Average waiting time	199	64	1,4
Maximal waiting time	400	131	7,5
Total customers serviced	71	185	227
Serviced in window №1	71	81	69
Loading of window №1	100%	100%	94%
Serviced in window №2	-	104	94
Loading of window №2	-	99%	89%
Serviced in window №3	-	-	64
Loading of window №3	-	-	88%

We have developed described models in direction of taking other factors. For example, the next situations were provided, when a long queue or big waiting time the customer leaves the drugstore. Also repeated turning of customer after the first purchase has been provided.

Extraordinary occasion describes existence of consulting services in the drugstore, for example when some visitors are being consulted by the doctor.

We have also researched the abilities of optimal windows quantity defining using comparing of its functioning costs and potential losses because of customers exit in case of long queue and big waiting time. We have developed described models in direction of taking other factors. For example, the next situations were provided, when a long queue or big waiting time the customer leaves the drugstore. Also repeated turning of customer after the first purchase has been provided.

Extraordinary occasion describes existence of consulting services in the drugstore, for example when some visitors are being consulted by the doctor.

We have also researched the abilities of optimal windows quantity defining using comparing of its functioning costs and potential losses because of customers exit in case of long queue and big waiting time.

The developed models have sufficient practical value. They allow to define concrete numerical parameters of service: a waiting time of buyers in turns, length of turns, a ratio of of a service window's charges and financial losses of leaving a part of buyers.

These models can be used for improvement of practical work of a drugstore, rationalization of loading of its personnel, acceleration of service for medical product's buyers. In particular, it is possible to choose such scheme of service (a number of service windows, cash departments) at which the waiting time for clients will be minimal. In case of certain epidemiological situations it is possible to determine expediency of allocation of separate service windows for sale of separate special drugs.

Also it is possible to change a number of service windows depending on times of day, days of week (for which the distribution of buyer's occurrence is various).

5 Conclusions

Thus, productive approach to imitating modeling for queues in drugstores and customers services have been developed and presented. Developed computer models in ExtendSim 7 LT environment allows get important characteristics of pharmaceutical services using statistical data and calculation experiment – existence of queues, waiting time, pharmacists loading etc.

As a result it is possible to improve the customer's service and to optimize functioning of the drugstores. After the calculative experiments it appeared that the developed models give a chance to improve the quality of service and at the same time to optimize the work of pharmacies on the basis of computer modeling and the analysis of different ways of selling medicine.

References

[1] Z. Mnushko, I. Sofronova. Analysis of economical, management and marketing research directions in pharmacy. *News of the Pharmacy,* 4: 53–58, 2004.

[2] B. Hromovyk, S. Mokryanin. Problematical questions of drugs consumers logistical servicing. *Pharmaceutical Journal*, 5: 14-18, 2007.

[3] I. Pestun, Z. Mnushko, V. Presnyakova. Marketing informational support of decision-making process in pharmacy. *Pharmaceutical Journal*, 1: 9-14, 2007.

[4] V. Tolochko, *Management of pharmacy*, NPhaU publishers, Kharkiv, 2004.

[5] L. Dorokhova. Model in management. *Pharmaceutical encyclopedia,* Morion, Kyiv, 2005.

[6] S. Dovhan, *Mathematical modeling in marketing,* NGU, Dnepropetrovsk, 2002.

[7] R. Shennone, *Imitating modeling of systems – science and art,* NGU, Moscow, 1978.

[8] E. Kopytov, L. Greenglaz, A. Muravyov, E. Puzinkevich. Modeling of two strategies in inventory control system, *Computer Modeling and New Technologies,* 1: 21-30, 2007.

[9] J. Moore, L. Weatherford, *Decision modeling with Microsoft Excel,* Prentice Hall, New Jersey, 2004.

MYKHAYLO DOROKHOV
Kharkiv National University of Economics
Faculty of Economics Informatics, Chair of Informations Systems
Prospekt Lenina 9-A, Kharkiv, 61001
UKRAINE
E-mail: mikedorohov@meta.ua

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

A comparative study between AES and BlowFish encryption algorithms

Mihai Fieraru, Madalina Iordache
Teacher Coordinator: Gabriela Moise

Abstract

The encryption represents the process of confidential information's codification so they can not be read by unauthorized persons. This vast area has been investigated by many scientists, the challenge of the researches being to discover a more reliable encryption algorithm. An ideal encryption algorithm is an algorithm that can withstand any attack by unauthorized persons and relieved that can be implemented at lower cost, but this represents a challenge for the researchers in this area.

In this paper there are presented two of the most used information encryption algorithms: AES and Blowfish. The algorithms were implemented in C programming language and were tested with different types of input data, and a comparison is presented taking into account the algorithms characteristics: response time, complexity and security offered.

1 Introduction

Securing information transmission tends to be an objective that becomes more difficult to achieve in a world where information is transmitted through middle of communication being accessible to anyone. In ancient times, people have tried to figure data and transmit to recipients through a secure means of communication. Nowadays, the main organizations and governments have implemented systems that enable them secure encryption of data according to their destination: the data "secret" is encrypted using standard keys of 128, 192 and 256 bits and data "top secret" are encrypted using keys of 448 bits (even more) [11].
While encryption algorithms and information security methods are developing, there are more and more people who are looking for weaknesses in computer systems in order to gain access to secret information. As can be seen in Fig.1, all encryption systems, respect a certain number of steps: encryption, transmission and decryption information [2].

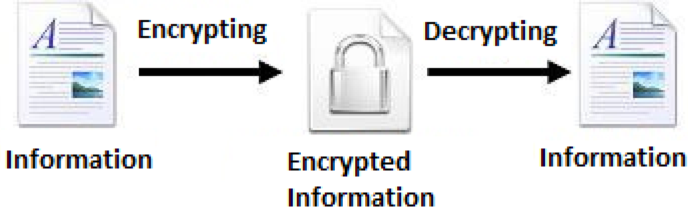


Fig. 1. Steps to protect information

Sometimes, if the situation requires, some information may be encrypted using several times the encryption procedure: for instance, TRIPLE Blowfish can encrypt information in 1334 bits, this process is made by repeating three times the encryption operation, using each time a 448 bits key. AES and Blowfish algorithms are among the most used in this domain because they are considered to be impenetrable because of the characteristics of the keys used by them [10].
Although these two algorithms were tested by different experts in encryption, until now there have not been made public results showing that two algorithms were "broken".
The comparative study done on the two algorithms take into account the next criteria: the efficiency of the encryption, time associated with an encryption operation and the implementation mode.
Cryptanalysis refers to the study of ciphers, ciphertext, or cryptosystems (that is, to secret code systems) with a view to finding weaknesses in them that will permit retrieval of the plaintext from the ciphertext, without necessarily knowing the key or the algorithm. This is known as breaking the cipher, ciphertext, or cryptosystem.
The scientific literature contains various attempts of penetration the security of a system that uses two algorithms, but the results were inconclusive because of 8 or 16 bits keys used by algorithms, which allows the access in the system of an unauthorized person that uses one of the two methods described above [1].

2 The encryption of information

The encryption is the process of encoding information so that it can be understood only by authorized persons.
Domeniul One of the most common methods of obtaining encrypted information is cryptanalysis, The cryptanalisys tehcniques can be brute force, linear, diferential, algebraic cryptanalysis techniques. This is a process that refers to "circumvent" security mechanisms and not exploiting an existing weakness in computer system because of more advanced technology.
This atatcks' techniques method can be more efficient than other methods of obtaining encrypted information because is watching human weakness and not weaknesses of the computer system, which often are extremely advanced workstations designed to not allow unauthorized access.
A method of cryptanalyasis is the brute force attack. This technique consists in, analyzing all possible combinations of bits of the keys to discover a key. The advantage of this method is that it can be very efficient in case of using a weak key in terms of security [9].
The encryption algorithms can be divided into symmetric and asymmetric algorithms.
Symmetric algorithms, or private, are the algorithms that are using the same key for both encryption and decryption of information. Because of this, the key used must be kept private. This key is used both by the system who encrypt the information and by the one who decipher it. For these keys to be harder to guess are used various systems who allow the generation of large dimensions keys, sometimes 448 bits random.
Another category of algorithms are asymmetrical algorithms, which are using different keys for encrypting and decrypting information. These keys are generated based on complex mathematical formulas which are performing operations with very large prime numbers. These keys are generated automatically, the public key being available to anyone and the private key being kept secret.
The difference between these two types of algorithms is working time. According to a study published by Stanford University, the asymmetric algorithms are 1000 times slower than the symmetric ones. The analyzed algorithms, AES and Blowfish are symmetric algorithms, that means they are using the same set of keys for both encryption and decryption information [8].

3. AES vs. Blowfish

This study aims to draw attention to the differences and similarities between the two algorithms take into account. In this case the following matters concerning the security offered by them and the necessary working time for obtaining a complete cryptographic operation.
The criteria on which the two algorithms are analyzed is represented by the security offered by the algorithm, the deployment costs and use and the flexibility that these algorithms give proof of their usability in various applications.
Advanced Encryption Standard or AES is an algorithm known as the Rijndael, standardized for use over the blocks, used internationally by all those who want an enhanced security of information such as the U.S. Government.[5]
Key length used by this algorithm takes standard fixed values:

- 128-bit;
- 192-bit;
- 256-bit.

Depending on the key, the algorithm will use a certain number of iterations:

The Used Key	Iteration Number	Level of security
128	10	Medium
192	12	Medium
256	14	Maximum

Table 1. Number of iterations performed by the algorithm for different types of keys

This algorithm is impossible to break because of the large number of iterations and because the key can be spitted into multiple parts that can be added to any block of information that we want to use.Also if we use a small keywe can multiply that key and after that we can add it to our blocks to encrypt the information.
Algorithm is much faster than its predecessors, DES and IDEA, who was studied by experts because of the fact that it is an open source. Encryption is performed in 16 rounds, each round being made in the assembly of words and mathematical operations based on previously calculated sub key encryption [6].

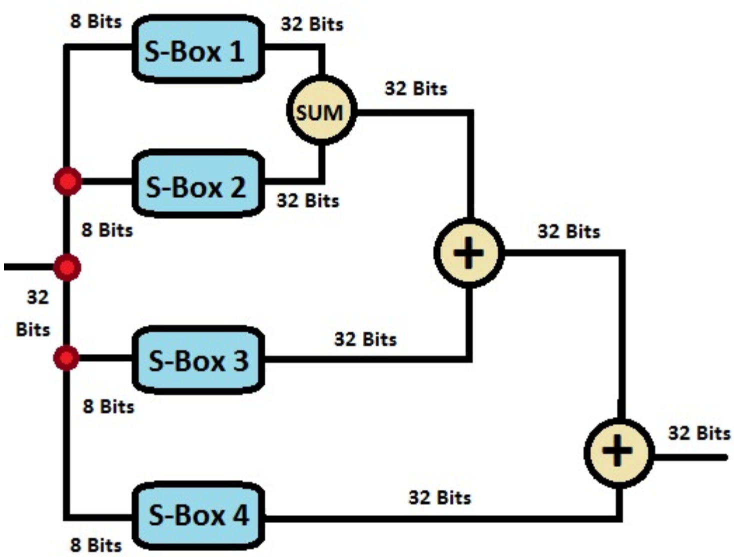


Fig. 2. The steps followed by two blocks of data encryption algorithm[12]

Figure 2 shows the mathematical operations sketch performed to encrypt data and sharing information in blocks of 8 bits in a 32-bit processor. On systems with processors running 64-bit input data to be encrypted data is "divided" into 32-bit blocks as it can be observed in the diagram presented in Fig. 3

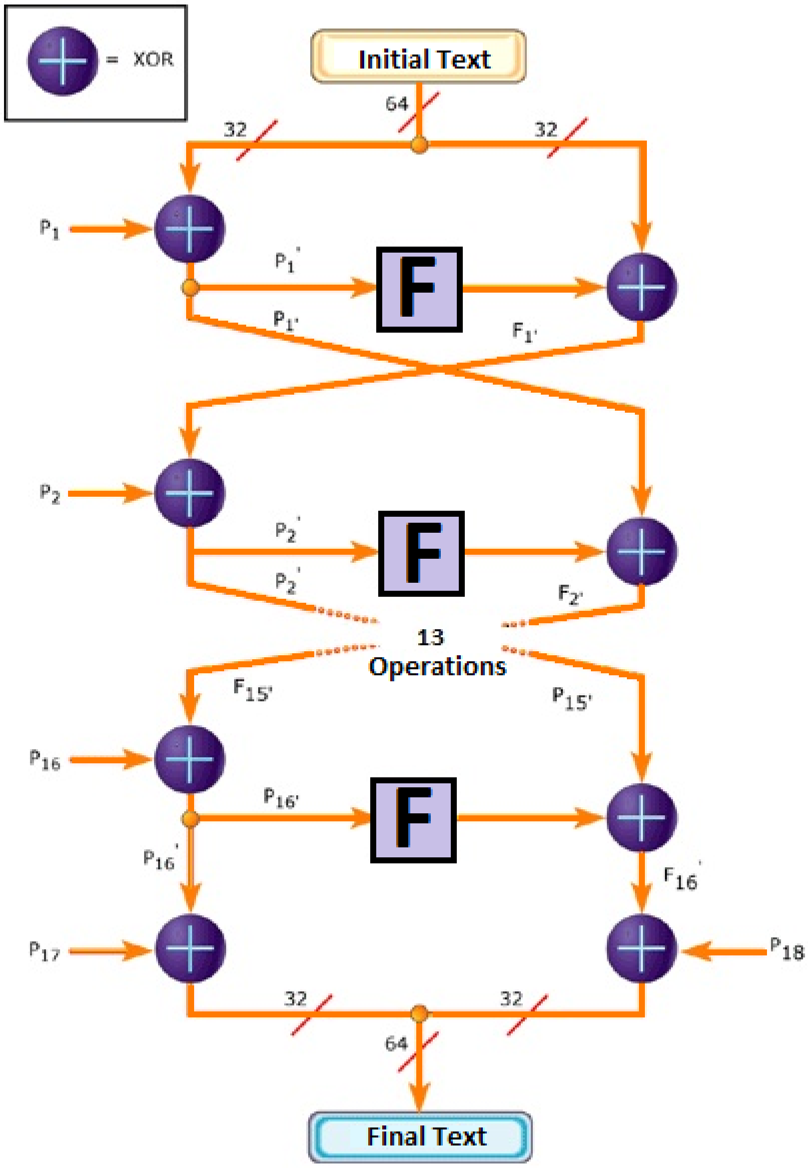


Fig. 3 Steps followed by the algorithm in the case of a 64-bit processor[12]

In terms of how to test these algorithms, they have been implemented using C programming language. As input data for testing the efficiency and security of a password algorithm was used whose number varies depending on the key characters are used. Testing was performed on systems with processors running 32-bit [3].

To showed more concrete concept of algorithm I realized you apply encryption key based on its own random: in the input file will be written on the front line of the alphabet, numbers and characters most used (eg +, - ,*,=, etc.) and on the second line of the file will be written with all the random characters who want some (eg a = +, B = $, + = 9). The characters were randomly positioned because it is more difficult to decrypt a function having no clear or well defined characters that would lead to discovery.

The application has the following steps: the sender can transmit the message as desired, encrypted or unencrypted, if he wants the message to be encrypted wanting to hide certain information he needs to

write and then it will encrypt encrypted message will be sent to the transmitter. If the message is intercepted by an unauthorized person will see only an array of characters with no logic to decrypt the message and will have to guess each character corresponds to point it almost impossible or would require too much work time for the message not to have validity or meaning. Once the message to the recipient will see it also as a person who fraudulently intercepted message, a sequence of characters that would have virtually no logic but the transmitter is in the application made by us to another file that has the chosen first-line characters as digital platforms and on the second line of characters in their natural form, with this file decryption will be achieved.
In the application is implemented by the authors and the characters used for encryption can be changed or can add elements and you can make each user's own code that only you know the sender and receiver, but must be careful when we transmitter and receiver have the same code because if they did not like the application could not "translate" the correct message.

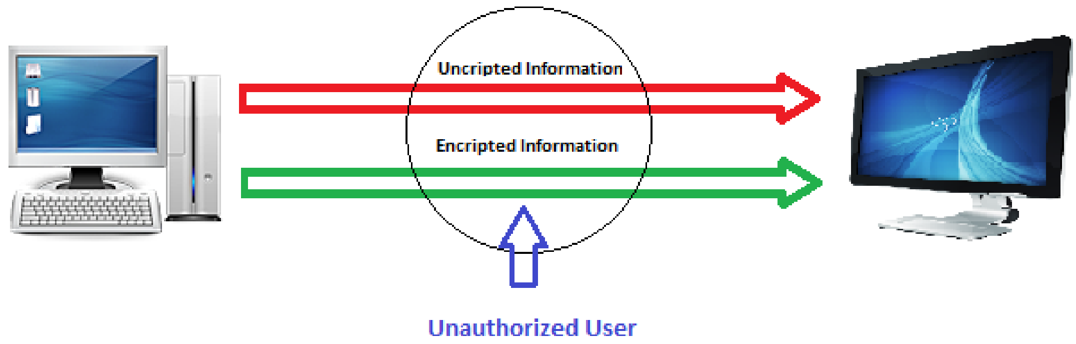


Fig.4 Access the information transmitted through a communication sistem

In Fig. 4 we can see that an unauthorized user can acces the information that is transmitted throught a communication chanel. The uncripted information can be visible to any unauthorized user and can be captured and retransmitted to destion .
When the information transmitted is cripted if any unauthorized user capture it, he can’t read the information because is encoded.
Experimental data used in this article were taken from various studies conducted by many specialists in cryptography and data security. They analyzed the many factors of great significance in implementing the two algorithms.

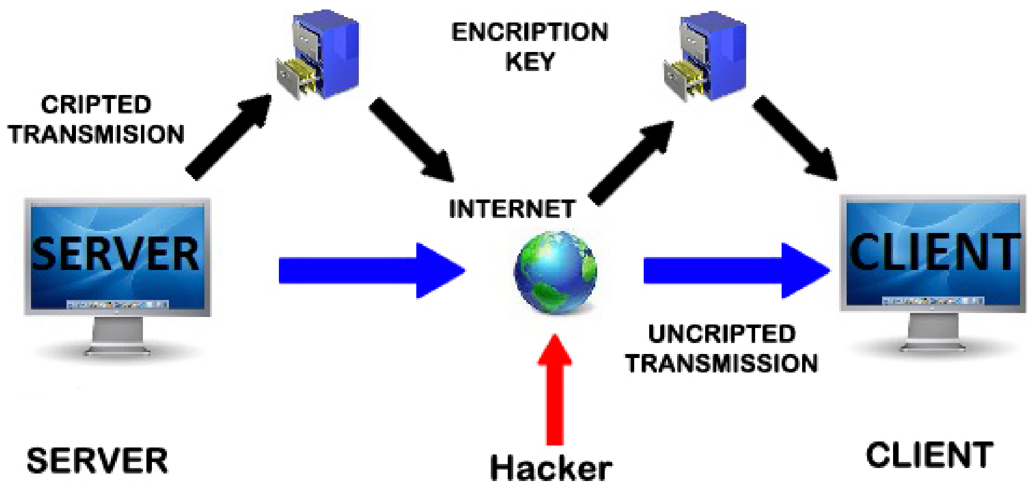


Fig.4-Our propossed application scheme

In Fig. 4 we can see the scheme of our proposed application for encryption and decryption of information.
With black lines we can see the route travelled by information when it will be encrypted. When the information is encrypted, the server it uses an algorithm with a key to encrypt the information that it will be sent to client over an unsecured transmission route.
As we see when the client receives the information, this is decrypted using the same algorithm with the same key that was used to encrypt .
With blue lines we can see the information that is transmitted over an unsecured network to the client. When this information is captured by a hacker he can read all the information without any problem.
With this application we want to demonstrate that any information that is tranmitted over an unsecured network it is available for any person to be readed.
The black line is represented by information sent by the server to the client which is encrypted and encryption keys are represented by a cabinet with drawers which can be found within files.
Red line represents the path through which a hacker can obtain information on the Internet and if it is not secure (encrypted) it may use the information in transmissions from the server to the client.

3 Results

They compare two algorithms, AES and Blowfish are symmetric key algorithms that use the same key to encrypt and decrypt information. These keys are obtained prior to commencement of operations as mathematical encryption is performed using sub-faction key information.
In terms of RAM which they operate is found, that Blowfish algorithm requires only 4 Kb of memory that uses AES as opposed to at least 16 Mb (according to official data available on the website www.aesdirect.gov) which presents a disadvantage for the AES algorithm.
Note that running the two algorithms to compile in C using Microsoft Visual C + + on a Pentium 2 processor with a speed of 2.1 GHz with Windows XP as Blowfish algorithm has the lowest response time [4].

ALGORITHM	Processed Mb	Response time (s)	Mb / Second
Blowfish	256	3.976	64.386
AES 128 biti	256	4.196	61.010
AES 192 biti	256	4.817	53.145
AES 256 biti	256	5.308	48.229

Table 2 Characteristics of the two algorithms analyzed

The table above shows that the Blowfish algorithm is the fastest, compared to AES, whose response time increases with the length of encryption keys. Precisely because of the speed of encryption / decryption of AES algorithm it was adopted by the U.S. government to encrypt some documents private.
Because the two algorithms can be compared as accurately demonstrations were conducted on two other workstations with different characteristics from this comparison is watching his time increasing the amount of data by getting random values of bytes.

Input Data (Bytes)	AES Algorithm	Blowfish Algorithm
20 527	39	19
36 002	74	35
59 852	125	58
137 325	285	136
166 364	355	162
191 383	378	176
232 398	460	219

Table 3. Necessary time to process the information

From Table 3 we see that on a workstation equipped with a weaker processor (266 MHz) Blowfish algorithm will be faster than the average working AES. Blowfish time will be taken in case of 115 seconds, unlike the AES has average working time of 245.14 seconds.
From these comparisons show that the Blowfish algorithm can perform a large number of mathematical operations, aimed at encrypting information in an extremely short time due to its structure levels.
In terms of encryption and transmission of information in a computer network can be seen that AES algorithm will perform the necessary operations in a shorter time than Blowfish. (According to one study published on www.nsa.gov)[7].
The time difference, the algorithm running, the two algorithms vary enough that starting with a difference of 2 seconds at 20 527 bits data and reaching a difference of 214 seconds at 232398 bits of information.

Algorithm	Transport mode	Cycles	Time
AES 128 biti	IP	65629	25.2 μs
Blowfish 128 biti	IP	112603	43.3μs
AES 128 biti	Tunnel	66620	25.6μs
Blowfish 128 biti	Tunnel	116392	44.7μs

Table 4 Time to encryption and data transmission and the number of cycles performed.

4. Conclusions

Of those presented in this document can be concluded that, in terms of security of the two algorithms are virtually impenetrable because of the mathematical operations performed for data encryption, and in terms of implementation and cost can be considered superior Blowfish algorithm AES algorithm.
Due to very small memory required for running the Blowfish algorithm, it has been implemented on mobile communication devices existing data security and encryption of data transmissions that use insecure communication channels.
As the comparation study show, if it used BlowFish or AES, the information that any encryption algorithm.
If we need a high degree off security we will use a strong key: 512 bit or 1024 bit that can offer us a high level of security because, theoretical, are impossible to break.

All the major cryptography experts have concluded that BlowFish with a 448 bits key and AES with a 256 bits key are impossible to break because of the large number of operations with big randmon numbers that are made to encrypt the information.
To highlight how a message can be taken by an unauthorized person in an unsafe environment and this application demonstrates our appliances once showed all important if we want to send encrypted messages important information.
The application demonstrates how a message can be sent encrypted, decrypted, and if stolen by another person can not be understood or a decryption time would be great or even impossible for some codes.

Our program can be considered as similar to symmetric algoritms because it uses a table to encrypt and to decript information and that table must be kept private.
If the table is obtained by any unauthorized person the algorithm is very easy to decrypt.

References

[1] Ciprian Racuciu, *Criptografie si securitatea informatiei*, Editura Universitatii Titu Maiorescu, Bucuresti 2008.
[2] Niels Ferguson, Bruce Schneier, *Cryptography Engineering*, Tadayoshi Kohno.
[3] William Stallings. *Cryptography and Network Security*, 4th edition. Prentice Hall, 2005.
[4] John Kelsey, Stefan Lucks, Bruce Schneie, Mike Stay, David Wagner, Doug Whiting, *„Improved Cryptanalysis of Rijndael, Fast Software Encryption Workshop,* Springer-Verlag, 2000.
[5] G. Agnew, *Random sources for cryptographic systems, Advances* in Cryptology, EUROCRYPT ’87 (LNCS 304), 77–81, 1988.
[6] N. Ahmed, K. R. Rao, Orthogonal *Transforms for Digital Signal Processing. Springer-Verlag*, New York, 1975.
[7] Ion Angheloiu, V. V. Patriciu, *Securitatea şi Protecţia Informaţiei în Sistemele Electronice de Calcul*, Ed. Militară, Bucureşti, 1986.
[8] Ion Angheloiu, E.Gyorfi, *Elemente de teoria transmiterei informaţiei*, Ed. Academia Militară, Bucureşti, 1976.
[9] U. Baum, S. Blackburn, *Clock-controlled pseudorandom generators on finite groups*, B. Preneel, editor, Fast Software Encryption, Second International Workshop (LNCS 1008), 6–21, Springer-Verlag, 1995.
[10] E. Biham, A. Shamir, *Differential Cryptanalysis of DES-like Cryptosystems, Advances in Cryptology – CRIPTO '90 Proceedings*, Springer-Verlag, 1991.
[11] ***, www.schneier.com, accessed on 25.02.2011
[12] The scheme is made according to existing standard model site: www.schneier.com / blowfish. html accessed on 25.02.2011

FIERARU MIHAI
Petroleum Gas University of Ploiesti
ROMANIA
E-mail: mihai.vest@yahoo.com

IORDACHE MADALINA
Petroleum Gas University of Ploiesti
ROMANIA
E-mail: lilium_martagun@yahoo.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Motion detection with voice alarm

Magdalena Filcheva
Teacher Coordinator: Milena Lazarova

Abstract

The paper presents a software system for motion detection with a voice alarm. It allows movement in specific zones to be detected automatically without the need for operator intervention and spoken voice warnings – alarms, to be issued. The aim is voice notification for a specified monitoring area for a specific object to be promptly issued once an unauthorized motion is detected. The system allows usage of user defined monitoring areas with different forms. The system allows specific phrases for certain areas to be set and different generated voices to be used. An important ability of the system is saving image frames in case the user was not present at the time the alarm is activated. Thus the frame can be monitored later or deleted.

1 Introduction

Video surveillance is a very difficult task for humans [1]. The video streams require a higher level of visual attention than most everyday tasks. One of the conclusions of a study by U.S. National Institute of Justice is that human control would not maintain effective security system even the person is dedicated and conscientious. Most of the security systems are more reliable than a human vision. After 20 minutes of video surveillance, the focus of most people is down. Video surveillance is boring and mesmerizing. There is no intellectually engaging, such as watching television program. Permanent viewing screen makes people feel tired. The little inattention can miss the potential offenders. That makes a very serious responsibility of the person. The System that will play the same role will be very helpful. There will not be need person who will monitor the camera. The software will detect a movement. The need for such systems is very high because they are observers all round the clock with a maximum concentration without fatigue.
It should be noted that the camera takes the picture much different from human eye. The human eye provides three-dimensional image in the brain, and the camera two-dimensional image for that it uses shadows and colours to create a sense of depth. The location of camera gives extra sense of depth, but even with that we can't achieve the same effect as the human eye [2]. Eye automatically adjusts itself for changes of light and shadow. The analogy camera can't do the same. Achievements in technologies of production of lenses CCD Sensors (Charged-Couple Device) and DSP processors (Digital Signal Processing) contribute a significant improvement in pictures for video surveillance systems.
System with voice alarm is system that quickly and easily play voice message that warns the person or group of people. The voice alarm is the easiest way for warning because isn't difficult for understanding. The aim is to attract people's attention. Providing voice alarm with specific characteristics of a particular place are future functions for all systems. Depends on the location of person the warning can be different.

Video Motion Detection is a tool for detecting a movement. It can be found in a frame by image analysis and differences between series of images. The function can be integrated in a network video products and video management software. They are many ways for implementation: motion detector in digital video recording system, motion detector in the network video systems or motion detector in software systems.
In motion detector in digital video recording system cameras are connected to DVR, which detects motion in each video stream. The disadvantage is that video motion detector is loading process. Using all channels can have a negative impact on system. [3]
The motion detector in the network video systems it is very specific because detection is processed in the network camera or in video server. This minimizes the risk of overloading the recorder system. Then it doesn’t send any video to the operator or recording system. If there any detected movements then system may series of actions: keeps records previously, during and after the event, sent an e-mail or SMS to the recipients, activate the light, monitoring the person, turn off or turn on alarm, turn off or turn on door, etc.
Video motion detectors can be placed in software for video management in that way increasing the functionality of network cameras. Software system allows a creation of a system based on the user's wishes. The software systems are quick and easy way for motion detection. It researches the picture sent from camera (USB camera or any other video device) in certain algorithms. When it detected a movement the program can activate several commands, including set an alarm for warning. Software can be easily upgrade by add a new functionality or improved algorithms. Good software is one that has algorithms that detect a characteristic movement in complex environments. Algorithm is effective when he discovers the characteristic movements of different real environments, such environments where the light is changing, when it rains, environments with fountains, shaking or vibration of the camera by wind, rain, snow, etc.
The paper presents a software system for motion detection with a voice alarm. It allows automatically without the need for operator intervention movement in specific zones to be detected and spoken voice warnings – alarms, to be issued.

2 System requirements

The purpose of the system for motion detection with voice alarm is without any observer to take notice when a movement is detected. The aim is voice notification for a specified area for a specific object to be promptly issued once an unauthorized motion is detected. The system allows usage of user defined monitoring areas with different forms. System alert alarm is started automatically only when motion is detected in these areas. The system allows specific phrases for certain areas to be set and different generated voices to be used. An important ability of the system is saving image frames in case the user was not present at the time the alarm is activated. Thus the frame can be monitored later or deleted. When the computer has more than one USB camera, the system allows user to select different cameras.
There are some requirements for proper operation of the system:

- Position of the camera must be on correct place (Fig.1). In other case the system will activate a false alarm (*Fig.2*).

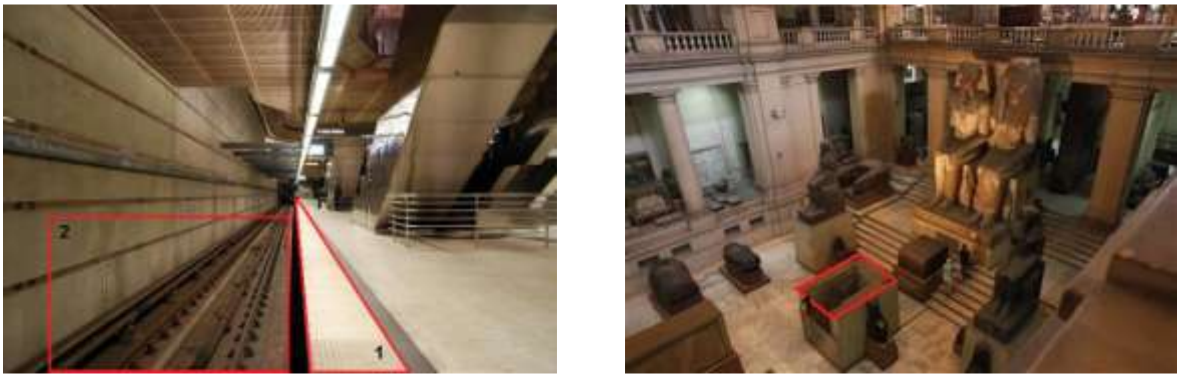


Fig.1.Example for right camera position

Fig.2.Example for wrong camera position

- User has to adjust the sensitivity of the system, to avoid false alarms.

In some cases the light or the other conditions has to be considered for adjusting the sensitivity. There may be a lot of noise form the camera and this will cause false alarms too. It can be eliminated with decreasing the sensitivity.

- Another requirement is the proper setting of areas. It is related with the position of the camera.

The user can choose a value, representing the percentage of the movement in the areas. With small number even noise and false alarms are detected but large numbers can pass an alarm. Therefore the adjustments have to be done by the administrator of the system. He will know all the aspects of the problems and the solutions for them.

3 System architecture

Fig. 3 represents the architecture of the system. It has to main modules “Motion detection” and “Voice alert”. “Motion detection” gets frames as an input from video camera stream or from the saved frames in the memory. It processes the frames, finds the differences and determined if there is a movement in the defined areas. All this process is passed to the user interface. If there is detected motion in one of the areas, the parameters are passed to the “Voice alert” module and they are saved in the history. “Voice alert” module has to find out what is the defined voice alarm for the area. It can be a different any time. Then it passes synthesized voice alarm to the speakers.

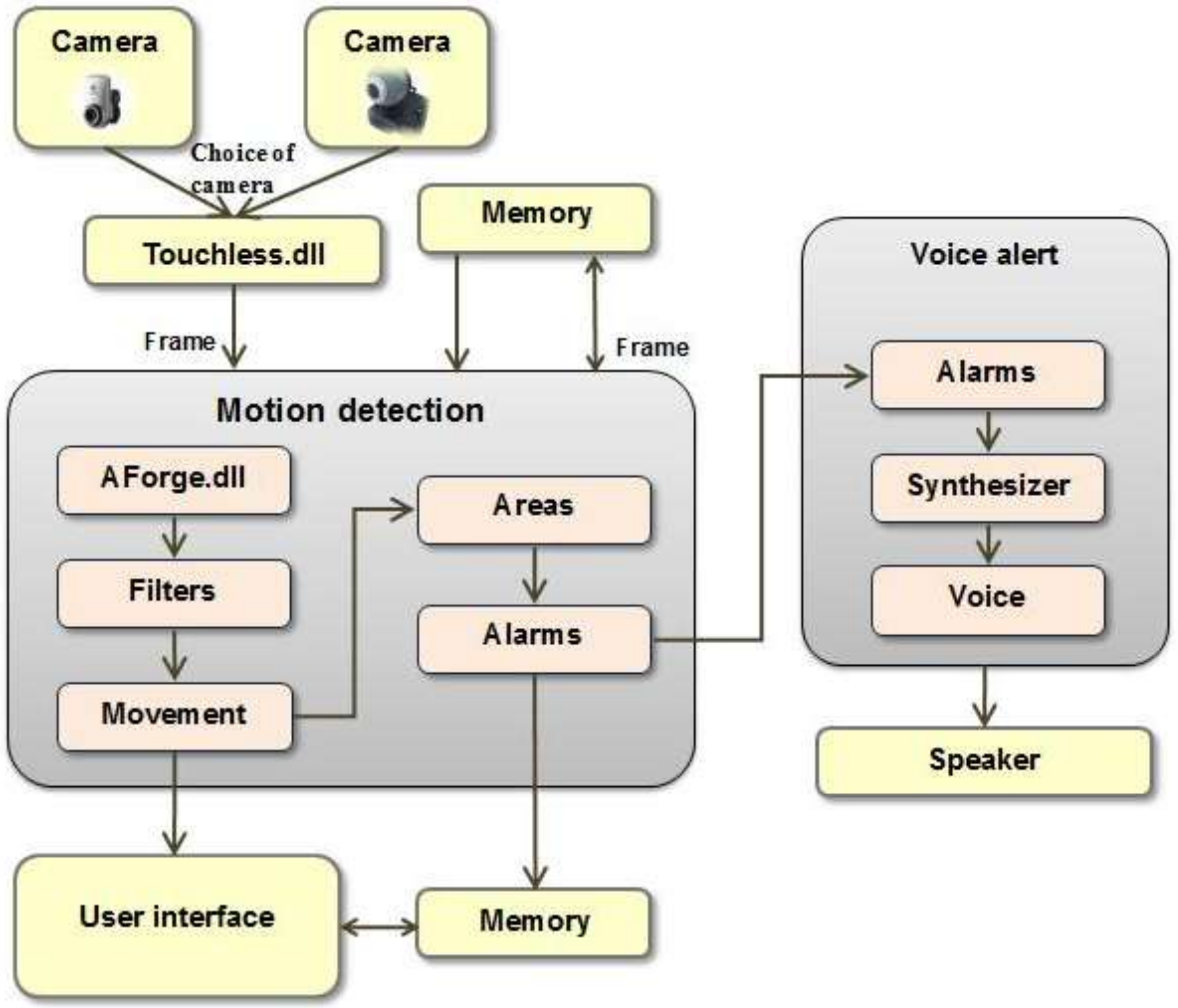


Fig.3. System architecture

2.1. Motion detection algorithm

One of the main ways to detect motion in a video stream is by comparing the two frames [4]. In the next section situations are discussed when comparing different scenes – first, last, previous, last and others.

To avoid small variations in the colors of individual frames copies with different levels of gray are used. The gray images are compared by difference filter. The result is a grayscale image where the biggest differences give brighther pixel in resultant image. At the same pixels, the resulting pixel is black, and at difference maximum (0 - white 255 - black) - white. On the resulting gray image threshold filter is applied. It forms a black and white image in which white pixels are differences in the two frames. Due to interference in the signal from the camera, often white pixels are obtained which are not caused by movement. This necessitates the use of erosion filter, which eliminates the self-white pixels. After applying these filters the result calculated is a “mask”. For a clear visualization the system turns a movement pixels to red color. For this purpose an extractchannel filter us used for the red color channel of the current frame. Merge Filter combines the mask with the image of the red channel, and the result is a gray scale image with white pixels in the movement. The last operation is the insertion of newly red channel in the current frame using the filter ReplaceChannel. In that way it sets a maximum amount of red in the current frame. Fig.4 shows the steps of the algorithm. It is clear that before step 4 image noise from the camera is present in the detected results. Erosion filter reduces the noise and prevents false alarms.

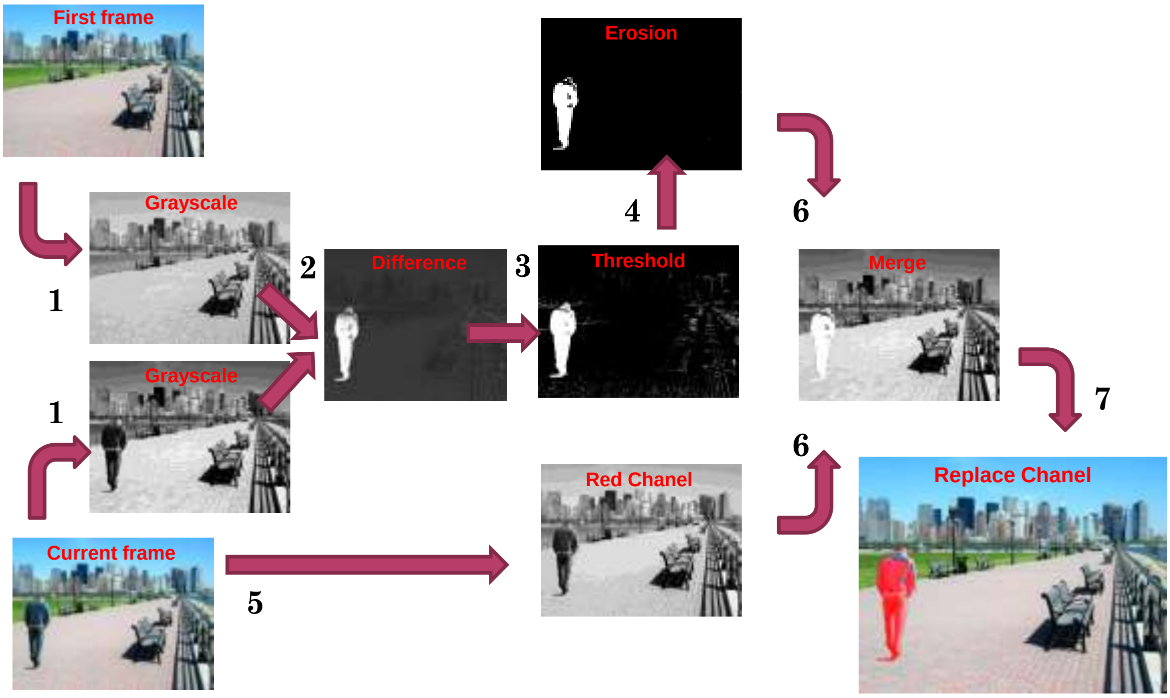


Fig.4. Motion detection algorithm [5]

2.2. Variations of the algorithm

Three different modifications of the algorithm for motion detection are considered and implemented in the developed system. Depending on the purposes and the application of the system the type of motion detection algorithm can be user adjusted.

a) Compare – first with last frame

In cases where light doesn't change or objects are moving slowly the appropriate algorithm is comparison of the first frame (*Fig.5*) with the last frame (*Fig.6*). The first frame is called the background. It is set at the start of the monitoring or dynamically during the monitoring process. This kind of algorithm is appropriate for static or rarely changing background. It is useful for motion detection in a room or a hall. The system is able to automatically or manually update the background in certain circumstances. Automatic update in the background is used when it dramatic differences in pixels is detected (changing the lights). User can update the background when the background has changed and the system reported movement in places where there is no any movements.

Fig.5.First frame

Fig.6.Last frame

Fig.7.Motion detection

b) Compare – previous with last frame

To avoid some of the disadvantages of the first detection approach a method for comparing the last two frames (previous and last frame) is defined. If there are some movements by static objects in the frame the algorithm doesn't detect an alarm. This method of detection is appropriate in terms of frequently changing light conditions or fast moving objects. If the object enters the danger zone and remain there stationary, the system will consider movement only when object crosses the area line. The disadvantage of this method is that slow moving objects can't be detected.

Fig.8.Previous frame

Fig.9.Last frame

Fig.10.Motion detection

c) Compare – combine

This motion detection approach is used as a combination of the previous two when there is no cear evidence which of the previous approaches is more suitable for the particular case. The algorithm compares all three frames – first, previous and last frame and thus makes the motion detection system quite universal. First, comparing the current frame with the previous and taking the number of different pixels a movement is reported if nonzero different pixels are found (*Fig.11*). Otherwise evaluation continues by comparing the current frame with the first – the background frame (*Fig.12*). In this way slow-moving objects will be detected.

Fig.11.Compare – previous with last frame *Fig.12.Compare - first with last frame*

2.3. Voice alarm

A situation of a frame with two specified monitored areas is shown on *Fig.13* as an example of the case for acquiring motion detection in specific user defined areas. After Erosion filter a picture with black and white pixels is obtained. In this example two masks are used for the two monitored areas. The number of masks depends on number of areas. The subtract filter takes two images (source and overlay images) of the same size and pixel format and produces an image, where each pixel equals to the difference value of corresponding pixels from provided images (if difference is less than minimum allowed value, 0, then it is truncated to that minimum value).

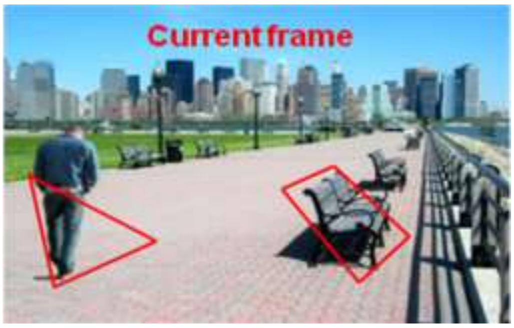
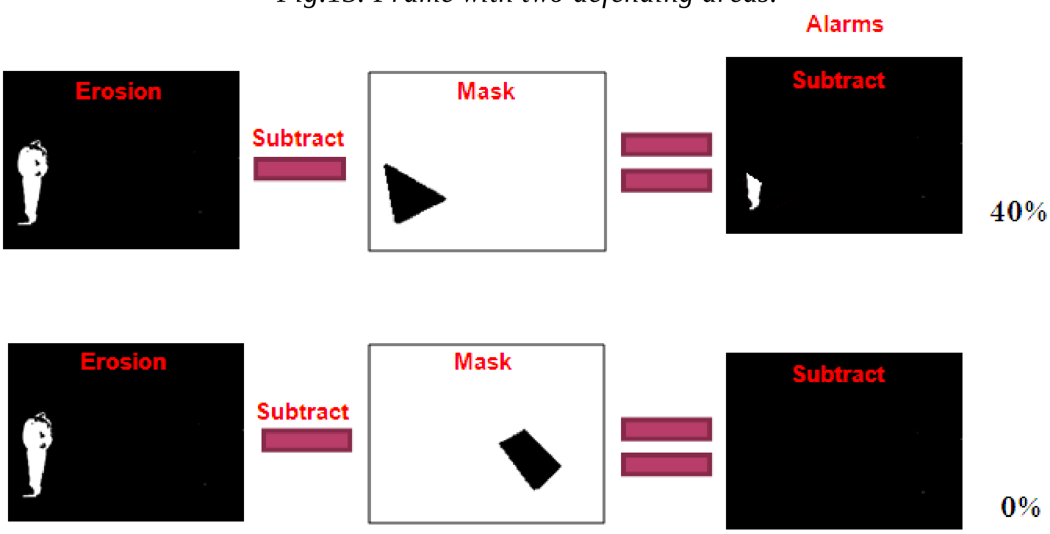
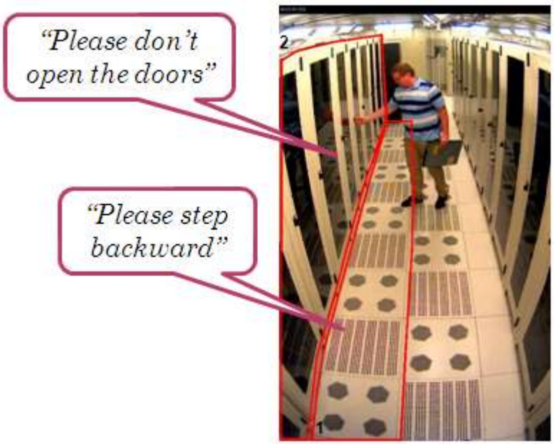


Fig.13. Frame with two defending areas.

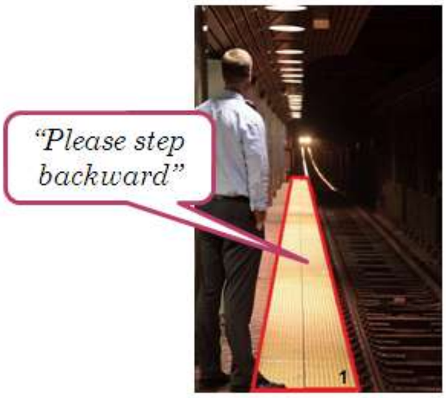


2.4. Voice phrases

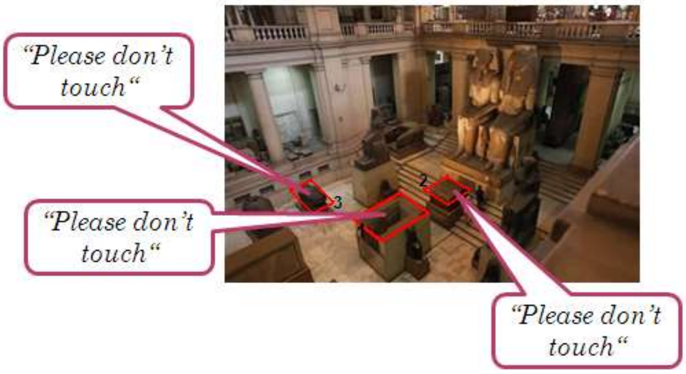
The system allow for user defined voice alarms. A default phrase is used in no user specific voice alarm is set. Same or different voice phrase can be used for the monitored (some examples are given on *Fig.14*).



a)



b)



c)

Fig.14 a) Default phrase b) Different phrases for the monitored areas c) Same phrases for all areas

2.5. Mutually exclusive areas

In order to provide possibility to activate the alarm by certain areas in different time the voice alarm need to be set to different phrases for each area. If the alarms from areas 2 or 3 is activated at the same time, the phrases will be heard one after another. In these cases when we have more than one area of voice alarm activation they have to be mutually exclusive. That means one alarm turn off another alarm by activating itself. Example is given on *fig.15*: when the subway is coming the alarm "Please step backward" will be turned off because people have to get on the subway.

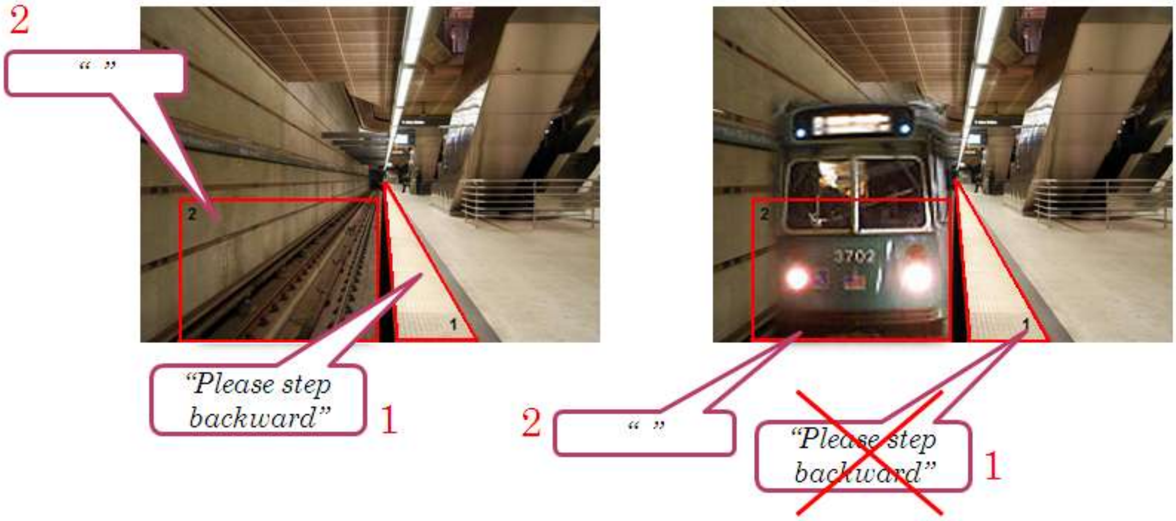


Fig.15. Mutually exclusive areas

3 Results and analysis

The implemented system is experimentally evaluated and tested in several case studies and conditions with different combinations of the adjustment parameters. The exact adjustment of all variables is a guarantee for successfull monitoring and less false alarms. The most difficult part is setting the camera to right posittion. The angle have to be considered in a way that people don’t walk between the camera and the monitored area – the position should be high enough providing the camera is from one of the sides or above the observed area. When the system is used outside, the weather differences can cause false alarm, therefore the sensitivity has to be adjusted very precisely and it has to be considered for each particular use of the system.
An example installation of the system for monitoring a picture is given on *fig.16*. The camera is positioned just above the observed object. This gives good view of the boundaries for the people in front of the object. Two different monitoring areas are defined with two different voice alarms. First warns the intruder to step backward in case of too close distance to the object. The second area monitors direct object access and issues a voice alarm for the intruder not to touch the object. The second area is set to have higher priority than the first and thus it turns off the voice alarm from the first area in case of mutual exclusion when the first one’s alarm isn’t active any more.

4 Conclusion

The system presented in the paper is aimed at video surveillance and motion detection with voice warning. It allows automaticaly, without the need for operator intervention, to detect movement in specific zones and issue spoken voice warnings – alarms. The main features of proposed solution are intuitive interface, choice of camera position, ability to set different areas of monitoring and motion detection, different voice alarms for each monitoring area, adjustable sensitivity, choice of different algorithms for video motion detection, storing detected motion causing alarm as a frame, functionality for mutual exclusion of several detections, choice of speaker’s voice and phrases.
The system can be used to prevent accidents in metro warning when people are approaching danger zone, warning visitors in a museum not to touch the exhibits, etc. The program can be used to draw attention of the computer operator when there is a movement detected.

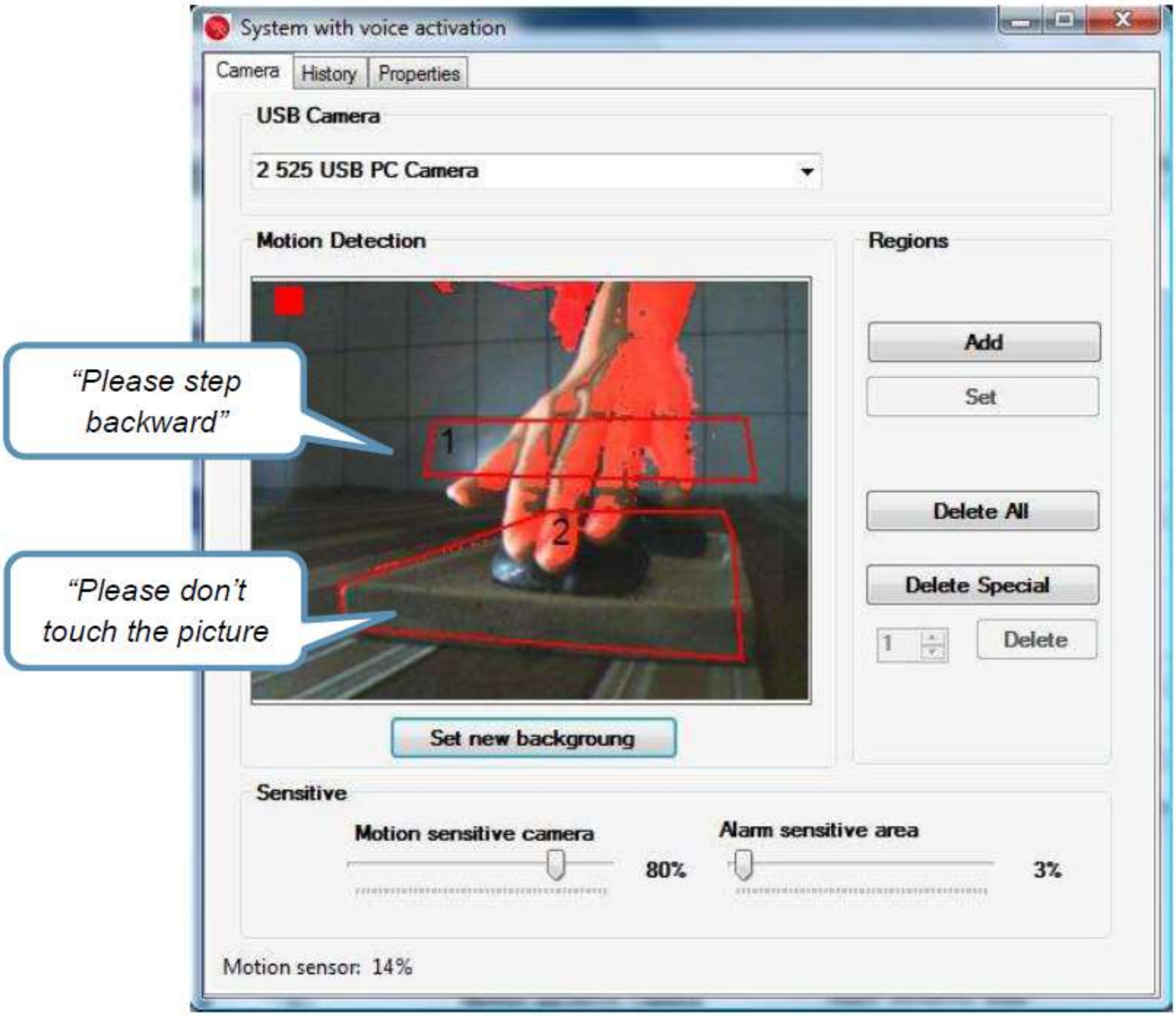


Fig. 16 Experimental setup with two areas

In the future system extension functionality can be added to allows simultaneous monitoring of multiple cameras and to record video when motion is detected.

References

[1] Special Section on Video Surveillance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Aug. 2000, vol. 22, no. 8, pp. 745-746,.

[2] R.Szeliski, *Computer Vision: Algorithms and Applications*, Microsoft Research, 2010.

[3] Axis Communications, *Technical guide to network video*, http://www.axis.com/products/video/about_networkvideo/vmd.htm

[4] R. Cutler, L. Davis, Robust real-time periodic motion detection, analysis, and applications, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2000, vol.22, №8, pp.781 – 796.

[5] AForge.NET - a C# framework for researchers in different areas of Computer Vision and Artificial Intelligence (http://www.aforgenet.com/framework/)

MAGDALENA FILCHEVA
Technical University of Sofia
Computer Science and Technologies
BULGARIA
magdalenafilcheva@abv.bg

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Computer Visualization in Differential Geometry

Razvan Hulea, Mihai Morariu
Coordinating Teachers: **Iuliana Munteanu, Liviu Ornea**

Abstract

With the aid of modern technology, scientific simulations are now more easily made than they were in the past. One of the branches of Mathematics for which applications can be created for a better understanding of the theoretical facts is Differential Geometry. Differential Geometry itself integrates techniques of differential and integral calculus, as well as linear and multilinear algebra, for studying the properties and the behaviour of surfaces and curves. The behaviour and properties of these entities are most of the time very difficult to comprehend without a visual representation. Morever, there are problems in fields such as Differential Geometry or Mechanics that involve the resolution of differential equations with initial values (also known as Cauchy problems). Explicit solutions are, in most cases, difficult to achieve, yet good approximations of the solutions can be yielded.

This is where computers become extremely useful. Using specialized software such as MATLAB, one can visually represent the desired surfaces and curves or obtain fairly good approximations of the solutions of problems imposed by the theory of differential geometry. In this paper, our goal is to present some of the applications we have created for some such problems and see why they are useful for a student who is willing to gain a better insight of the phenomenon.

1 Introduction

During the Differential Geometry course at our faculty, we were introduced to the theory of curves and surfaces. These entities are of great use in the real world. Applications of them can be found in Arhitecture (building shapes), Medicine (tissue engineering) or even Computer Science (3D modelling).

We wanted to create a package of easy-to-use graphical applications that could help a newbie student understand the theory presented at the course, how and why it works. We wanted those applications to be of use to the Professor, too, as a teaching material. When we designed them, we always kept in mind that a theoretical notion is always better understood when there is a “visual part” available for the model that is being discussed.

In the following, we will present some of the applications we have worked on.

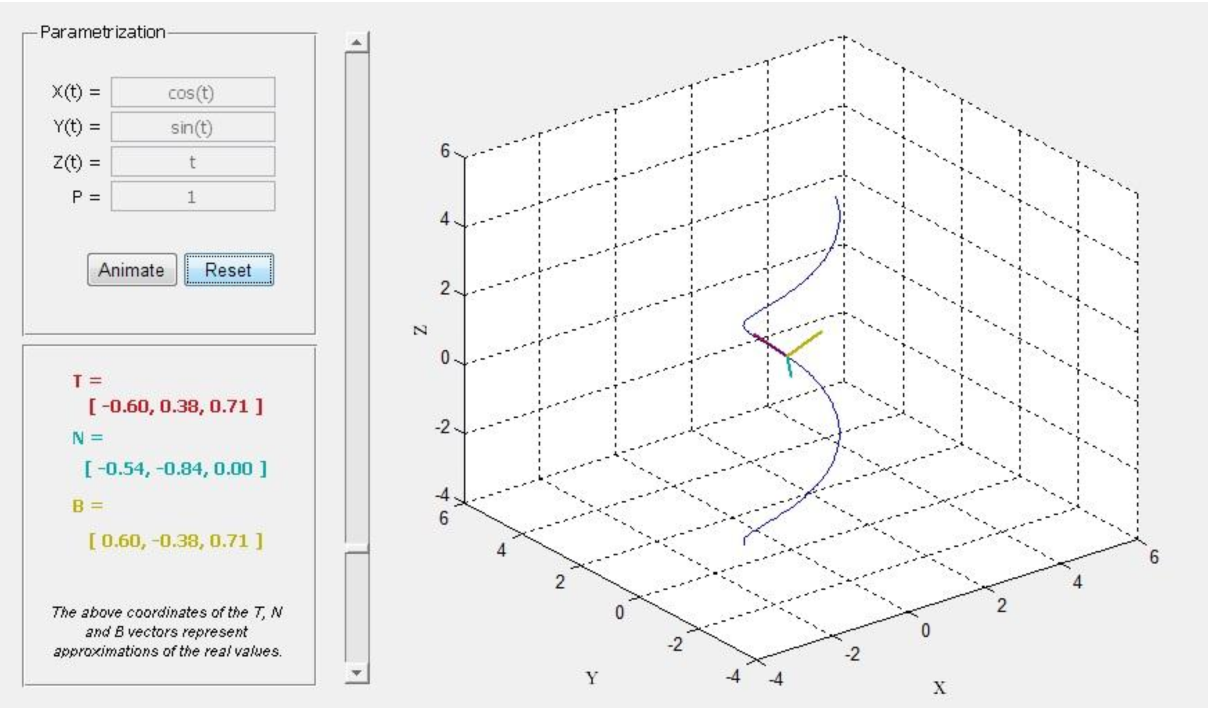
2 The Applications

2.1 The Frenet Frame

Mathematical Description

Let $c: [A, B] \rightarrow R^3$ be a differential curve and - P a point on c. Let T be the unit tangent vector at the curve in P, pointing in the direction of motion, N - the derivate of T, divided by its length, and B - the cross product between T and N. The frame (T, N, B) is called the Frenet frame [2]. By moving the frame along the curve and studying the rotation of its axis in a neighbourhood of P, one is able to tell "how much" c curves or twirls at that point. These properties are known as the "curvature" and "torsion" of the curve at the P point.

The Application



Our application constructs the Frenet frame for a given curve at a given point. The user is asked to enter the parametrization of the curve and the value of P. After pressing Ok, the program computes the T, N, B components and displays the frame. Finally, the frame can be animated along a neighbourhood of P by pressing the "Animate" button.

Implementation Details

For this program we used the GUIDE utility from MATLAB to generate the interface. We created four edit-text boxes for entering data (three for the parametrization of the curve and one for the value of the point at which the frame is constructed) and two push buttons (for confirming the action or resetting the application). The curve and the Frenet Frame are plotted in an "axes-plot" box and the output data (the components for the T, N, B vectors) is displayed in three static-text boxes.

The data received from the edit-text boxes is converted into symbolic objects using the *sym* function and a vector *c* that retains the parametrization of the curve is created. After checking that the received data is valid (that is, there are no empty text-boxes and the given curve is regular at the P point), the program computes the formulas for the T, N and B vectors:

```
c = [X; Y; Z];                // store the functions received from the X(t), Y(t) and Z(t) boxes
                              into c to obtain the parametrization of the curve

// [...] check if the edit-text boxes are empty

dc = diff(c);  // compute the differential of c

// […] check if the curve is regular at P

n_dc = simple(sqrt(dot(dc, dc)));     // compute the norm of the dc vector
T = simple(dc / n_dc);                // divide dc by its length to obtain T

dT = diff(T);                         // differentiate T
n_N = simple(sqrt(dot(dT, dT)));      // compute the norm of the dT vector

N = simple(dT / n_N);                 // divide dT by its length to obtain N
B = cross(T, N);                      // compute the cross product between T and N to obtain B
```

Using the *subs* function, the program replaces the *t* variable from the T, N and B vectors with the value at P to obtain the components of the vectors at this point:

```
TP = subs(T, t, P);
NP = subs(N, t, P);
BP = subs(B, t, P);
```

The program then checks whether the curve is in a two-dimensional or a three-dimensional space. We consider the curve to be to in a two-dimensional space if Z(t) is constant; that is, diff(Z) = 0. If the curve is in a three-dimensional one, we plot all T, N and B vectors; otherwise, only the first two are plotted (since B is the cross product of T and N; therefore, it does not lie in the curve's plane).

```
if (diff(Z) == 0)

    // draw the curve (p is an array with components ranging from p-5 to p+5);

    plot(subs(c(1), t, p), subs(c(2), t, p)); // plot c on the [P-5,P+5] interval

    // draw the T, N, B vectors , cP represents the coordinates of the P point,
    // TP is an array that holds the components of T and NP is an array that holds the components of
N, both taken  at the P point

    line([cP(1), cP(1) + TP(1)], [cP(2), cP(2) + TP(2)], 'Color', [0.83, 0.06, 0.10], 'LineWidth', 3);
    line([cP(1), cP(1) + NP(1)], [cP(2), cP(2) + NP(2)], 'Color', [0, 0.66, 0.66], 'LineWidth', 3);

    // update the T, N, B labels
```

```
            set(handles.lblTRez, 'String', sprintf('[ %-3.2f, %-3.2f ]', TP(1), TP(2)));
            set(handles.lblNRez, 'String', sprintf('[ %-3.2f, %-3.2f ]', NP(1), NP(2)))
            set(handles.lblBRez, 'String', 'NaN');
            [...]
        else

            plot3(subs(c(1), t, p), subs(c(2), t, p), subs(c(3), t, p));

            line([cP(1), cP(1) + TP(1)], [cP(2); cP(2) + TP(2)], [cP(3); cP(3) + TP(3)], 'Color', [0.83, 0.06, 0.10],
'LineWidth', 2);
            line([cP(1); cP(1) + NP(1)], [cP(2); cP(2) + NP(2)], [cP(3); cP(3) + NP(3)], 'Color', [0, 0.66, 0.66],
'LineWidth', 2);
            line([cP(1); cP(1) + BP(1)], [cP(2); cP(2) + BP(2)], [cP(3); cP(3) + BP(3)], 'Color', [0.73, 0.69, 0],
'LineWidth', 2);

            set(handles.lblTRez, 'String', sprintf('[ %-3.2f, %-3.2f, %-3.2f ]', TP(1), TP(2), TP(3)));
            set(handles.lblNRez, 'String', sprintf('[ %-3.2f, %-3.2f, %-3.2f ]', NP(1), NP(2), NP(3)));
            set(handles.lblBRez, 'String', sprintf('[ %-3.2f, %-3.2f, %-3.2f ]', BP(1), BP(2), BP(3)));
    end
```

After displaying the curve and the frame, the “Ok” button changes to “Animate”. This will animate the Frenet Frame along the curve on the [P-2, P+2] interval. A vector *r* with 81 components ranging from P-2 to P+2 is defined and the Frenet Frame is drawn at each of its values.

2.2 Curvature and Torsion

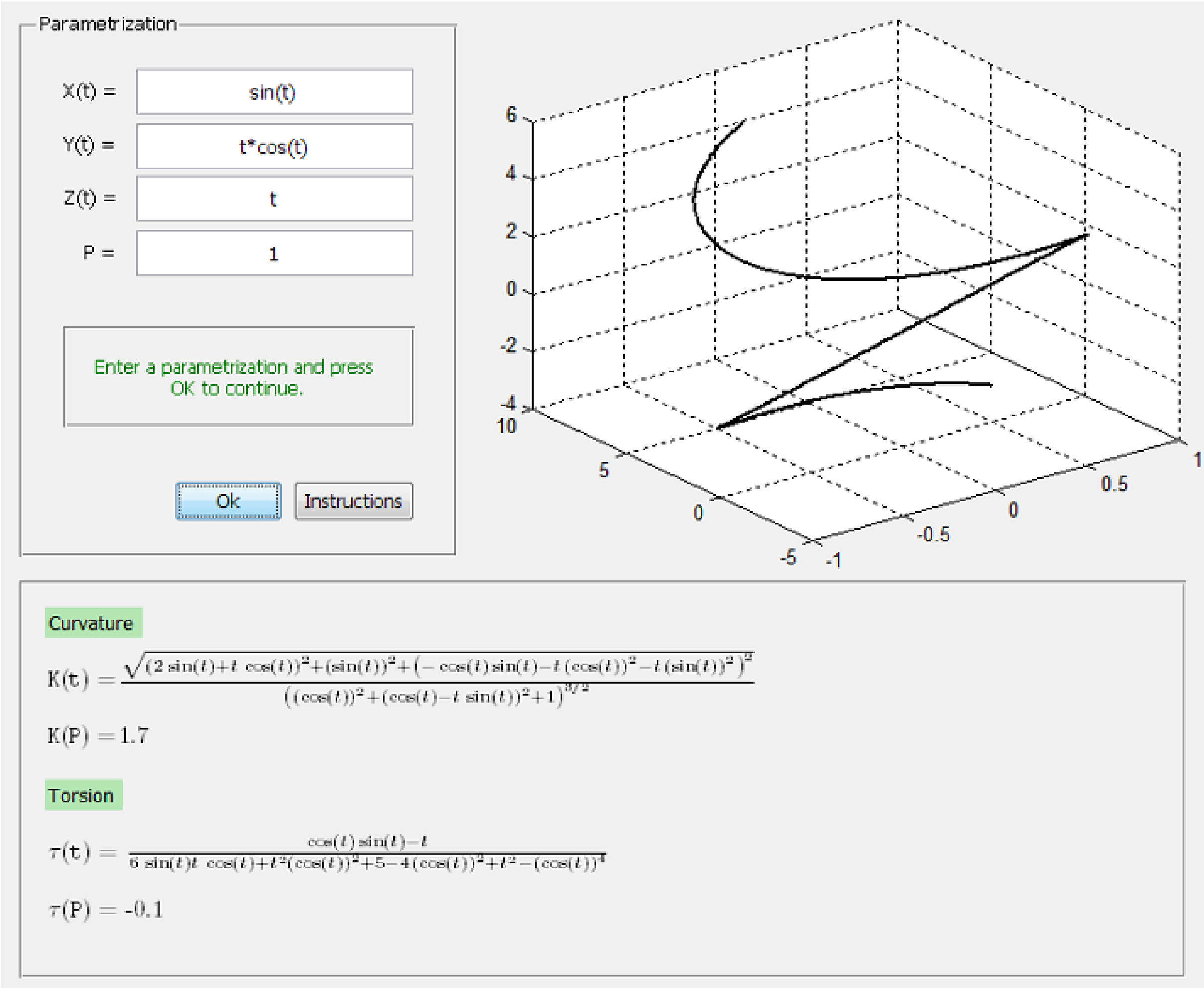
Mathematical Description

Let $c: [A, B] \to R^3$, $c(t) = (X(t), Y(t), Z(t))$ be a differential curve parametrized by arc length and P a point on c. Let T, N, B represent the Frenet frame associated at the point P.

Let $\frac{dT}{ds}$ represent the derivate of T with respect to the arc length parameter s. $\frac{dT}{ds}$is a non-unit vector which lies onto the direction of N, so $\frac{dT}{ds} = K(s)N(s)$. The value K(s) is called "the curvature of c at the P point". From the Frenet-Serret formulas [2], the derivate of B is the vector $\frac{dB}{ds} = -\tau(s)N(s)$. The value $\tau(s)$ is called "the torsion of c at the P point".

The Application

Our application computes the curvature and the torsion at a given P point lying on the curve. The user is asked to enter the parametrization of the curve and the value P for the point, $c(P) = (X(P), Y(P), Z(P))$. After pressing Ok, the program displays the formula for the curvature, the curvature at P, the formula for the torsion, and, the last one, the torsion at P.



Implementation Details

For this program we used the GUIDE utility from MATLAB to generate the interface. We created four edit-text boxes for entering data (three for the parametrization of the curve and one for the value of the point at which the curvature and torsion are computed) and two push buttons (for confirming the action or resetting the application). The curve is drawn in an "axes-plot" box and the output data (the formulas for the curvature and torsion and their values at P) are displayed in another such box. We chose to display the formulas in an axes-plot box because it allows us to integrate LaTeX code into MATLAB for a neat representation of the formulas.

We computed the formulas for the curvature and torsion symbolically and updated a "loading" label throughout the code to see the progress of the computation:

```
c = [X; Y; Z];                  // store the functions received from the X(t), Y(t) and Z(t) boxes
                                   into c to obtain the parametrization of the curve
c1 = jacobian(c, t);            // compute the differential of c
c2 = jacobian(c1, t);           // compute the second differential of c
c3 = jacobian(c2, t);           // compute the third differential of c

set(handles.lblLoading, 'String', 'Processing data... [16%]');      // update the „loading" label
drawnow update;

// compute the formulas for the curvature and torsion
```

```
A = det([diff(Y) diff(Z); diff(Y, 2) diff(Z, 2)]);
B = -det([diff(X) diff(Z); diff(X, 2) diff(Z, 2)]);
C = det([diff(X) diff(Y); diff(X, 2) diff(Y, 2)]);

K1 = sqrt(A^2 + B^2 + C^2);
K2 = sqrt(diff(X)^2 + diff(Y)^2 + diff(Z)^2) ^ 3;
KF = simple(K1 / K2);                        // KF will store the curvature's formula
K = subs(K1, t, P) / subs(K2, t, P);         // K will store the value of KF at P

// [...]

TAU1 = det([c1 c2 c3]);
TAU2 = A^2 + B^2 + C^2;
TAUF = simple(TAU1 / TAU2);                  // TAUF will store the torsion's formula
TAU = subs(TAU1, t, P) / subs(TAU2, t, P);   // TAU will store the value of TAUF at P
```

Finally, we printed the result in the second axes-plot box using LaTeX code:

```
set(gcf, 'CurrentAxes' , handles.axesRezultat);
cla;
text(.02, .90, 'Curvature ', 'FontName', 'Tahoma', 'FontSize', 9, 'BackgroundColor', [.7 .9 .7], 'Interpreter', 'tex');
text(.02, .75, ['$\mathtt{K(t) = }', latex(KF), '$'], 'FontName', 'Tahoma', 'FontSize', 12, 'BackgroundColor', [.9412 .9412 .9412], 'Interpreter', 'latex');
text(.02, .60, ['$\mathtt{K(T) = }$', sprintf('%3.1f', K)], 'FontName', 'Tahoma', 'FontSize', 12, 'BackgroundColor', [.9412 .9412 .9412], 'Interpreter', 'latex');
```

2.3 Osculating Circle

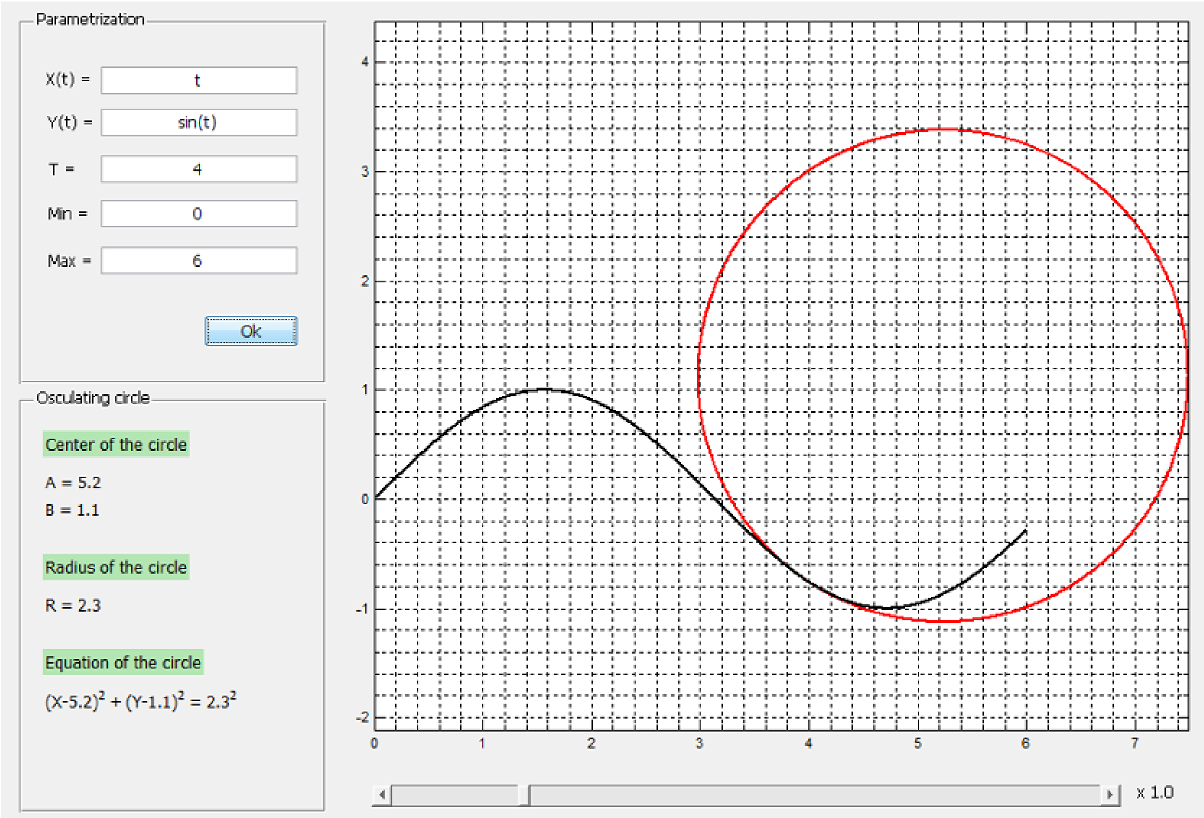
Mathematical Description

Let $c: [A, B] \to R^3$, $c(t) = (X(t), Y(t), Z(t))$ be a differential curve parametrized by arc length and P a point on c. Let T, N, B represent the Frenet frame associated at the point P. The plane determined by T and N is called the osculating plane [1].

The osculating circle of c at P is the circle that lies in the osculating plane and has the same tangent as c at that point and the same curvature.

The Application

Our application computes the center and the radius of the osculating circle. The user is asked to enter the parametrization of the curve and the value P for the point $c(P) = (X(P), Y(P), Z(P))$. After pressing Ok, the program displays the curve with the osculating circle at P, its center, radius and equation.



Implementation Details

For this program we used the GUIDE utility from MATLAB to generate the interface. We created five edit-text boxes for entering data (two for the parametrization of the curve, one for the value of the T point at which the osculating circle is constructed) and two for the [Min, Max] boundaries of the curve. We added one button for confirming the action, one "axes-plot" box on which the curve and the circle are drawn and one axes-plot box for displaying the centre of the circle, its radius and equation. We used this instead of a simple text-box so that we could insert LaTeX code that would produce a neat representation of the output.

The program first checks if the input data is valid; that is, if all the fields have been filled in and the values for Min, Max and T are chosen so that Min <= T <= Max. It then computes the centre (A, B) of the osculating circle and its radius at the chosen point:

```
X1 = diff(X);                    // compute the differential of the X function
X2 = diff(X1);                   // compute the second differential of the X function
Y1 = diff(Y);                    // compute the differential of the Y function
Y2 = diff(Y1);                   // compute the second differential of the Y function

// determine the formula for the centre and the radius of the osculating circle

A = X + Y1*(X1^2+Y1^2) / (X2*Y1-X1*Y2);
B = Y - X1*(X1^2+Y1^2) / (X2*Y1-X1*Y2);

K = (X1*Y2-X2*Y1) / (sqrt(X1^2+Y1^2)^3);
R = 1 / abs(K);
```

```
// compute the centre and the radius of the osculating circle at T

Rv = subs(R,t,T);
Av = subs(A,t,T);
Bv = subs(B,t,T);
```

The curve and the circle are then plotted onto the axes-plot box and the centre, radius and equation of the circle are displayed:

```
tv = linspace(LMIN, LMAX, 1000);        // define the vector used for drawing the curve
tc = linspace(0, 2*pi, 3000);           // define the vector used for drawing the circle

set(gcf, 'CurrentAxes', handles.axesPlot);
cla reset;

plot(subs(Av + Rv*cos(t), t, tc).*ones(size(tc)), subs(Bv + Rv*sin(t), t, tc).*ones(size(tc)), 'Color', 'r', 
'LineWidth', 1.5);       // plot the osculating circle
hold on;
plot(subs(X, t, tv).*ones(size(tv)), subs(Y, t, tv).*ones(size(tv)), 'Color', 'k', 'LineWidth', 1.5);    // plot the 
curve

// [...]

set(gcf, 'CurrentAxes', handles.axesRezultat);
cla;

text(.075, .95, 'Center of the circle', 'FontName', 'Tahoma', 'FontSize', 9, 'BackgroundColor', [.7 .9 .7]);
text(.075, .85, ['A = ', sprintf('%3.1f', Av)], 'FontName', 'Tahoma', 'FontSize', 9, 'BackgroundColor', 
[.9412 .9412 .9412]);
text(.075, .78, ['B = ', sprintf('%3.1f', Bv)], 'FontName', 'Tahoma', 'FontSize', 9, 'BackgroundColor', 
[.9412 .9412 .9412]);
text(.075, .63, 'Radius of the circle', 'FontName', 'Tahoma', 'FontSize', 9, 'BackgroundColor', [.7 .9 .7]);
text(.075, .53, ['R = ', sprintf('%3.1f', Rv)], 'FontName', 'Tahoma', 'FontSize', 9, 'BackgroundColor', 
[.9412 .9412 .9412]);
text(.075, .38, 'Equation of the circle', 'FontName', 'Tahoma', 'FontSize', 9, 'BackgroundColor', [.7 .9 
.7]);
text(.075, .28, ['(X-', sprintf('%3.1f', Av), ')', '^2 + (Y-', sprintf('%3.1f', Bv), ')', '^2 = ', sprintf('%3.1f', Rv), 
'^2'], 'FontName', 'Tahoma', 'FontSize', 9, 'BackgroundColor', [.9412 .9412 .9412]);
```

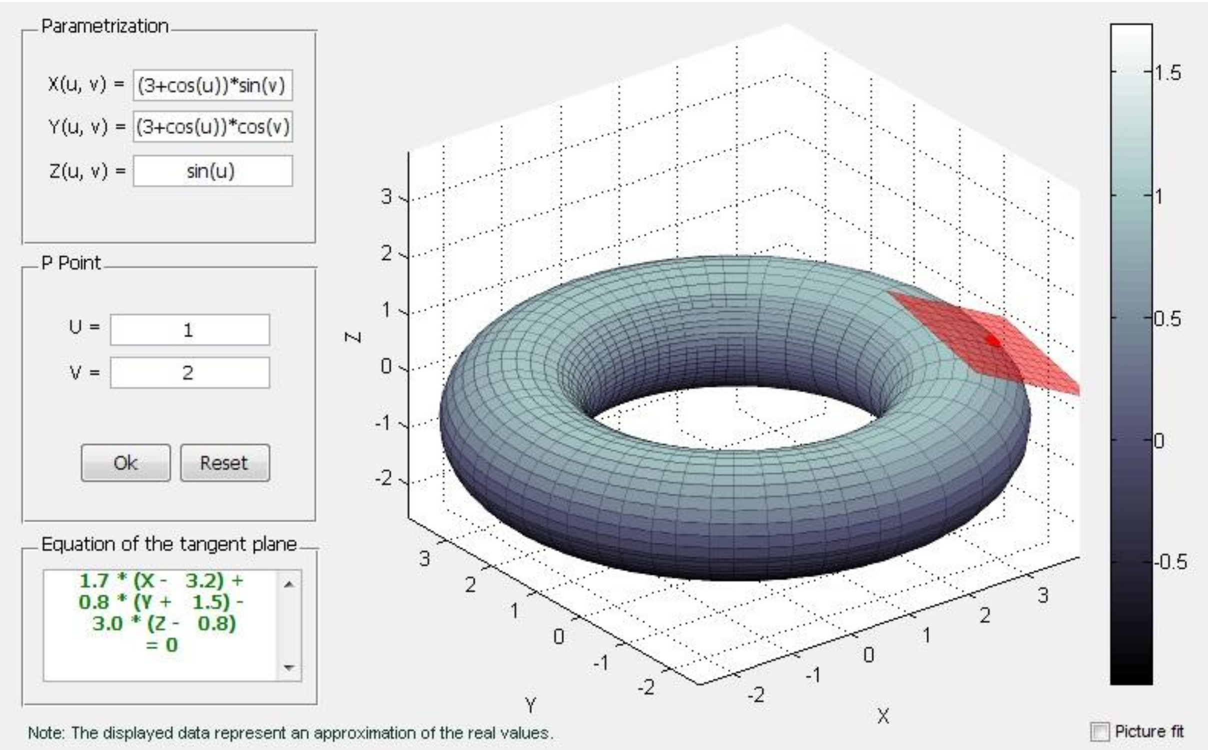
2.4 Tangent Plane

Mathematical Description

Let $h: U \times V \to R^3$, $h(u,v) = (X(u,v), Y(u,v), Z(u,v))$ be the parametrization for a differential surface, M. Let $h_1 = \frac{dh}{du}$ be the derivate of h with respect to u and $h_1 = \frac{dh}{dv}$ the derivate of h with respect to v, both taken at the P point. The plane spanned by the vectors h_1 and h_2 is called "the tangent plane of the M surface at the P point" [5].

The Application

Our application computes the equation of the tangent plane. The user is asked to enter the parametrization of the curve and the values U, V for the point $h(u,v) = (X(u,v), Y(u,v), Z(u,v))$. After pressing Ok, the program displays the given surface, the given point, the tangent plane and the plane's equation.



Implementation Details

For this program we used the GUIDE utility from MATLAB to generate the interface. We created five edit-text boxes for entering data (three for the (X(u, v), Y(u, v), Z(u, v)) parametrization of the surface and two for the (U, V) point at which the tangent plane is computed). We added two push buttons (one for confirming the action and one for resetting the application, respectively) and one text-box for displaying the equation of the tangent plane. We also added an axes-plot box for displaying the surface and the plane and one check-box that fits the picture in the axes-plot if checked.

The program first checks to see if the received data is valid: all the required fields have been filled in and the given parametrization represents a surface (that is, the jacobian of the parametrization has rank 2). We also check to see if the surface is regular at the chosen (U, V) point (the cross product between the differential with respect to u and the differential with respect to v must not be null at (U, V)). This is a necessary condition for computing the tangent plane.

The program computes the formula for the tangent plane and plots it in the axes-plot box. If the "Picture fit" check-box is checked, the axes are adjusted so that the surface and the plane can wholly be seen on the screen. This is done by computing the maximum and minimum values on the X, Y and Z axis and taking the maximum (LMax) and minimum (LMin) of these values, respectively. The axes-plot box will be set to the [LMin, LMax] interval on all of the axis.

```
if (isempty(X) | isempty(Y) | isempty(Z) | isempty(U) | isempty(V))
    msgbox('Please fill in all the required fields!', 'Error', 'error', 'modal');
```

```
    else
        S = [X; Y; Z]; // symbolic expression for the parametrization of the surface
        Ru = jacobian(S, u); // compute Ru, the differential with respect to u
        Rv = jacobian(S, v); // compute Rv, the differential with respect to v
        J = [Ru Rv]; //compute the jacobian of the surface

    if (rank(J) == 2) // check to see if the jacobian of the parametrization has rank 2
        ABC = subs(simple(cross(Ru, Rv)), {u, v}, {U, V}); // compute the cross product between Ru and Rv at
the (U, V) point

    if (isequal(ABC, [0; 0; 0]) | isnan(ABC(1)) | isnan(ABC(2)) | isnan(ABC(3))) // if the cross product is null or not
                                                                defined at the (U, V) point, display an error message

        msgbox('The chosen point is not regular. Cannot build the tangent plane.', 'Error', 'error', 'modal');
    else

        // [...]

        iu = U-5:0.2:U+5; // the surface to be drawn is h : (U-5, U+5) x (V-5, V+5) -> R^3, h(u, v) = (X(u, v),
Y(u,v), Z(u, v))
        iv = V-5:0.2:V+5;
        [vu, vv] = meshgrid(iu, iv);

        // compute the maximum and minimum values on the X, Y and Z axis


        max_XYZ = [max(subs(X, {u, v}, {iu, iv})); max(subs(Y, {u, v}, {iu, iv})); max(subs(Z, {u, v}, {iu, iv}))];
        min_XYZ = [min(subs(X, {u, v}, {iu, iv})); min(subs(Y, {u, v}, {iu, iv})); min(subs(Z, {u, v}, {iu, iv}))];

        global LMin LMax;

        LMin = min(min_XYZ);
        LMax = max(max_XYZ);

        surf(subs(X, {u, v}, {vu, vv}), subs(Y, {u, v}, {vu, vv}), subs(Z, {u, v}, {vu, vv}), 'EdgeAlpha', .2); //plot the
surface on the [U-5,U+5]x[V-5,V+5] interval

        // [...]


        XP = subs(X, {u, v}, {U, V}); // compute the coordinates of the chosen point on the surface
        YP = subs(Y, {u, v}, {U, V});
        ZP = subs(Z, {u, v}, {U, V});
        plot3(XP, YP, ZP, 'x', 'Color', 'r', 'LineWidth', 8); //plot the point P

        // [...]

        if (ABC(3) ~= 0) // compute the formula for the tangent plane on the [XP-1, XP+1] x [YP-1, YP+1]
interval
            [vx, vy] = meshgrid(XP-1:0.2:XP+1, YP-1:0.2:YP+1);
            vz = -1/ABC(3)*(ABC(1)*(vx-XP)+ABC(2)*(vy-YP)-ABC(3)*ZP);
        elseif (ABC(2) ~= 0)
            [vx, vz] = meshgrid(XP-1:0.2:XP+1, ZP-1:0.2:ZP+1);
            vy = -1/ABC(2)*(ABC(1)*(vx-XP)+ABC(3)*(vz-ZP)-ABC(2)*YP);
        else
            [vy, vz] = meshgrid(YP-1:0.2:YP+1, ZP-1:0.2:ZP+1);
            vx = -1/ABC(1)*(ABC(2)*(vy-YP)+ABC(3)*(vz-ZP)-ABC(1)*XP);
        end

        surf(vx, vy, vz, 'FaceColor', [1 0 0], 'FaceAlpha', .5, 'EdgeAlpha', .2); // plot the tangent plane

        // [...]

    end

function chkIncadrare_Callback(hObject, eventdata, handles)
```

```
global LMin Lmax;

val = get(hObject, 'Value'); // get the state of the check-box

if (val == 1) // if checked, let MATLAB adjust the axis
        set(handles.axesPlot, 'XLimMode', 'auto');
        set(handles.axesPlot, 'YLimMode', 'auto');
        set(handles.axesPlot, 'ZLimMode', 'auto');
else
        set(handles.axesPlot, 'XLim', [LMin, LMax]); // else, adjust the axis according to the LMin and
LMax values
        set(handles.axesPlot, 'YLim', [LMin, LMax]);
        set(handles.axesPlot, 'ZLim', [LMin, LMax]);
end
```

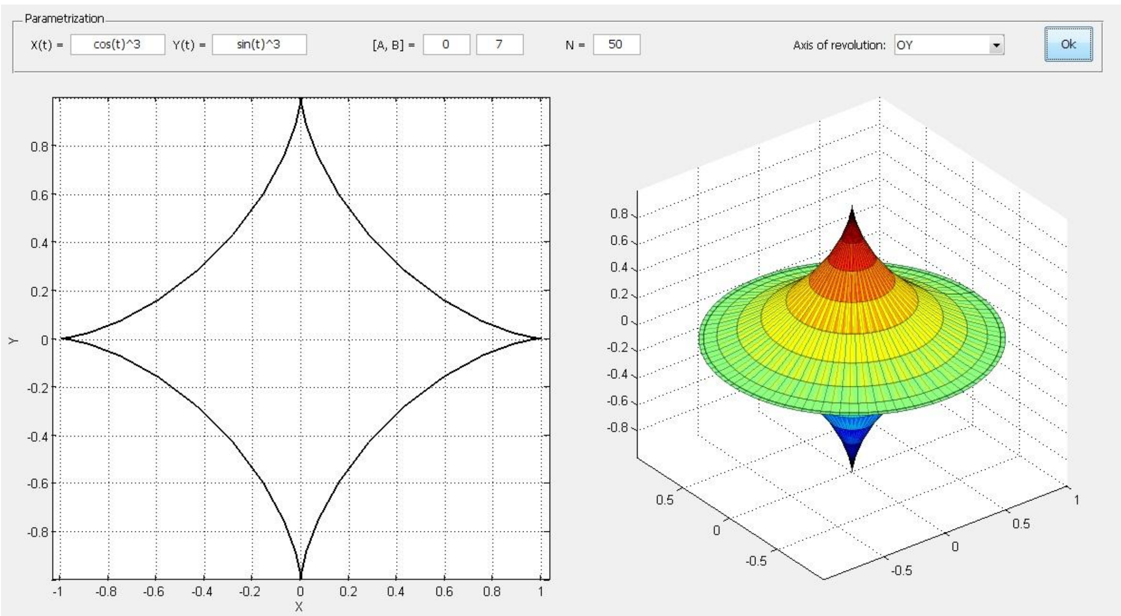
2.5 Surface of Revolution

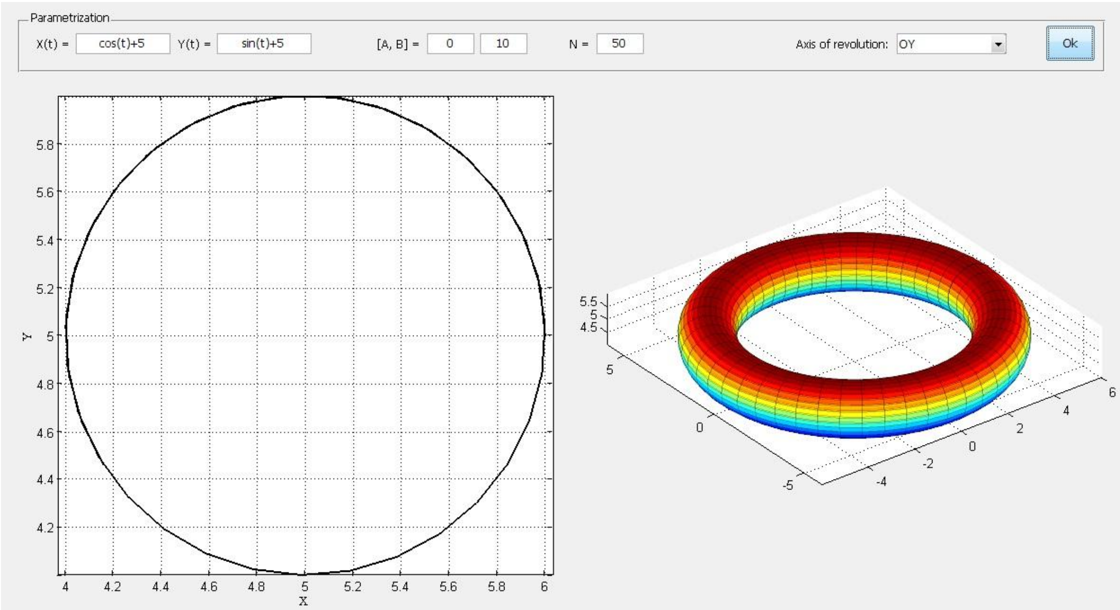
Mathematical Description

Let $c: [A,B] \to R^2$ and $c(t) = \big(X(t), Y(t)\big)$ be a differential curve in the XY plane. The surface obtained by rotating c around one of the axis is called "the surface of revolution of the c curve". The parametrization for the surface of revolution around the OX axis is $(Y(t) * \cos(u), Y(t) * \sin(u), X(t))$ and $(X(t) * \cos u, X(t) * \sin(u), Y(t))$ around the OY axis, where $t \in [a, b], u \in [0, 2 * \pi]$.

The Application

Our application displays the surface of revolution for a given curve around the OX or OY axis. The user is asked to enter the parametrization of the curve, the extremities A, B of the [A, B] interval, as well as N, the number of points out of which the curve is to be constructed. After selecting the axis of revolution and pressing Ok, the program displays the appropriate surface of revolution.





Implementation Details

For this program we used the GUIDE utility from MATLAB to generate the interface. We created five edit-text boxes for entering data. Two of them represent the parametrization of the curve, two represent the interval on which the curve is plotted and one for N, where N is the number of points that make up the curve and N^2 is the number of points that make up the surface. We also added one button for confirming the action, one popup menu for selecting the axes of revolution, one “axes-plot” box on which the curve is drawn and one axes-plot box for displaying the surface of revolution.

The program first checks if the entered data is valid: all the fields have been filled in, N >= 5 (we set 5 to be the minimum number of points for the curve) and the values for A and B are chosen so that A < B. It then checks which of the axes of revolution is selected from the popup menu and plots the appropriate surface of revolution.

```
if (isempty(X) || isempty(Y) || isempty(A) ||isempty(B))
        msgbox('Please fill in all the required fields!', 'Error', 'warn', 'modal');
else
        if (N < 5)
                msgbox('Please choose a value for N so that N >= 5!', 'Error', 'warn', 'modal');
        else
                if (A >= B)
                        msgbox('Please choose two values for A and B so that A < B!', 'Error', 'warn', 'modal');
                else
                        AX = get(handles.popupAxes, 'Value'); // get the state of the popup menu

                // [...]


                        T = linspace(A, B, N);
                        plot(axesC, subs(X, t, T).*ones(size(T)), subs(Y, t, T).*ones(size(T)), 'Color', 'k',
'LineWidth', 1.2); // plot the curve
                        [T, U] = meshgrid(linspace(A, B, N), linspace(0, 2*pi, N));

switch AX
        case 1 // if the user selected the OX axis
```

```
                X1 = subs(Y, t, T).*cos(U);
                Y1 = subs(Y, t, T).*sin(U);
                Z1 = subs(X, t, T).*ones(size(X1));

                surf(axesS, Z1, Y1, X1, 'EdgeAlpha', .2); // plot the surface of revolution around the OX axis

                case 2 // if the user selected the OY axis

                X1 = subs(X, t, T).*cos(U);
                Y1 = subs(X, t, T).*sin(U);
                Z1 = subs(Y, t, T).*ones(size(X1));

                surf(axesS, X1, Y1, Z1, 'EdgeAlpha', .2); // plot the surface of revolution around the OY axis
end

// [...]
```

3 Conclusions and Future Work

We believe these applications will prove useful for anyone interested in the Differential Geometry field. We are planning to develop more of them so that they could be used within the course at our faculty.

For example, one such application could be created for the isoperimetric inequality, which states that if we take a simple, closed, plane curve of length L and denote the area of the region bounded by this curve by A, then the inequality $4\pi A \leq L^2$ is always true. This means that that among all simple, closed, plane curves of length L, the circle is the one with the largest area of the region bounded it.

We are planning to develop an application for this theorem that asks the user to enter the parametrization of a closed curve on a given interval. The program would draw the curve on the left side of the screen. The computer would determine its length and draw a circle of the same length on the right side. The user would then be able to deform the curve by dragging points from it and see, in real time, that indeed the area bounded by the new deformed curve is always less than the area bounded by the circle.

Another application we are planning to develop is related to the fundamental theorem of curves. This states that, for two given functions $K, \tau: [A, B] \rightarrow R$, $K(s) \geq 0, \forall s \epsilon [A, B]$there exists a single curve parametrized by arc length that has the K and τ curvature and torsion at every point in the (A, B) interval.

The user would input A, B and the formulas for K and τ. The program would then display the curve which has the curvature and torsion specified by K and τ.

4 References

[1] *Amos Gilat* – MATLAB – An Introduction with Applications, John Wiley & Sons Inc., ISBN 0-471-43997-5 2003.

[2] *Manfredo do Carmo* - „Differential Geometry of Curves and Surfaces", Prentice Hall ,ISBN 0-13-212589-7, 1976

[3] *John Oprea* - „Differential Geometry and its Applications", Prentice Hall, NJ 07458,ISBN 0-13-340738-1, 1997

[4] *Liviu Ornea* - "O introducere in geometria diferentiala" , (note de curs 2010)

[5] *Iuliana Paraschiv-Munteanu - D. Massier,* Probleme de mecanică rezolvate în MATLAB (Mechanical problems solved in MATLAB), Editura Universității București (Bucharest University Publishing House), ISBN 978-973-737-518-6, 2008

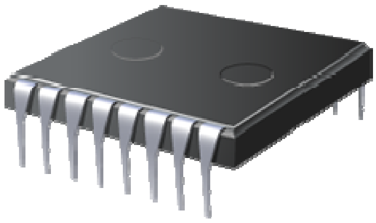
[6] *** MATLAB, User Guide, 2001

RAZVAN HULEA
University of Bucharest, Faculty of Mathematics and Computer Science
Department of Applied Mathematics
Str. Academiei 14, Sector 1, 010014, Bucuresti
ROMANIA
E-mail: hulea.razvan89@gmail.com

MIHAI MORARIU
University of Bucharest, Faculty of Mathematics and Computer Science
Department of Applied Mathematics
Str. Academiei 14, Sector 1, 010014, Bucuresti
ROMANIA
E-mail: mihaimorariu@gmail.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Simulating logic circuits



Andrei Ivan, Rădoi Ovidiu
Teacher Coordinator: Anca Vasilescu

Abstract

Simulation is a key part in understanding any system. Advances in computer science have allowed us to reproduce many aspects of a system including how it works and problems that may come up in implementing it in a working environment. Logic circuits are the key part of any computational system.

We have created a tool that can be used to design a logic circuit and simulate its functionality. This allows the users to better understand the workings of such circuits and prevent certain problems before actually building it. This tool can enable its users to design circuits faster and also understand the topic of Computer Architecture.

1. Introduction

1.1. Anything goes

The world we live in is constant changing. Everything in our environment is making large steps towards the future, according to the people's needs. We, being in the power to change the face of the earth, must take care that this changing process goes well and without affecting the environment and its living beings (including us). In order to achieve that, people must have a certain level of knowledge, discipline and respect for the nature to be able to live in peace and harmony.

But what leads to that level of knowledge needed for our future existence? The answer is education. Starting at an early age, education is the pillar of forming us as humans.

1.2. Computer simulation

Simulation is a key part in understanding any system. Advances in computer science have allowed us to reproduce many aspects of a system including how it works and problems that may come up in implementing it in a working environment. Computer simulations are used extensively in many areas of science, engineering, physics, chemistry, architecture and others. The use of such simulations enables people to quickly build a computer model of a system and visualize its interconnected components. It also allows them to quickly spot potential problems and redesign the model to overcome such problems.

All this can be done in the early stages of the design process, before the system has been built. If we were to compare the traditional design process which involved only pen and paper, we can agree that using a computer can speed up the design process and also make it more reliable before any effort has been spent in building an actual prototype. Computer simulation enables faster, cheaper and a more reliable design of any system.

1.3. Computer education

Learning is changing as well, especially the technologies of learning. E-education is the process of learning where computers are used at each possible step of the process. Obviously, computers cannot replace human teachers as children need affection, compassion and communication. Computers, on the other side, are good tools for the learning process. One of the reasons computers seem to be excellent tools for learning is the attraction they exercise upon children and teen-agers. But if we go deep into this phenomenon, it is possible to observe that this attraction is due to two main reasons: what we call the "cosmetics" and the "video game" effects.

In the first case, users are attracted by the multi-media effects, such as fascinating pictures, sound and animation. In the second, by an excitement similar to that felt when playing a video game: the setting is perfectly, mathematically defined, and the user feels the power of completely dominating the machine. When the desired result is not reached - either not being able to force the machine to do what was expected, or not being able to discover by trial-and-error an appropriate command or sequence of commands - the user enters into a state of excitement which stems from a purely intellectual challenge. We mean here a challenge which has nothing to do with physical ability (such as those required in sports). The certainty that one will eventually discover the right way to do something with the computer attract the user to such a degree that he forgets everything else, entering what we have called "the obsessive user state".

2. Preliminaries

2.1. Information technology: a high-level science

Although it is not a very old scientific domain, Information technology is nowadays perhaps the most known one. This is because of the fact that computers became widely used, making our life much easier and enjoyable.

Maybe for the ones of us who are in charge of computer programming and hardware designing life would be simpler if computers didn't exist. Being the result of such many domains (Algorithmics, Geometry, Numerical Calculus, Data Structures, Boolean Algebra) is giving the specialists in this domain a hard time because a bit of knowledge about all of them is necessary.

Also, it is unnecessary to mention the hard work needed to become a specialist in this domain. Here, in Romania, we study both mathematical principles behind computers as well as the practical ones. A discipline that is caught somewhere on the border between theoretical and practical aspects is the *Computer Architecture* discipline.

2.2. Computer architecture

In computer engineering, computer architecture is the conceptual design and fundamental operational structure of a computer system. It is a blueprint and functional description of requirements and design implementations for the various parts of a computer, focusing largely on the way by which the central processing unit (CPU) performs internally and accesses addresses in memory.

It may also be defined as the science and art of selecting and interconnecting hardware components to create computers that meet functional, performance and cost goals.

We won't go into deep details about its subcategories, but we will stop at one of them. It represents the engine of the chips: the logic circuits. [4], [5]

2.3. The main purpose of the application

From the practical point of view, this paper refers to an educational piece of software developed as a technological tool for understanding the logic circuits operation. By using such an interactive designer application, we have the main advantage and the opportunity of removing all the possible design errors even before proceeding to the expensive component manufacturing and assembly. Based on a friendly user interface developed in C# programming language [2], the student can interact with its logic circuit, flip switches, change the input data, and watch the output data changes for sequential models starting from the main window represented in Figure 1.

From the educational point of view [1], our application, *Logic Circuit Designer*, is aimed to help both those students and users, studying computer architecture or more specifically, digital logic circuits. This product is a simulator for logic circuits, allowing the student to better visualize and understand how a logic circuit works, by enabling him to build whatever logic circuit he wants, and visually simulate its operation. Based on a friendly user interface, the student can interact with its logic circuit, flip switches, change the input data, and watch the output data changes for sequential models.

We are sure that this tool will be of great help to any student who has interest in this field, and enable him to better understand the logic circuit design process.

3. The application

3.1. The main window

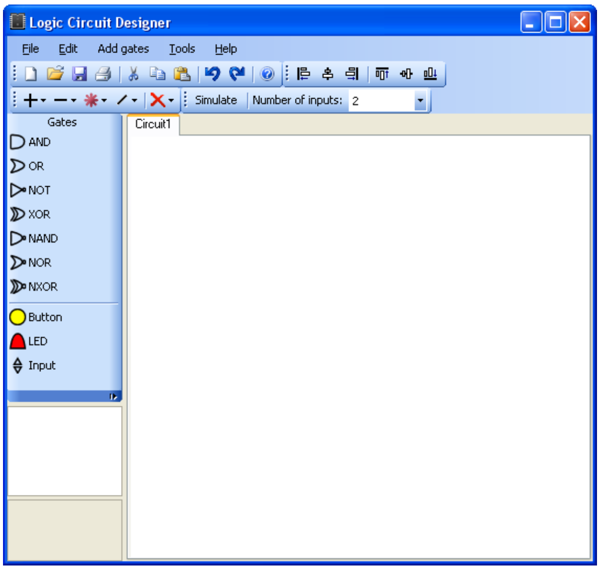


Figure 1

As you can see, in the main window we can find the tools we need to build the circuits. At the top there is a contextual menu: File, Edit, View, Selection, Align, Tools and Help. We won't go very deep into their description because the majority of them can be found in the menu just below and on the left side of the window.

Below the main menu, we can find a tool container represented in Figure 2 which is divided into 4 categories as follows:

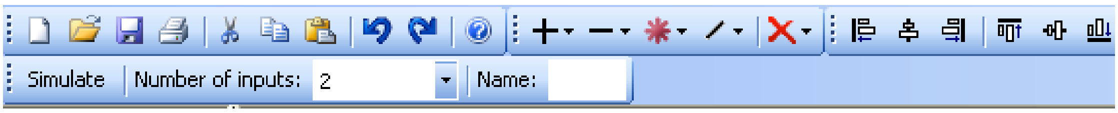


Figure 2

a) **Basic tools**: *New*, *Open*, *Save*, *Print*, *Cut*, *Copy*, *Paste*, *Undo*, *Redo*, *Help*;
b) **Selection tools**:
   - *Select*: only gates, only wires or both of them
   - *Deselect*: only gates, only wires or both of them
   - *Invert Selection*: inverts the selection of gates, wires or both of them
   - *Crop Selection*: only the selected components remain, the rest of them are deleted. It can be applied only to gates, wires or both of them
   - *Delete Selected Components*: can be applied like all the above ones
c) **Alignment menu** (applies only to the selected gates):
   - Align left sides: the components will be aligned to the left of the most left control
   - *Align vertical centers*: the components will be aligned at the middle of their vertical coordinate

- *Align right sides*: the components will be aligned to the right of the most right control
- *Align top edges*: the components will be aligned to the top of the most above control
- *Align horizontal centers*: the components will be aligned at the middle of their horizontal coordinate
- *Align bottom edges*: the components will be aligned to the bottom of the most below control

d) **Circuit control menu**: from this menu we are able to set the simulation engine of the circuit on or off. Also, we can choose how many inputs should any new created gate have (except for the NOT gate). When dealing with complicated circuits, we notice that every input and output has a meaning and we can easily forget what means each one. So a name to any input or output can be given to overcome this issue.

Considering the left side menu represented in Figure 3, this is where we choose the gates to add to a circuit. Just click on the one you want to add and it will appear on the right-top corner of the circuit.

Under the basic gates section we see the main logic gates needed in any circuit, then there are some more advanced ones including the inputs and outputs (we can also call them buttons and LEDs) and of course the modules which we will explain later.

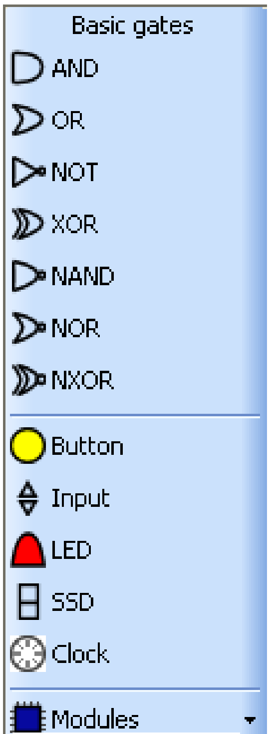


Figure 3

a) **Basic gates**
   - ***AND*** **:** the return value is 1 only when all of the inputs must are 1
   - ***OR*** **:** the return value is 1 if at least one of the inputs is 1
   - ***NOT*** **:** the return value is 1 if the input value is 0 and 0 if the input value is 1
   - ***XOR*** **:** the return value is 1 if the number of inputs which have 1 is odd and 0 otherwise
   - ***NAND*** : same as *AND* but its result is inverted
   - ***NOR*** **:** same as *OR* but its result is inverted
   - ***NXOR*** **:** same as *XOR* but its result is inverted

b) **Extended gates**

- ***BUTTON*** : this is a simple input. When simulating, its value can be changed by clicking it
- ***INPUT*** : this is more like a SSD (Seven Segments Display) input. Its hex value is converted to a binary form which can be used further in the circuit. Its value can be easily changed
- ***LED*** : this is the basic output of a circuit. If its value is 1 then the *LED* is lighted
- ***SSD***: this is and output which has 4 inputs representing the hex value it will show
- ***CLOCK***: the value on its output will be 1 for a certain period of time and 0 for another period of time. Both time periods can be set by the user

c) **Modules**: the modules package allows users to integrate a circuit in a more complicated circuit for as many times as they want without recreating it. This saves a lot of time and also increases efficiency. We will widely discuss about modules later.

3.2. Our first "Hello World!" circuit

We are going to create our first circuit, namely a circuit corresponding to a basic AND gate. It will be built using one gate, two leds and one output. As the first step we choose to create a new circuit. As the second step we click the AND button to add an AND gate on the circuit surface. Notice that the gate will appear on the top left corner of the circuit and we have to drag and drop it to the right place on the surface, like in Figure 5. You can move the new created gate anywhere you like and you can also rotate it with the mouse wheel after selecting it.

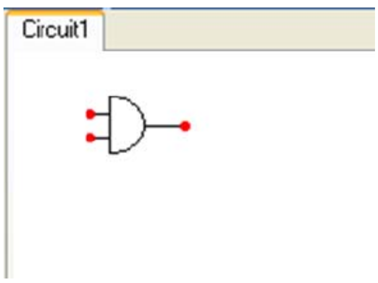


Figure 5

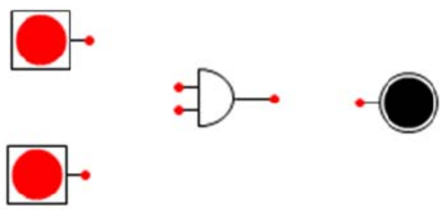


Figure 4

Now it's time to add the two inputs (buttons). To add two buttons click the *BUTTON* twice. The buttons will appear in the same place as the gate. Move them to a place where both of them can be seen. Any circuit must have an output, so add a *LED* to the circuit and move it to the right of the gate. The circuit must look like in Figure 4.

But what's the point of creating the gates, inputs and outputs if they are not linked together? For that we will use wires which connect the red dots. To create a wire you must move the mouse over the dot you need to link (the cursor changes to a cross), click the mouse and holding it move to the second dot which needs to be linked where you release the mouse button. What we need to do now is to link the buttons with one of the *AND* gate inputs and the output of the gate with the led. Now the circuit is looking like in Figure 6.

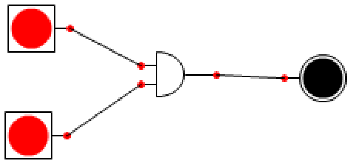


Figure 6

And this circuit is complete. It’s time now to simulate it. As you know, the *AND logic operation* returns the 1 value if both inputs are 1. To simulate the circuit click the *SIMULATE* button. Now the circuit enters the simulation state. By clicking an input you change its value affecting all the gates to which it is connected. Feel free to play with the simulation until you understand the logic of the *AND* gate but also the main purpose of this program. In Figure 7 we can see the simulation states:

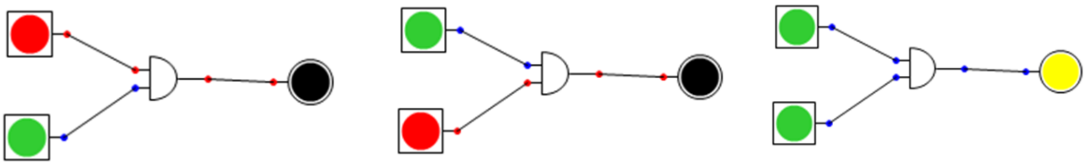


Figure 7

We notice that only in the third case when the inputs are 1 the value of the output is 1 and the LED is lit.

There is one more thing that should be explained. It’s about the links between the gates. Because of the increasing complexity of the circuits, the wires tend to be very hard to maintain and it becomes difficult to realize what wire is connected to a certain gate and so on. This is where the wire segmentation comes in. Any wire may be divided into as many segments as the user wishes. To create a segment for a wire, just click the middle mouse button on the respective wire and at the point you want the segment to be. This is a simple way to manage the wire connections. If you want to delete the segment, you can just middle-click the mouse again in the respective blue dot. Figure 8 shows us how the circuit may look if we play a little bit with the wire segmentation.

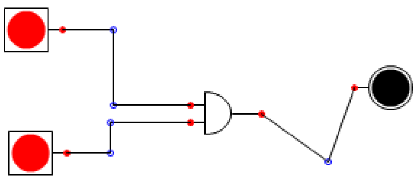


Figure 8

3.3. Building a parity circuit

Next, we will demonstrate the importance of the *XOR* gate. *XOR* gate is short for exclusive OR. This means that precisely one input must be 1 (true) for the output to be 1 (true). A way to remember XOR is "one or the other but not both". The XOR operation is a binary operation and is therefore defined only for two inputs. It is nevertheless common in electronic design to talk of "XOR-ing" three or more signals.

The most common interpretation of this usage is that the first two signals are fed into an XOR gate, and then the output of that gate is fed into a second XOR gate together with the third signal, and so on for any remaining signals. The result is a circuit that outputs a 1 when the number of 1s at its inputs is odd and a 0 when the number of incoming 1s is even. This makes it practically useful as a parity generator or a modulo-2 adder.

Certainly, there are many uses of this *XOR* gate. We will make a simple circuit that outputs 1 only when the number of 1 on the inputs is odd. The circuit has 3 inputs and one output. The main part of the circuit is represented by 2 *XOR* gates like in Figure 9. Also, we can build it using one *XOR* with 3 inputs, like in **Error! Reference source not found.**.

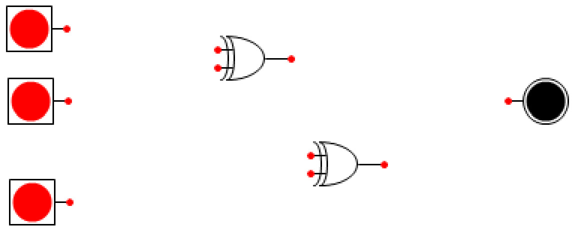


Figure 9

Until now we only added the gates, the inputs and outputs. Now it's time to link them: the first two buttons are linked to the inputs of the first *XOR*, and then the output of the first *XOR* is wired to the first input of the second one. The last button must be linked to the second output of the second *XOR* and finally we link the second's *XOR* output to the led. The next step is the simulation: we will notice that the LED is lit only if an odd number of the buttons are on. The final circuit should look like in Figure 11.

I also want to show how this circuit can be simplified. We can achieve the same behavior of the *XOR* gates by using only one. The difference is that it has 3 inputs, not only 2 like the other two had. The simplified circuit is shown in Figure 10.

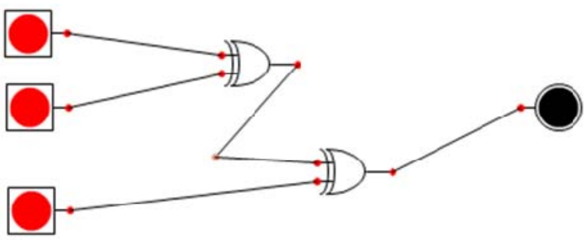


Figure 11

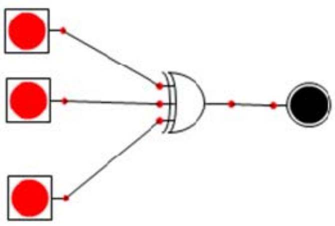


Figure 10

3.4. Building a modulo-2 adder circuit

Practically, a modulo-2 adder circuit adds 2 binary digits giving the result of this operation. Going deeper into the theory, we know that in binary system there are only two digits: 0 and 1. Of course, we can add them in order to obtain a result, but this operation's output isn't only the result but a special digit called *transport*. It is like when you add 9 and 1 and it results 10 [3]. The following table shows all the possibilities of adding two binary digits:

a	b	a+b		a_0	b_0	s_0	T_0
0	0	0		0	0	0	0
0	1	1	=>	0	1	1	0
1	0	1		1	0	1	0
1	1	10		1	1	0	1

We notice that the transport digit is 0 at most of the time. Only when we add 1 and 1 we get a 0 result but a 1 at the transport digit. The circuit that matches these tables is depicted in Figure 12.

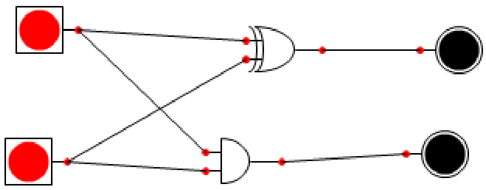


Figure 12

It is built using one *XOR* gate and one *AND* gate connected to the inputs and each one with its own output. It's easy to see that the first output is the result of the operation and the second one represents the transport digit.

4. Design and implementation

4.1. Tools used to build the application

This application was built in C# and the .NET Framework version 3.5. We used Microsoft Visual Studio 2008 to develop this application because of its ease of use, and also because we wanted to learn more about windows programming using C#. While working on this application we have learned many programming techniques and ways to improve our code. In the space of approximately two years we have written 3 versions of the application and we have also re-written it from scratch once, continually improving its performance and stability and also incorporating techniques and design patterns we have learned at school.

4.2. Application design

We have designed the application to be simple and flexible, being prepared for future expansion. The application is object-oriented, making use of features from the C# language. The implementation of the simulation is separated from the interface. This allows us to easily modify any of them without breaking the other.

4.3. The class hierarchy

The core of the application is structured into a hierarchy of classes. All the gates in the application are derived from an abstract class named *Gate*. It is used as a base allowing all the gates to be treated uniformly throughout the application, even though they have different implementations and functions. This base class contains some common properties that are common to all gates: the location of the gate, the angle at which the gates are rotated, etc. Also, this base class contains some functions that are common to all the gates, but which have different implementations for each gate: simulate (executes the simulation of the gate), draw (draws a representation of the gate), etc.

We can distinguish between two different types of gates: gates with variable number of inputs, and gates with a constant number of inputs. For the first class of gates we have derived a class from *Gate* called *BasicGate*. It is also an abstract base class but it adds functionality that helps with adding and removing inputs from a gate.

The basic gates (AND, OR, NOT, XOR, NAND, NOR, NXOR) are derived from the *BasicGate* abstract class. They implement the *simulate* and *draw* methods from the base class with their specific implementation.

The other gates (Button, LED, Input, SSD, Clock and Module) are gates that have a constant number of inputs and are derived directly from the abstract *Gate* class. They also implement the *simulate* and *draw* methods, and they also add functionality that is specific to their gate type. For example, the Clock gate adds functionality that captures mouse clicks and modifies its timing accordingly if the user has clicked the two arrows on the gate.

A diagram of the class hierarchy of the gates is presented in Figure 13.

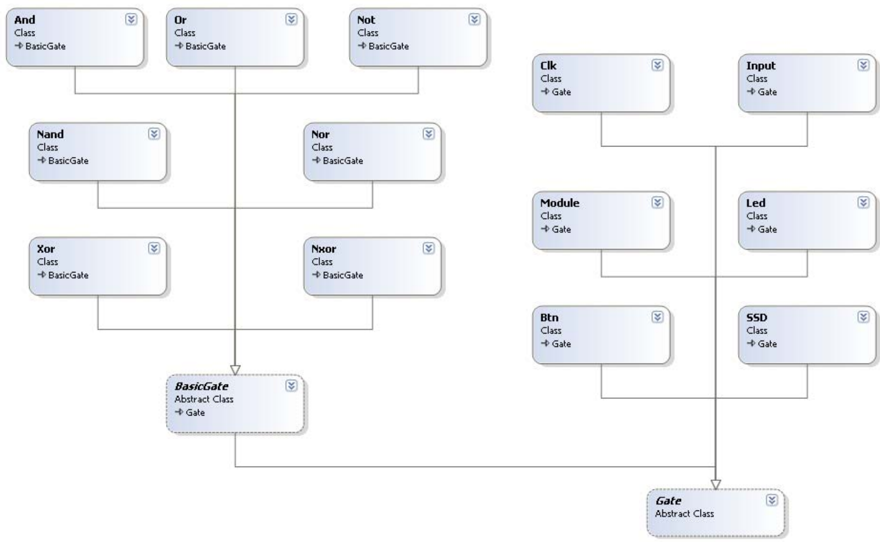


Figure 13

Using this coding style allows us to avoid repeating code and keeping to the DRY principle (don't repeat yourself). Also, this allows for cleaner and more flexible code.

4.4. The simulation algorithm

When simulating a logic circuit, the algorithm calls the *simulate* method for each gate that is present in the circuit. Each gate has its own mechanism for simulating. Before simulating, the gate consults the wires that are linked to the inputs of the gate and checks to see what value the output of the gate at the other end of the wire has. After that, it simulates the gate and sets the output of the gate to the simulated value.

For example, the simulation of the XOR gate is presented in the next code snippet:

```
public override void Simulate()
{
    foreach (Dot d in inputs)
    {
            // gets the value from the gate
            // on the other side of the wire connected to the input
            SetDotValue(d);
    }
    bool o = false;
    foreach (Dot d in inputs)
            o = o ^ d.Value;
    // set the output to the result value
    output.Value = o;
}
```

4.5. The drawing algorithm

The code that draws the XOR gate on the screen is presented in the following code snippet:

```
public override void Draw(Graphics g)
{
    //w - width; h - height; x - X coord; y - Y coord
    int w, h, x, y;
    int n = inputs.Count;
    x = Location.X; y = Location.Y;
    w = Width; h = Height;
    //Translate the origin and rotate the whole drawing context
    g.TranslateTransform(x, y);
    g.RotateTransform(Angle);
    //Draw the gate
    for (int i = 1; i <= n; i++)
            g.DrawLine(Pens.Black, new Point(0, 2 * i * 6 - 3), new Point(10, 2 - 3));
    g.DrawLine(Pens.Black, new Point(w - 18, h / 2 - 0), new Point(w - 1, h );
    g.DrawArc(Pens.Black, new Rectangle(1, 0, 10, h - 1), -90, 180);
    g.DrawArc(Pens.Black, new Rectangle(-3, 0, 10, h - 1), -90, 180);
    g.DrawArc(Pens.Black, new Rectangle(-19, 0, 50, h - 1), -90, 180);
    //Draw the input and output connectors
    foreach (Dot d in inputs) d.Draw(g);
    output.Draw(g);
    //Surround the gate with a rectangle if selected
    if (Selected) g.DrawRectangle(Pens.Blue, new Rectangle(0, 0, w, h));
    //Rollback the context transformations
    g.RotateTransform(-Angle);
    g.TranslateTransform(-x, -y);
}
```

5. Conclusions

5.1. Knowing how computes are created is a must

Even if you are a casual computer user, it is good to know some aspects regarding how computers work and how science was needed to create such wonderful machines. In case you are an advanced computer user, then knowing these aspects is crucial. Not only they help you understand them better, but also you can easier troubleshoot any problems you may encounter.

5.2. Two reasons why this program can help you

By using this program, we get very close to the science of the circuit building making us understanding them very well from the logical point of view, not the physical one. Our program is not just a tool for creating logic circuits, but is also a tool for simulating them. This simulation process can be used before any testing of a project, showing us what to expect from the final circuit.

The program was successfully tested by the first year students of the Faculty of Mathematics and Informatics, University of Transylvania, Braşov. It has been of great help for them to study the Computer Architecture discipline.

We have also made our application available for free on the Internet, and it has been downloaded over 25000 downloads. We have also received positive e-mail feedback from professors from the University of Maryland and the University of Texas who have come across our application on the Internet.

So that being said, we hope that the users will enjoy this program together with the capabilities and the facilities it has.

References

[1] Constantin Cucoş, *Pedagogie*, Editura Polirom, Bucureşti, 2006

[2] Jesse Liberty, Donald Xie, *Programming C# 3.0, Fifth Edition*, O'Reilly, 2007

[3] Anca Vasilescu, *Arhitectura calculatoarelor, suport de curs*, Universitatea Transilvania Braşov

[4] M.M. Mano, *Digital Logic and Computer Design*, Prentice-Hall of India Pvt Ltd, 2007

[5] ***, http://en.wikibooks.org/wiki/Digital_Circuits

RĂDOI ovidiu
Universitatea Transilvania
Facultatea de matematică-informatică
Str. Iuliu Maniu nr. 50
ROMÂNIA
oviradoi2003@yahoo.com

IVAN Andrei
Universitatea Transilvania
Facultatea de matematică-informatică
Str. Iuliu Maniu nr. 50
ROMÂNIA
antraxx_cs@yahoo.com

Anca VASILESCU
Universitatea Transilvania
Facultatea de matematică-informatică
Str. Iuliu Maniu nr. 50
ROMÂNIA
vasilex@unitbv.ro

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

BPMN-EPC-BPMN Converter

Vladimir Kotsev
Teacher Coordinator: Ivan Stanev

Abstract

Creating business process models is a major task of enterprise architecture. There are good tools for it available. World standards change rapidly, but some of the old tools are still good and many people are used to them.

In this paper we introduce our point of view about business process modeling notations, their advantages and disadvantages. We share the results of our discussions and researches about creating a converter between tools that are using different notations.

1 Introduction

Business process modeling has become a significant part of enterprise management during the last decade. Nowadays' business modeling aims at the integration of the partial models that represent particular views on an enterprise. This means not only those models of distinctive and important parts of the enterprise should be created, but also that semantic relationships between partial models can be expressed.

The main idea of business modeling is to help people gain better understanding of complex systems by providing different appropriate abstractions. Therefore, a corresponding modeling language is used based on specific terminology and notation that is common within particular view. The best way to show the common situation is diagrams. This makes business processes and relationships between them easier to understand by wider and different circles of people within an organization.

One of today's most powerful mean of business process modeling is the Business Process Modeling Notation (BPMN). It is a graphical notation for expressing business processes in a Business Process Diagram (BPD). The objective of BPMN is to support business process management by providing a notation that is intuitive to business users yet able to represent complex process semantics.

Naturally, BPMN is not the one and only notation and it also has its disadvantages. According to most researches found on the Internet the most significant drawback of BPMN is the lack of good linking to other elements of the enterprise architecture. And here people turn back to the historically older Event-Driven Process Chains (EPC) that can strengthen this aspect of the business process modeling. Fig. 1 describes the abilities of BPMN and EPC to link different dimensions.

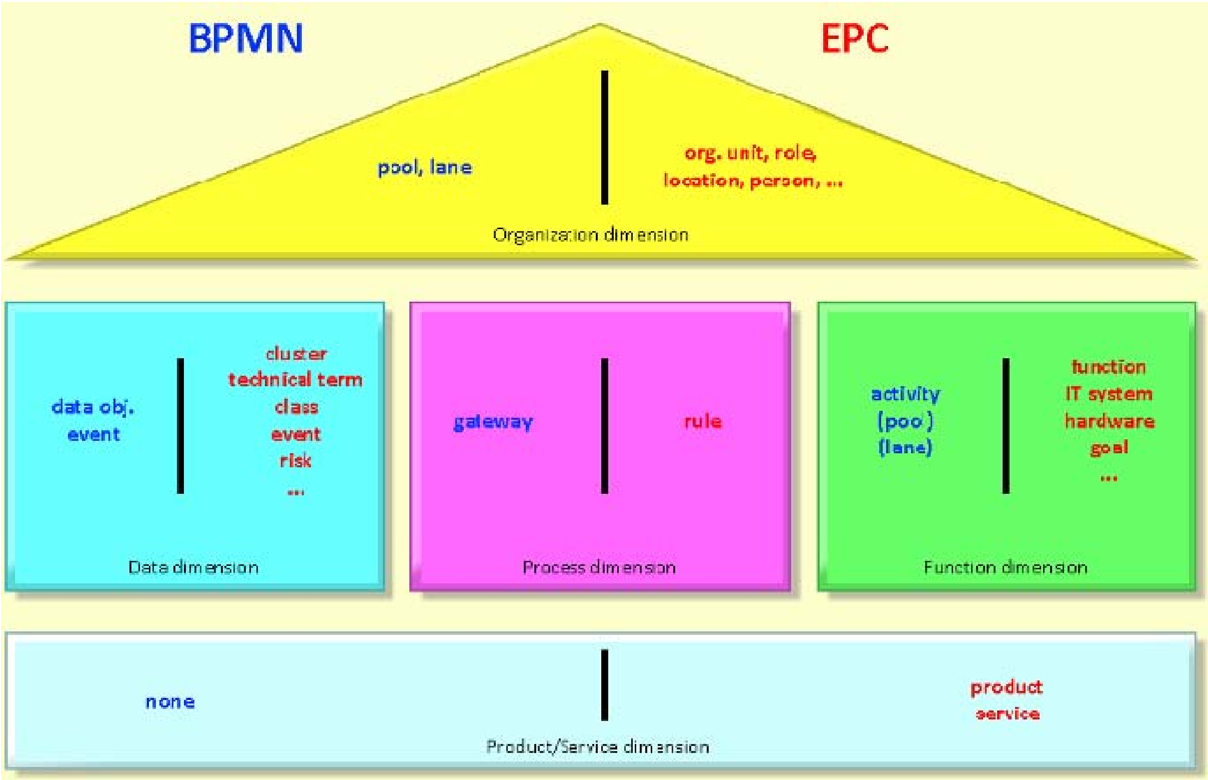


Fig. 1

In fact, EPC and BPMN can do a great job, if used together. The reason behind this is that each of them can compensate the disadvantage of the other. In this paper we show how we try and hopefully manage to combine both EPC and BPMN to achieve our goals.

2 Definition of the problem

Since EPC was used in business process modeling for a longer period of time than BPMN, there is a vast amount of tools that are based on the EPC standards. One such tool we worked on is a Business Process Generator (BPGen-EPC) which uses EPC rules to link different tasks and processes from a database and help us generate business processes. On the other hand UML and BPMN are getting stronger and are considered as the world leading standard in the sphere of modeling as a whole.

In order to keep using our business process generator BPGen-EPC but show our business process diagrams pursuant to the newest standards and requirements we need to link our tool to some UML based modeling environment. Aiming for a continuous period of usage we have to choose such that is being updated frequently enough to follow the world standards. These requirements made us have a closer look on and eventually select Sparx Systems Enterprise

Architect. To make this linking possible it is necessary to create a converter and use it as a bridge between the tools.

Creating such a tool that converts the business process and its elements from a diagram and into a database can be used not only in our particular case.

Database models of business structures are usually hidden and the user does not access them directly, but rather through an interface on the software tools for capturing and analyzing the models. In many cases the database design plays an important role for making possible the kinds of analysis that the business designer wishes to perform on the model. If the designer is authorized to have knowledge of the structure of the business model in the database, more efficient understanding, implementation and support may be possible. For example, this can allow the designer to structure the representation in such a way to facilitate faster analysis or higher degree of reuse of business process templates [1].

When the business designer needs to deal with only a small part of business model working with its database model is very convenient. There are a number of aspects to the enterprise, each giving a different view on it, presented by corresponding business structure.

3 BPGen-EPC Principles and Software Architecture

Most developers of business process modeling environments try to provide the users with a variety of tools that can be helpful for making modeling easier and more descriptive. For administrative processes and others, that tend to be repeatable without being changed frequently, it is enough to create a model that can be used multiple times.

However, if we think about processes in the industrial branch things are a bit different. Those processes are very dynamic because they are being modified frequently. The users need not only a toolset that allows them to change the business model, but one that can also automatically generate new processes or edit the existing ones using the already available information. For this purpose the possibility of the following modifications should be provided:

- Inserting and editing data about processes (process properties and additional useful information);

- Storing data about different versions of a process;

- Inserting and editing data about the connections between processes and their sub-processes;

- New process generating with information about its sub-processes information available;

- Editing of existing process, which includes sub-process adding and removing;

- Creating a potentially executable business model;

- Defining the sub-processes of a process;
- Printing the data about a process as an aggregation of sub-processes on all levels;
- Printing the data about a process as a chain of sub-processes.

These are the main principles of BPGen-EPC tool. Its software architecture and interfaces are shown on Fig. 2. As you can see it is connected to a database where all the information is stored. The messages server is responsible for the communication with the BPGen-XML converter.

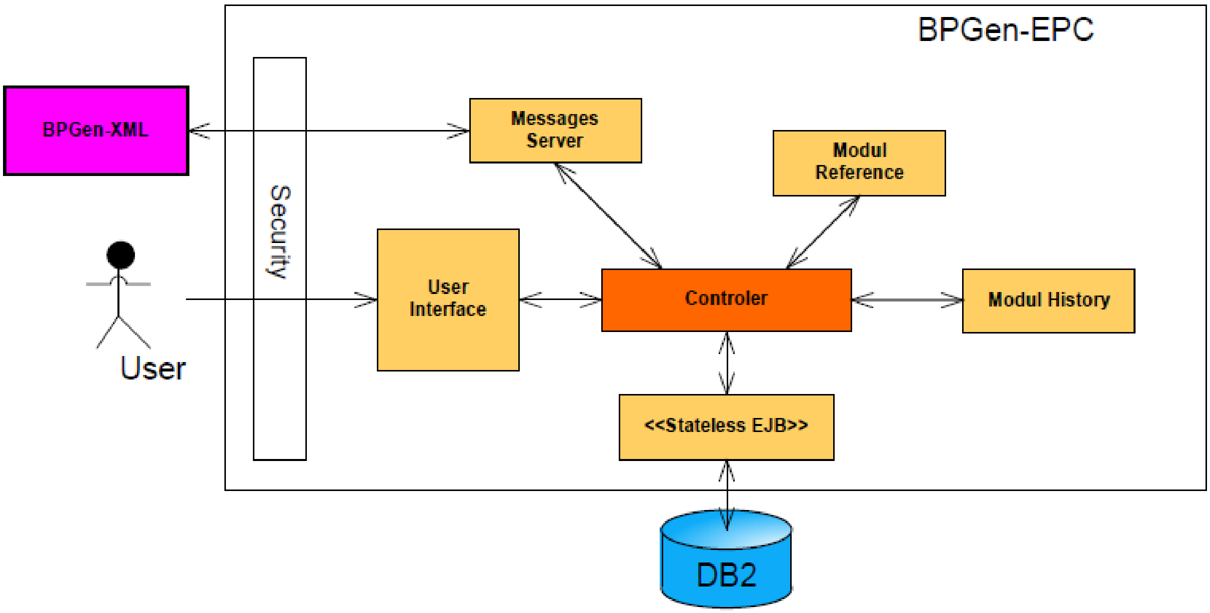


Fig. 2

4 BPMN-EPC Conversion Rules

In order to convert the information about the business processes from BPMN and translate it to information suitable to fill it in an EPC database we meet the necessity to map EPC to BPMN notations first. Once mapped, the elements of both notations can be translated in either direction.

Direct mapping is the best solution for translating one of the notations to another but there are still some minor problems. According to most researches on the Internet it is almost impossible to have a universal mapping between EPC and BPMN because of some differences of their nature. It is recommended that the direct mapping is done only for the core of the notations. The rules of the mapping are described by Willi Tscheschner [2]. Fig. 3 provides the translation between the notations.

In EPC a function is described as an activity which supports the completion of a business objective. It is semantically a process rule for transforming an input system state to a following output system state. In BPMN an activity is a generic term for work that a company performs [3].

A task is specified as an atomic activity which cannot be broken down to several activities. With both definitions a function in EPC is mapped to a task in BPMN.

A process model, described in EPC, is determined by one or more events (an explicit entrance state of this process). Similar to this, a process model in BPMN is also implicitly or explicitly started through a start event. Out of this, an event in EPC which has no incoming control flows can be mapped to a start event in BPMN.

Process models defined in EPC are terminated by events (an explicit end state of this process). In BPMN a process model is also implicit or explicit terminated through an end event. With this, an event in EPC which has no outgoing control flows can be mapped to an end event in BPMN.

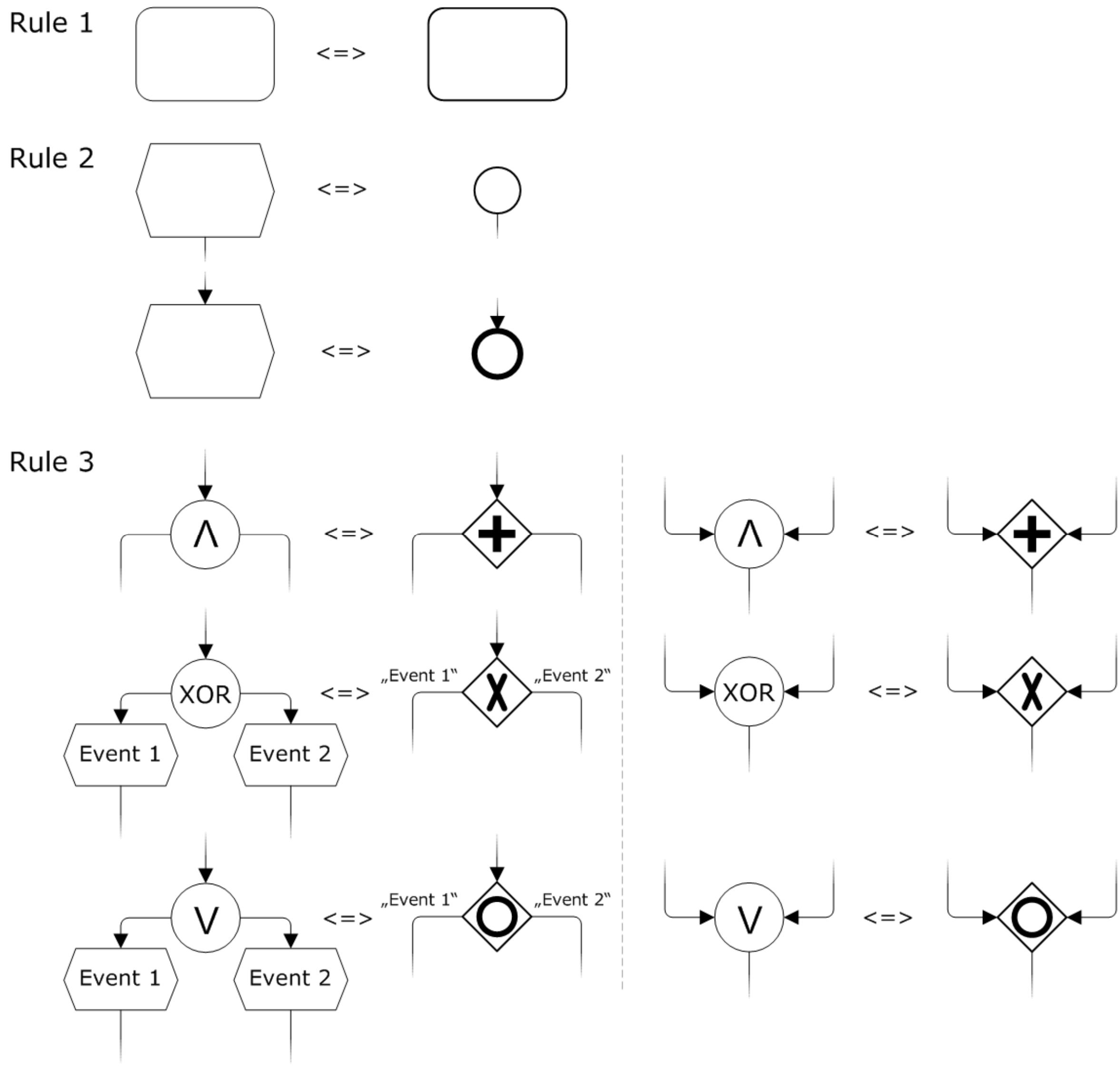


Fig. 3

Connectors (xor-connector, and-connector, and or-connector) in EPC are defined as concatenation points in the process for events and functions [4]. Gateways in BPMN are defined for controlling the divergence and convergence of the sequence flow [3]. With this, the several connectors can be mapped to the same gateway (xor-connector to data-based exclusive gateway, and-connector to parallel gateway, and or-connector to inclusive gateway) independent from the splitting or joining behavior.

In addition, if a xor-connector or an or-connector occurs which has splitting behavior (by concatenation of events) the process flow is affected by the states of the following events. Depending on the previous function and the output situation, this event gets chosen which represents the appropriated state. Similar to this, in BPMN the process flow after a gateway is determined by the condition of the following sequence flow. Therewith, those events in EPC can be mapped to the condition expression of the following sequence flows.

In future versions we desire to go even further and make the converter capable to translate the full sets of each notation to the other. However, it will be a hard task because every complex diagram can use some of the elements for its own purposes that don't obey some common regulation. It is still believed that making the conversion on a deeper level is risky and should be decided for each business process model diagram.

5 BPGen-XML Software Architecture

As we have the BPGen-EPC and Enterprise Architect ready to use, we also need a connection between them. We have already decided the conversion rules between BPMN and EPC elements. Since Enterprise Architect provides a diagram exporting to an XML file, we can use it as a bridge between the environments.

Our converter can receive the exported XML file of a diagram and extract the useful information. This data can directly be inserted into the BPGen-EPC database, while a backup copy of the XML file shall be saved in order to keep the information about the tasks and events that may be useful for someone else.

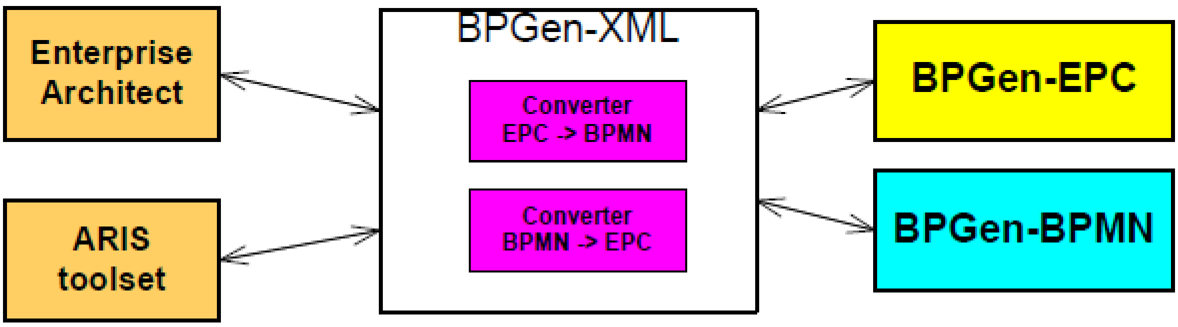


Fig. 4

After the business process model is ready the converter receives an appropriate message from BPGen-EPC Messages Server and is allowed to access the database and extract the needed information to build the BMPN diagram up. The software architecture of BPGen-XML is shown on Fig. 4. As you can see we've planned to create another business process generator for BPMN and possibly include ARIS Toolset in our business process environments, thus making it possible to solve different tasks without worrying about what kind of notation or element linking the businesses need.

6 Conclusions

After a short research of business process modeling principles we managed to create a tool to serve our and hopefully others' needs. We designed and created a converter that will let us use our existing business process generator that works with Event-Driven Process Chains in combination with a powerful tool such as Sparx Systems Enterprise Architect. The converter successfully translates data from a business process model XML file and provides it for modification to the business process generator. After the linking task is done and the business process is ready the converter translates the information back to the XML file.

Future plan is to enhance the converter and make it possible to work with EPC environment (such as ARIS Toolset) and a business process generator that works under the rules and conditions of BPMN.

References

[1] Grigorova K., P. Hristova, G. Atanasova and J. Q. Trelewicz, *Levels of Business Structures Representation*, IBM Research Report, RJ10349, 2005.
[2] Tscheschner W., *Transformation from EPC to BPMN*. Oryx Research, Potsdam, Germany, 2010.
[3] Object Management Group (OMG), Business Process Modeling Notation. *V1.1. Technical report,* January 2008.
[4] G. Keller, M. Nüttgens and A. W. Scheer. Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)". *Technical Report 89,* Institut für Wirtschaftsinformatik Saarbrücken, Saarbrücken, Germany, 1992.

VLADIMIR KOTSEV
Ruse University
Department of Informatics and Information Technologies
8 Studentska Str., Ruse
BULGARIA
E-mail: vladimir.kotsev@gmail.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

E-Marketing Service Oriented Architecture

Valentina Lazar
Teacher Coordinator: Lect. PhD. Florin Stoica
Lect. Ralf Fabian

ABSTRACT

Internet marketing is considered to be broad in scope because it not only refers to marketing on the Internet, but also includes marketing done via e-mail and wireless media. Digital customer data and electronic customer relationship management systems are also often grouped together under internet marketing.

Internet marketing ties together the creative and technical aspects of the Internet, including design, development, advertising, and sales. Internet marketing also refers to the placement of media along many different stages of the customer engagement cycle through search engine marketing (SEM), search engine optimization (SEO), banner ads on specific websites.

System architecture is known as marketing analysis and it is based on customers and building on the data submitted to the appropriate strategies.

1 Introduction

Given that we live in a world in constant motion, a world where time is the main problem, the present application is built precisely as an alternative to this problem in the banking field. It is known that in banks, as large firms, the size of the database is a reasonable size, therefore, browsing and extracting data representing their interests is a problem. Thus, the present application comes with the following improvements: not only that the data are strictly filtered to obtain what it seeks, in this case that speaks about marketing on the Internet (E-Marketing), data are taken very on the market where the company operates, generating a double advantage: market knowledge and improvement of the marketing plan.

The first approach to bank marketing evolution is performed by Philip Kotler. It involves five steps, which were identified according to the diversity and effectiveness of marketing techniques implemented.

- The first stage is designed as a bank marketing technique of advertising and sales promotion, image banks (providing small gifts, but useful: umbrellas, flashlights, pens.). The emphasis is on friendly atmosphere within the bank.

- In the second phase of banking marketing is defined as relaxed smile - to stop officers are trained how to have a more open attitude towards the customers. Is rearranged so that the interior banks to be more pleasant atmosphere and the outside is "humanized" Benevolence has ceased to be the deciding factor in choosing a bank.

- In the third stage of development, marketing segmentation and upgrading banking means. Banks have discovered a new competitive tool when they began to segment their markets and develop new products for each target market segment.

So what keeps the bank's ability to develop new products continuously determines the position of market leader. Marketing has entered the insurance and investment services. Innovations have resulted in widening globalization of financial markets and banking business through the provision of less traditional services and the use of information technologies and computer networks.
At the fourth stage, marketing means bank positioning. Banks realize that no commercial bank is unable to provide all products and be the best for all customers. It concludes that a bank must "occupy a specific position, taking account of its possibilities.

Currently, the increasing role of bank marketing and organizational strategic role of analysis, planning and monitoring of banks, based on actual usage of information technologies. Banks are developing policies to attract and maintain clients, which involves evaluating the "cost" of losing customer orientation to new customer segments, moving to a personalized approach to customers and expansion into new markets, develop new products and services, use of distribution channels virtual sale of banking services on-line.
In international practice, basing the content of bank marketing was based on research findings conducted by the major industrial and commercial companies.
Also marketing a bank is the bank's management system, which involves the recording and studying the processes taking place in the capital market as a whole, and distinct sectors: the banking sphere, the credit system and securities market.

The main factor was the emergence of a significant increase in bank marketing in developed countries and consequently on the financial sector bank. Although marketing has emerged in close connection with the sale of tangible goods in recent years has shifted, as the service sector has risen significantly, than other areas of international economies. Accelerating growth of financial services and banking has led to the need for customization of economic subjects such as marketing and management specifically dedicated to this field.
Uneltel banking sector growth was not the only factor stimulating the development of marketing. Currently in the Conditions; costs increase, productivity stagnating and quality of service tends to deteriorate, an increasing number of large firms to marketimg apelează. Disruptions in services, industrial, financial and telecommunications have brought new competitors internţională market, overcapacity and intensified price competition in the field. This competition has played undoubtedly dezvoltatrea an important role in bank marketing.

2 Submission of application. What's the issue?

The application seeks marketing process when the customer comes into contact with the environment and not just marketing, and until the construction of new marketing strategies that allow customers to increase sales and loyalty, qualities essential to the existence and profitability of business activity, that is a virtual bank.
This starts from the idea of building a second apliatii which are interrelated: one to retrieve data from other clients and to help in developing future strategy.

GATHERING INFORMATION

It starts with the realization utilizatoril interface, and more specifically the Web Site. The first point is to be sought for interface design.

Traditionally, research into human-computer interaction started from information theory and from the knowledge. Main purpose of this interface is to understand and represent the interaction of man-calclator information and knowledge transfer perspective.
To motivate customers to view those presented in the site is necessary, from choosing colors presentation to the selection of information to be submitted.
Such colors were chosen to be not tired eyes (blue, gray and white) and the information was compressed and the essence to avoid loss of interest when making the client aware of the set. Another way to attract customers who are prezantate is interactivity and data that is not much effort on the part of clients to reach desired information.
This presentation is especially necessary if you are talking about a world in constant motion in time is most valuable.

In terms of marketing, creating and developing a web site must have the support but needs the existence of specific, concrete, mostly related to the way marketing communication that takes place the bank. It is important to note that in general, it does not create products, does not set prices, do not send products to the recipient, not the bank's reputation and do not double sales tripled.
The website can contribute significantly to the manner in which the organization realizes the exchange of information and marketing environment, mainly with clients (actual and potential) and its competitors.

- In the first case is about providing a volume of information and attractive enough to cause a favorable reaction to customers: marketing buying or ordering products (loans) promoted or at least call for additional information about them.
- In the second case is about differentiating the bank in relation to its competitors, in general and especially in terms of marketing communication.

The bank's Web site should be perceived as a tool for online marketing, strategic nature, its. It can not be created to respond to needs of the moment, to limit or eliminate some bad tactical situations.
Involves the creation of the bank, clear commitment to:

1. integration and its use in marketing communication to the bank;
2. allocation of specific resources (technology, money, people and time) to update and expand its content.

Here are some questions whose answers may guide the bank in its attempt to create a web site:

1. Is the market large enough in terms of its coverage area and to assume the Internet?

To what extent consumer / user potentially has access to the Internet and, respectively, use it to obtain information or to purchase different loans?

2. Is the market large enough in terms of its ability to promote and ensure profitability, possibly via the Internet distribution of bank loans?

3. To what extent products (loans) to be created for the site can be promoted properly and using the Internet as a tool specific Web site?

5. What will be the overall effect on the lending process by creating web site?

6. What will be the overall effect on the bank's image by creating web site?

Before the user to have access to better products marketing is that it is hereby notified to the legislation.

Interface provides access from the first page of this information, and the user to view what you are looking for.

Given the desired efficiency of customer-bank relationship is necessary to work out a framework for client consulting.

Interface in question features a discussion forum where customers can express their concerns or questioning about the work or services over the bank.

To establish the terms of bank-client discussions are urilizate forms as methods of collecting such information.

The form will ensure the construction of Web pages that allow users to effectively introduce information and to send them serverului.El can vary from a simple text box for entering a string as a search key - a characteristic element of all engines cautaredin Web - up to a complex structure with multiple sections, which offers powerful features for data transmission.

A session with a Web page containing a form includes the following steps:

1. The user completes the form and we expedieaza a server.

2. A dedicated application on the server analyzes the completed form and (if necessary) the data stored in a database.

3. If necessary expedieaza server user response.

Thus, in order to benefit from products offered by Virtual Bank is the choice of specific form required for the loan in question.

These fields allow the client to obtain information about which can be exploited and establish a market segment which addresses Bank.

To avoid any confusion or ambiguity, the forms are explicitly constructed and require more data Deat classical, although we recommend using the shortest forms.

RECOVERY INFORMATION.

Because we wanted to obtain an interface more "friendly" user-marketing specialist, was chosen for its construction using the Java programming language.

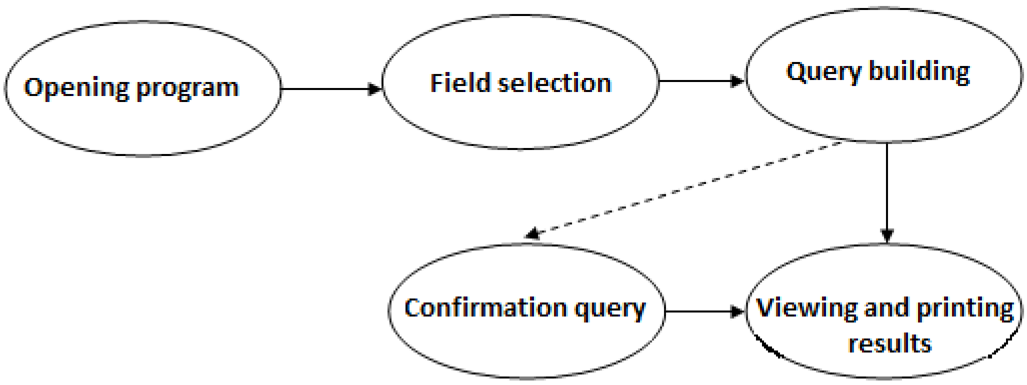
This program allows linbaj objects and user-friendly interactive graphic.

It is preferable that applications be created using Java Swing technology because it offers a far wider range of facilities, but will not completely abandon the basic AWT classes as there are re-used in Swing.

In principle, the creation of graphical applications include the following things:
• Design
- Create a display area (such as a window) that will be placed graphic objects (components), to communicate with the user (buttons, text editing controls, lists, etc.);
- Creating and placing components on the display surface to corresponding positions;

• Functionality
-Definition of actions that must be executed when the user interacts with application specific graphic objects;
- "Listen" event generated by objects when user interaction and execution of appropriate actions, as they have been defined.

The functionality of this interface refers to the opportunity of marketing reports following certain queries in the database.
As a result, to be mentioned of how to obtain these rapoatre with an interactive design.
This starts from the idea of market segment knowledge by knowing its characteristics: the size of this segment, the physical market place, etc..
To know this requires a market study and that study is better than on their own resources.
Therefore it is intended to achieve some queries in the database in which customer data are recorded.
This process is the result of a program in Java programming language that allows data after some queries.
Thus was born the MDC that is Developer Marketing Strategies.
Schematically, the stages of this system are:



The apparent ease of use and application that the user is assisted at every step to avoid mistakes in the following queries.

The core of any marketing information system to be the central repository of data. They contain solid information on all customers and potential customers. For the sake of brevity, this paper uses the generic term for a specific site in the database are stored informatiile.În addition, to gain maximum advantage from the exploitation of such information is necessary for each system to provide data verified and documented on the markets and target customers.

Applications for marketing uses the information in your customer database, but there are some typical functions of this system:

- simultaneous queries-making in several tables, queries made after a certain number of orders placed in the database;
- tabular display of the result queries, offering the possibility of saving-results;

The main advantage of the reviews is that offers an overview of the business relationship with a client or customer group. This overview allows a deeper understanding of consumer behavior and market in general, providing a starting point, among others, in marketing research, especially in studying the market.

The main feature of the system lies in the possibility of handling large volumes of information from various sources, internal and external, in order to develop models of market segments or business conduct. Analysis reports can be quickly generated, shaped and saved for future use, and business plans and scenarios - refined and exquisite.

An analysis system must have three basic functions: the ability to absorb information, the ability to manipulate and analyze the information according to user's desire and ability to report the results of all tests.

The most frequent problems related to the implementation and use of market analysis systems are:

- choosing the right information is crucial to the ultimate success of the system;
- if users are not fully familiar with the related data structures, they may draw some erroneous or lacking in content analysis;
- possible, the system functions should be developed so that independent information and data structures underlying the system.

The main advantage of market analysis applications is the increased ability of users to understand in detail the market is operating. Thereafter, the information conveyed through the system can be used to assist policy development process of the market and range of products and services, both activities affecting the performance of the organization in general.

3 Conclusions. Assumptions of future development.

The main problems were identified in the bank market are:

- lack of financial resources for supporting lending;
- reaprecierea risk in financial markets;
- lack of subjectivity in the banking institutions on the evolution of power market and adapting to this trend;
- Bank wish to raise the profile marketing analysis reports;
- increase in loans as a result of applying a proper stratefii segment that led to the reduction of eligible clients;

For successful operation of commercial banks requires a complex system of marketing and the creation of services and special sections. They are employed in developing marketing plans and programs and, of course, in studying the experience to take over its market by the bank.

The main application data that exploit them are among the most objective, consisting of personal data in case of individuals, and data identified for legal persons, the forms they filled in preparation for credits.
These data are helpful to know the market segment which the Bank is addressing, and how to attract and customer loyalty. This is accomplished by knowing the market characteristics (ie size regardless of the number of customers that speaks fizie or legal persons, the degree of urban development and rural knowledge obtained through the address, for example, which type of companies best suited Bank offer, obtained by legal form of recorded knowledge, etc..) and thus it can be so tender to attract the loyalty of current customers and potential customers.

Still using this marketing system can be set, for example, price and promotion strategies, namely an increase in bank profitability by increasing interest rates so as not to diminish the number of customers via a promotion in the proper direction (achieved in promoting principle by specifying a cutting preioade advantageous for the client, representing a percentage rate acceptable credit, etc..). As prerequisites for future dezboltarii, we aimed to introduce an agent-oriented system that allows greater ease with the different types of queries, taking into account existing data. This type of system automatically performs the user expects the results to each update data, hence the feature intelligent system.

References

[1] **Michael E. Porter**, *Competitive Strategy: A manual for survival and growth of firms in market economy conditions* , Teora, Bucharest ,2007

[2] **Michael E. Porter**, *Competitive advantage: A manual for survival and growth of firms in market economy conditions*, Teora, Bucharest ,2007

[3] **Cetină I.**, *Banking and Financial Marketing*, Economic Publishing House, Bucharesti, 2005
[4] **Dumitru, L.**, *Banking and Financial Marketing*, Dimitrie Cantemir University Press, Târgu-Mureş, 2001
[5] **Gheorghiu, L. ,** *Ways of promoting banking services in Romania, marketing view*, ASE Bucharesti, 2003
[6] **Odobescu E.**, *Marketing of modern banking*, Sigma House, Bucharest, 2007

[7] **Kotler Philip, Armstrong Gary**. *Principles of Marketing.* Ed. a 3-a. Bucharesti: Teora, 2004.

[8] **Cetina Iuliana, Odobescu Emanuel**, *Bank marketing strategy*, Economic Publishing House, Bucharest, 2007.

[9] **Matt Haig**, *E-marketing guide*, “Rentrop & Straton” Publishing House, 2005

[10] **Gay R, Charlesworth A., Esen R.,** *Online marketing, customer-oriented approach,* “All” Publishing House, 2009

[11] **Strauss J., El-Ansary A., Frost R.,** *E-Marketing*, Pearson Education, Inc Upper Saddle River, 2003

[12] **Ioan-Gheorghe Ratiu, Nicoleta David**, *Working with databases in Java,* „George Bariţiu“ University, Braşov

[13] **David Powers**, *PHP Solutions.Dynamic Web Design Made Easy*.pdf

[14] **Brad Bulger, Jay Greenspan, David Wall**, *Mysql/Php Database Applications***,** 2Nd Ed - 2004 - Wiley.pdf

[15] **David M. Geary**, *Graphics Java 2 - Mastering the JFC*, 3^{RD} Edition, vol. II, Swing
[16] **Jason Hunter**, (E-Book) Java Servlet Programming

[17] **Andy Harrys**, *PHP 6-MySQL Programming for the Absolute Beginner*.pdf

[18] **Tricia Ballad, William Ballad**, *Securing PHP Web Applications*.pdf

[19] **Mircea Neamtu**, *Distributed Programming Course*, ”Lucian Blaga”,University of Sibiu, Science Faculty 2010

[20] **Ralf Fabian**, *Course of Graphics and Image Precessing*, ”Lucian Blaga”,University of Sibiu, Science Faculty 2010

[21] **E-marketing** http://en.wikipedia.org/wiki/Internet_marketing

[22] **Marketing Environment** http://www.studentie.ro/Cursuri/Marketing/Curs-Marketing-MEDIUL-DE-MARKETING_i47_c1038_55121.html

[23] **E-Marketing Strategies** http://cerex.unitbv.ro/teze/rezumate/2010/rom/DajAlexisIon.pdf

[24] **E-Marketing Plan** , http://www.dotcom.ro/imarket.php

[25] **Competition Analysis,** http://www.scribd.com/doc/23835544/analiza-concuren%C5%A3ei

[26] **Competitive Strategies** http://www.preferatele.com/docs/marketing/4/analiza-strategica-a1.php

[27] **Sales optimization techniques**, http://www.newschannel.ro/categorii/vanzari-marketing/4/

[28] **About safe online transactions** http://www.talcioc.net/secure.php

[29] **Advantages and disadvantages of PHP** , http://kitsoftware.ro/Item/Framework-ul-PHP-care-sunt-avantajele-si-care-sunt-dezavantajele-lui/Section/Dezvoltare

[30] **Working with the SQL database ,** http://www.ub-filosofie.ro/~solcan/minf/d06_07/minf.pdf#page=135

[31] **About Web Technology,** http://tehnologii-web.ro/

[32] **About JavaScript,** http://en.wikipedia.org/wiki/JavaScript

[33] **About CSS,** http://en.wikipedia.org/wiki/Css

[34] **About HTML,** http://www.w3schools.com/html/default.asp

[35] **Working with databases in Java** http://software.ucv.ro/~aion/nad/jdbc.pdf

[36] **PHP Web Security,** http://www.slideshare.net/calin.iepure/securitatea-web-in-php-presentation

[37] **Agent-oriented systems. Agent oriented analysis and design applications.** http://profs.info.uaic.ro/~busaco/publications/articles/agenti2.pdf

[38] **Florin Stoica**, *Agent-oriented Systems Course*, ”Lucian Blaga”,University of Sibiu, Science Faculty 2011

Valentina Lazar
”LUCIAN BLAGA” UNIVERSITY OF SIBIU
FACULTY OF SCIENCES
COMPUTER DEPARTMENT
Adress: No. 5-7, Dr. Ion Ratiu st., 550012 Sibiu, Romania
E-mail: lyah_valy@yahoo.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Virtual PC – solution for virtual shops

Manta Alin, Ignisca Madalin, Ciorogar Ciprian
Coordinator: Prof. univ. dr. Dana Simian

Abstract

The purpose of the article is to present an application used for an online shop specialized in computer hardware, its most important feature being the promotion of the application used to run the website and not the product commercialization. This application can be successfully used for any product category. HTML, CSS, PHP, MySQL, AJAX, are open source web 2.0 technologies that have been used to build our application.

Keywords: VirtualPC, application, online shop, online trade, web 2.0 technologies

1. Introduction

Since the beginning of trade, human kind has tried to find better ways to trade, to buy and sell larger quantities spending as little time as possible. This process has been constantly improved along with the development of technology and the “online trade era” emerged once the internet was a common part of our lives.

During the last years, technology had an exponential evolution – the internet era brought in the foreground the commercial branch, online catalogs as well as online shops emerged thus enabling the small and medium size merchants to improve their business and maximize their profits. Those who chose to transpose their business on the internet, gained a lot, since a relatively small investment led to larger sales and revenue.

Case studies made on the commercial development of businesses point out their dependence on the spot where the sales point is located. The removal of this dependence can be easily accomplished by using the online trade. Other advantages of the online trade are for instance, the decrease of expenses through lowering the number of employees as well as through lowering the number of the physical selling points – thus reducing the running charges and operational costs.

Closing a selling point and opening an online shop would mean an increase in the number of clients and that the business would not be dependent on the clients that live in a certain radius with the origin in the location of the shop anymore. At first, the online shop would not bring a lot of customers of course - especially if the business doesn't have a well known name of brand – but with minor investments in advertising that could be changed and the site would bring a lot more potential customers than the shop had before.

In short, what we are trying to emphasize, is the high numbers of clients that an online shop could have, if the business would also develop on the internet - let alone the low maintenance costs - the success or failure being still dependent on a good or bad management of the business.

E-commerce was not limited to finished products and new businesses that operate exclusively on the internet have emerged. Many companies have begun selling services, information or digital products such as books, pictures, software, and so on.

Under these circumstances, the present paper wants to introduce an application used for an online shop. This application has already been developed and is present on the internet and can be accessed at the following web-address: http://www.virtualpc.ro.

This paper is structured as follows: in section 2, we make a small presentation of the application, bringing out the used technologies and then a detailed presentation of the actual application follows. Section 3 is dedicated to drawing of conclusions and pointing out the advantages of the "VirtualPC" application as well as future plans and tendencies of our team.

2. Main results. VirtualPC application

2.1 Application technologies

An online shop is implemented by means of a website that is administrated by the company that expands its business over the internet, which wants to sell its products online.

It is a well known fact that a website is a group of web pages (with rich text content, images, animations, multimedia, etc.) accessible to any user, built on a certain theme and interconnected through so called hyperlinks. These websites can be built by a company, a freelancer, public institutions, actually by everyone who has the necessary knowledge in design and web programming.

To begin a web-based business it is necessary to have: a domain name of the virtual shop, web hosting, the actual website and content for the end user.

In our case, the purchased domain is: www.virtualpc.ro, the information has been hosted on a Romanian server, and the website has been built by our team consisting of the project manger /

web designer (Manta Alin) and two programmers/developers (Ignisca Madalin and Ciorogar Ciprian.)

An online shop should answer to a few objective requests, should have a well defined purpose, should be accessible to a target public and it should make itself remarkable through attractive, simple design (because in this area of business, image counts). The site access should be as quick as possible. An important feature of an e-shop is security - user data security (user confidentiality) as well as transaction security (banking).

The technologies used in the development of an e-shop are chosen by the developers – we chose to use the following: XHTML, CSS, PHP, MySQL, AJAX, jQuery and CodeIgniter, since they are open source and are also the latest generation of web programming.

We will not insist on those technologies, more information on them can be found on the internet at the following web addresses:

http://www.w3schools.com/ - where one can learn from scratch how to build a website using HTML, CSS, PHP, http://www.php.net/ the main site for PHP used to generate dynamic content http://www.mysql.com/ - information on a relational data base managing system.

Modern programming offers new ways of building a website, frameworks being widely used, these being a collection of methods and pre-written classes, actually a source code, that is already optimized and secured, and more or less documented. They offer the user the possibility to create applications starting from something already defined.

XHTML, CSS, PHP, MySQL, AJAX are technologies known to most web programmers. We can say about CodeIgniter that it is a PHP based framework, developed by ELISAB - an open source project that reached version 2.0. It is very easy to use, very fast and has well implemented security elements as well as a strong documentation. These features make CodeIgniter very popular between PHP programmers. More information on CodeIgniter can be found at http://codeigniter.com/.

We have inserted, in Fig. 1, a print screen of our online shop – for more detailed visualization please visit www.virtualpc.ro.

Fig. 1 Home page of the VirtualPC online shop

2.2. Application´s functionality

The shop has 3 modules: interface module, user module and administration module.

2.2.1 Interface Module

This module represents what the user sees when he accesses the web site – item display, item selection criteria, keyword search, promotion and banner display, payment and contact information.

2.2.1.1 Shop Sections Accessed by User

The home page of the online shop has in its upper part a menu. With its help the user can navigate to the “Login”, “Register”, “Contact”, “How to buy” pages. In this header section we also have the logo, which is brought out in the center. The right part of the header has information about the purchased items/shopping cart.

An important element in the site header is the search module that enables the user to quickly search for the desired item within the site. This module has an own search engine, a module that is constantly improving since we try to provide the most relevant results depending on the searched keywords.

The next section offers the user the possibility to select a desired category. This menu was at first static, created through a list of information, the present version takes the information out of a data base and converts them in hyperlinks.

We tried to bring out promotions and special items in the middle and on the right of the home page. These items are randomly out of the data base (the centered part), combining these elements with an advert section which leads to our new product, the “VandPC” website.(a website specialized in free online IT adverts – momentarily still under development).

Starting from the home page the user selects from the left menu the category/subcategory of interest, while the center part displays the results of the selection the user made. Once selected, the subcategory in the right part of the site will display a filter with the purpose to generate results using user-desired criteria. This filter has the possibility to cancel some already chosen criteria through simple click on the “delete” button thus reversing to a wider selection.

Regarding this filter - we can also mention that it also enables clean information in the site’s URL. The searched information appears in the address bar under an easy to read form, as well as for the human factor as for the search engines – an extremely important feature for SEO. This cleaning of the URL of unwanted factors has no impact whatsoever, positive or negative on the correct functionality of the application.

Once the information has been listed after a category has been chosen or after filtering, if the number of results exceeds 12 (12 is the number chosen so that a symmetrical display structure could be provided), then the rest of the results will be displayed on consecutive pages, the criteria chosen by the user not being affected by the pagination of the products within the site.

For a better and quicker access of the next page we used the WEB 2.0 technology named AJAX, which enables the interrogation and display of the desired results without refreshing the page the user is currently on, a technology the gains more and more popularity in web programming.

When the user found the desired item he has two options: to purchase the item or to read further detailed information about it.

By pressing the “purchase” link, the user is sent to a new page, where he can see the content of his shopping cart or the quantity, the price, the VAT, partial sums and the total sun for the purchased items. The user can also refresh the information on quantity or even has the option to cancel an item he initially put in the shopping cart (quantity = 0 or by pressing the “delete” link).

If the user is not logged in, then a form will be displayed on this page which gives him the opportunity to login – the login being necessary to finalize the order.

2.2.1.2 Registration Interface / Login

These two sections can be debated together since their database is one and the same and they have common grounds, namely a form that enables the user to register or to login, the only difference being the nature of the fields contained by each form.

Creating a new account is easy; just press the link the page header namely “Register”. Very little information is requested, avoiding to lose the user’s interest by having him complete a very complex form.

The e-mail field is the one that will make the difference between the users. E-mail addresses are unique in our database, the user is warned when a new account is created if the e-mail address is already used in our database through a message “This e-mail address is already registered with us”.

As for the password field – we would only like to mention that the data is transmitted encrypted on 32-bit in our database.

Another mandatory feature for every web form is the element that makes the distinction between the human factor and a bot. We chose for the verification of a sum resulted by the generation of two random numbers, in the interval 0-9. The competition of the right number enables the “finalize registration” button and the registration process can be ended once this button is pressed.

To increase security, both the registration and the login form, present a real time warning system through jQuery – this system provides the user with information about the fields that are currently active.

Still regarding the safety of the information fetched from forms – these information are also filtered and cleaned of unwanted elements before they are saved into our database, with the help of functions offered by the development environment or algorithms developed by us.

The Login form also gives the opportunity to recover a lost password, by pressing the link “I forgot my password”, which leads to a field that once completed, will send to the specified e-mail address a new password consisting of a random group of letters and numbers, its generation being based on an algorithm that uses a random function.

We do not want to insist on the contact page, although we could mention the missing contact form, a negative aspect that will be set right in next version.

The “How to buy” page offers information on the buying and delivery process of the purchased items. We would like to specify that our online shop is meant for presentation, our main objective being the development of web orientated software products and not the actual online hardware trade.

2.2.2 User Module

This module resents that part of the site in which the user has access by means of an account, being able to purchase products from our online store.

The first step to access this module is to register the user’s information, this being explained in the previous section, by creating a new user account (unique) and a password, which is stored in our database through a 32-bit encryption system.

Once the user logs in for the first time into his new account, he is welcomed by a message that will prompt him to provide the details for billing and delivery. This information is mandatory, the lack of it means that the user will not be able to send an order to the shop’s administrators.

Regarding the user interface, we would like to add that there is also a module through that the user can edit its personal data, his password, his status as a natural person or as a company as well as the order history.

Once the user has entered the personal data, billing and delivery information, he will be granted access to the shopping cart through which he will be able to buy the items he chose from the VirtualPC online shop. The registered version of the shopping cart will enable the user to cancel, change or to end the purchase process. When the purchase order is finalized, a copy of the order will be sent to the user’s e-mail address.

2.2.3 Administration Module

This is the most complex of the three modules because it is used to manage the entire online shop. It contains the adding, editing, deleting, search and criteria based search of items features, it manges the orders as well as the billing. The normal user does not have access to this module, only the shop owners will be able to access and manage this module.
Not being accessible to the end user, we will not insist on the presentation of this module, but we would like to mention that it is a CMS (content management system), so that the development of unique e-commerce or other nature applications can be considerably reduced in the future.

3. Conclusions and further developments

As conclusions we would like to point out the main advantages of the presented application and future development plans and directions of our team.

The main advantages are:

1. Usage of the MVC system.

We can brag with the successful implementation of this new technology, MODEL – VIEW – CONTROLER, a technology that has as main purpose the the segmentation of the source code on separate categories, thus separating the VIEW (interface), from the MODEL (the part that deals with the application control, interrogations or with the information control functions), through the means of the CONTROLLER that receives the GETs and POSTs and decides how to manage them.

2. Frendly URL.

The URL is used by the search engines as a access modality for crawlers in different pages of the site. That is why a as relevant as it can be structure is necessary and CodeIgniter comes to help us in need, by offering this feature.

3. Filter System.

The information filter system is a necessary feature for a virtual catalogue, online shop, advert site and so on, since it facilitates a more precise display of the wanted product/service. Our team has developed its own information filter system to help the user to get the most accurate results.

4. Old Browser Compatibility.

A very important thing for every website, is that it can be viewed correctly in every browser, because the preferred browser for each unique visitors not known to the application developer. A major help in this matter is offered by W3C, a consortium that deals with the standardization of the rules used by XHTML and CSS. They also offer support for the verification and correction of the errors found in the source code of a website. We can say that we have a clean, well structured code, that has been validated by W3C.

5. Refreshing Information through AJAX.

Out of the desire to display information as quickly as possible, a system quickly developed that refreshes the content on a webpage without reloading the whole page. This can be achieved with the help of group of technologiesknown as AJAX (Asynchronous JavaScript and XML). VirtualPC makes use of this system in the pagination of the results within the application.

The future developments of our application follow the improvement of the performances and the filling of the gaps. Our team has set as its major goal the development of the presented project. We now have a new version of VirtualPC, which also gives the opportunity of credit card usage. We would also like to improve the search engine within our application by using more complex algorithms in the interrogation part of the data base.

Another goal in the development of the application is the improvement of the filter system, by turning it into a more powerful and accurate, by migrating to HTML 5 and CSS 3, by introducing the contact form and "face-lifting the site" with a new, more attractive design.

By starting with the "VirtualPC – online shop" application, we also developed a new site that, as far as we know, it will be a novelty on Romania's web market, because it will stand out through the combination of a virtual store and an advert site, thus resulting a HYBRID site. At the moment, based on our research we concluded that only cars and immobile sites allow the user to access the information within the site based on a filter system. We would like to accomplish this for an IT-specific advert site.

Bibliography:

http://www.w3schools.com/

http://www.php.net/

http://www.mysql.com/

http://codeigniter.com/

MANTA Alin
Lucian Blaga University
Computer Science Department
Str. Ion Ratiu nr. 5-7 550012
Sibiu, Romania
E-mail: mantalin@gmail.com

IGNISCA Madalin
Lucian Blaga University
Computer Science Department
Str. Ion Ratiu nr. 5-7 550012
Sibiu, Romania
E-mail: madalin.ignisca@gmail.com

CIOROGAR Ciprian
Lucian Blaga University
Computer Science Department
Str. Ion Ratiu nr. 5-7 550012
Sibiu, Romania
E-mail: contact@virtualpc.ro

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Image manipulation in frequency domain

Marko Milosevic, Jelena Tasic
Teacher Coordinator: Milan Tuba

Abstract

There are many ways to analyze images and for some purposes convenient spatial domain algorithms are appropriate, but for some operations like compression or many types of image enhancements more natural form of image representation is by amplitude coefficients in frequency domain. Images are represented as weighted sums of sine and cosine functions of different frequencies. Nowadays image analysis and enhancement become increasingly important. Besides everyday use in digital cameras, analysis and image enhancement are very important in areas like medicine, robot vision, satellite images etc. In this paper a software system that we developed for image manipulation in frequency domain is described as well as applications on some image enhancement processes.

1 Introduction

Digital images are becoming more popular and there is a growing number of situations where they are used. Having images in digital form allows processing by many useful mathematical operations that significantly increases their overall quality [1].

In computer graphics as well as in electronics, statistics and many other disciplines, frequency domain is a term used to describe the domain for analysis of mathematical functions or signals with respect to frequency rather than time (or space) [2].

The usual, well known way to represent image is by coding pixel by pixel individually, keeping information about each one of the components, whether it is RGB (stands for Red, Green, and Blue component), or any other color model. Such representation of images is called spatial domain representation. It is very handy for showing image on screen or printing it. That is how, for example, popular BMP file format keeps image data - pixel by pixel, memorizing three bytes (or some another amount, depending on supported number of colors in particular situation) for each pixel.

Storing images in spatial domain works very well, it is intuitive concept that is easy to understand, but for many image manipulation algorithms it is inadequate. Basic reason is simple: for a random set of pixels, we have to store them that way. But if we analyze standard pictures, it is obvious that they are not random collections of pixels, as spatial domain considers them. Most images we store are taken by camera, and they represent nature, sky, sea, people, and so on. Looking at them, we immediately notice some edges and objects. That is because large portions of images are usually identically or very similarly colored. For example, sky is blue or gray, but it is identically (or at least almost identically) colored, perhaps with some clouds.

Representing images in frequency, rather than spatial domain, is useful to reduce required space for storing them, but also for other purposes. In frequency domain it is easier to see some important features of an image. If we (or a software) take a look at image of the beach, it is obvious that there are some boundaries, or so-called - edges on it. These edges are easier to be noticed in frequency domain. Anomalies can be noticed as well. Let us consider camera with some dirt spots on the lens or somewhere else in the way of the light to the sensor, or to the film (if an old fashioned analog camera is used). That dirt spot will prevent light to get to the pixel in the "sky area" of, for example beach picture. By observing the image in frequency domain, and by filtering out high frequencies, these spots can be easily removed. Same applies to the problem of image noise. Having in mind that image noise is a random variation of brightness or color information in images, they can be removed with ease, if the image is processed in frequency domain.

Many of problems presented can be solved with simple filters that are easy to implement and yet very effective. Besides simple ones, plenty of more sophisticated techniques are used, like combinations of different filters. One of these approaches is a combination of a nonlinear low-pass and high-pass filters. This approach performs noise reduction as well as edge enhancement [3]. Even principles from artificial intelligence can be used in this area, for example, double sine basis function neural network is used for the design of 2-D low pass filters [4].

Image manipulation is also very important in many sciences. Analyzing images for medical purposes is one of applications. It is critical to filter unimportant details from some material, like X-ray or ultra-sound image. There are many techniques to achieve this, some of them are easier to implement, like method based on image enhancement using frequency spectrum analysis, presented in [7]. X-ray radiographic can be used along with frequency domain analysis also for some completely different purposes, like for inspection of welding joints [5]. Another application in medicine is analyzing optical coherence tomography in the coronary arteries with help of frequency domain image processing [6]. Also, one of newly used methods for medical image retrieval is based on frequency layer features [9]. Frequency domain manipulation is also used in chemistry for reconstruction of multi modal optical molecular image with frequency domain measurements [8]. A complicated technique based on same principles is identifying direction parameter of motion-blurred image achieved by using three second order frequency moments [10].

This paper is organized as follows: Section 2 describes a software system developed for image manipulation in frequency domain while Section 3 introduces tools for frequency domain manipulations: Fourier transform, Discrete Fourier transform and Fast Fourier transform. Section 4 is about image manipulation and filters.

A software system was written in C# that facilitates experiments with different frequency domain based algorithms.

2 Software

We developed a software system that allows experiments with images in frequency domain. Software application is built in Microsoft Visual Studio environment, in C# programming language. C# is chosen as a very wide spread language, easy to understand and as very fast developing tool. It is object-oriented and facilitates easy maintenance and upgrades.

To make it more portable, application does not use high-level libraries and all the features C# offers. Image data is manipulated manually, reading input file as array of bytes, as if it would have to be done in language like C, for example.

Software consists of options for reading and displaying images, it has some basic operation built in, and, also, it allows converting image date from spatial to frequency domain and applying filters.

3 Frequency domain fundamentals

3.1 Fourier transform

Jean Baptiste Joseph Fourier was French mathematician born in 1768. His important work was published in book "The Analytic Theory of Heat", and it was a statement that any function that periodically repeats itself can be expressed as the sum of sines and cosines of different frequencies, each multiplied by a different coefficient (we now call this sum a Fourier series). It had a huge impact on many sciences; it allowed to transform speech or electric signals, or, what we find most important here, an image data. Even functions that are not periodic (but whose area under the curve is finite) can be expressed as the integral of sines and cosines multiplied by a weighting function. The formulation in this case is the Fourier transform, and it is even more valuable in most practical problems.

$$g(t) = \frac{1}{2}c + \sum_{n=0}^{n} a_n \sin(2\pi n f t) + \sum_{n=0}^{n} b_n \sin(2\pi n f t) \tag{1}$$

where $f = \frac{1}{T}$ is the fundamental frequency, a_n and b_n are the sine and cosine amplitudes of the n^{th} harmonics, and c is a constant. This decomposition is called a Fourier Series, and it allows reconstruction of a function if the period T is known and the amplitudes are given.

Data signal with a finite duration can be considered like signal that is repeated forever, (for example, the interval from T to $2T$ is the same as from 0 to T.

The a_n amplitudes can be computed for any given $g(t)$ by multiplying both sides of Eq. (1) by $\sin(2\pi k f t)$ and then integrating from 0 to T. Having in mind that

$$\int_0^{\pi} \sin(2\pi k f t) \sin(2\pi n f t) dt = \begin{cases} 0 & k \neq n \\ \frac{T}{2} & k = n \end{cases} \tag{2}$$

only a_n survives, while b_n summation is vanished. Same idea is used to derive b_n, multiplying Eq. (1) by $\cos(2\pi k f t)$ and integrating between 0 and T. Constant c can be found by integrating both sides. Finally, result of performing these operations is:

$$a_n = \frac{T}{2} \int_0^T g(t) \sin(2\pi f t) dt \tag{3}$$

$$b_n = \frac{T}{2} \int_0^T g(t) \cos(2\pi f t) dt \tag{4}$$

$$c_n = \frac{T}{2} \int_0^T g(t) dt \tag{5}$$

3.2 Discrete Fourier Transform

Fourier transform is a general way to convert data signal (data) from spatial (or time) to frequency domain, but for image manipulation, it can be done by simpler transformation, since image data are discrete values.

Simplification of Fourier Transform that works just with discrete values is called Discrete Fourier Transform.

Definition 1 *The sequence of N complex numbers $x_0, ..., x_{N-1}$ is transformed into the sequence of N complex numbers $X_0, ..., X_{N-1}$ by the DFT according to the formula:*

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \tag{6}$$

where i is imaginary unit ($\sqrt{-1}$) and $e^{\frac{2\pi i}{N}}$ is a primitive N^{th} root of unity.

The inverse discrete Fourier transform (IDFT) is given by

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} \qquad n = 0, \dots, N-1. \tag{7}$$

If $x_0, \ldots, x_{N-1}$ are real numbers, and they are in most practical problems including image manipulation, then the DFT obeys the symmetry:

$$X_k = X^*_{N-k}. \tag{8}$$

The star denotes complex conjugation. The subscripts are interpreted modulo N.

3.3 Fast Fourier Transform

A fast Fourier transform (FFT) is an efficient algorithm to compute the discrete Fourier transform (DFT) and its inverse. There are many distinct FFT algorithms involving a wide range of mathematics, based from simple complex-number arithmetic to group theory and number theory.

Fast Fourier Transform computes the Discrete Fourier Transform and produces exactly the same result as evaluating the DFT definition directly, but much faster. Sometimes, in the presence of round-off error, many FFT algorithms are also much more accurate than evaluating the DFT definition directly.

Basic idea is simple: computing a DFT of N points in the naive way, using the definition, takes $O(N^2)$ arithmetical operations, while FFT can compute the same result in only $O(N\log(N))$ operations. The difference in speed can be substantial, especially for long data sets where N may be in the thousands or millions. In practice, the computation time can be reduced by several orders of magnitude in such cases, and the improvement is roughly proportional to $N/\log(N)$.

Let $x_0, ..., x_{N-1}$ be complex numbers. The DFT is defined by the formula

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi k \frac{n}{N}} \qquad k = 0, \ldots, N-1. \tag{9}$$

Evaluating this definition directly requires $O(N^2)$ operations: there are N outputs X_k, and each output requires a sum of N terms. FFT is any method to compute the same results in $O(N\log N)$ operations.

The most well-known FFT algorithms depend upon the factorization of N, but (contrary to popular misconception) there are FFTs with $O(N\log N)$ complexity for all N, even for prime N. Many FFT algorithms only depend on the fact that $e^{-\frac{2\pi i}{N}}$ is an N^{th} primitive root of unity, and thus can be applied to analogous transforms over any finite field, such as number-theoretic transforms.

Since the inverse DFT is the same as the DFT, but with the opposite sign in the exponent and a $1/N$ factor, any FFT algorithm can easily be adapted for it.

The well-known radix-2 CooleyTukey algorithm, for N a power of 2, can compute the same result with only $(N/2)\log_2 N$ complex multiplications.

4 Image Manipulation

4.1 Filters

To apply frequency filter to the image, Fourier transform has to be applied to the function, multiplied with the filter function and then re-transformed into the spatial domain.

Frequency filters can also be implemented in the spatial domain by finding a simple kernel for the desired filter effect. But frequency filtering is more appropriate if no straightforward kernel can be found in the spatial domain, and sometimes also more efficient.

Here is a full list of steps required to apply filtering in frequency domain:

1. Multiply the input image $(-1)^{x+y}$ to center the transform
2. Compute F(u, v), the DFT of the result image.
3. Multiply F(u, v) by filter function H(u, v).
4. Compute the inverse DFT of the result

5. Obtain the real part
6. Multiply the result of previous step by -1^{x+y}.

In the equation form, we will call $f(x, y)$ input image used in first step. Then the Fourier transform of the input image is given by

$$G(u, v) = H(u, v)F(u, v) \tag{10}$$

The multiplication of H and F involves two-dimensional functions and is defined on an element-by-element basis. That is - the first element of H multiplies the first element of F, and so on. In general, the components of F are complex quantities, but filters are real. Anyway, each component of H multiplies both the real and imaginary parts of the corresponding component in F. Such filters are called zero-phase-shift filters, because they do not change the phase of the transform.

That represents general way to apply any filter on image in frequency domain. What will be covered now is a few basic filters, useful for certain purposes, like smoothing and sharpening images. There are basically three different kinds of filters: low-pass, high-pass and band-pass filters.

4.2 Low-pass filters

Low-pass filters leave the low-frequency content of an image intact while attenuating the high-frequency content. Low-pass filters are good at reducing the visual noise contained in an image. They are also used to remove the high frequency content of an image so that the low-frequency content can be examined more closely.

Low-pass filters are not used just in image processing; they exist in many different forms, including electronic circuits, digital filters for smoothing sets of data, acoustic barriers, and so on.

The easiest one is the ideal low-pass filter. That is a very sharp filter, that cuts off all high frequency components of the Fourier transform that are at a distance greater than specified from the origin of the transform, and it looks like this:

$$H(u, v) = \begin{cases} 1 & D(u, v) \leq D_0 \\ 0 & D(u, v) > D_0 \end{cases} \tag{11}$$

where D_0 is a specified nonnegative quantity and $D(u, v)$ is the distance from point (u, v) to the center of the frequency rectangle, which is, as expected, at $(\frac{M}{2}, \frac{N}{2})$.

Filter like this is not used a lot, we rather use more sophisticated filters, like Butterworth, or Gaussian, although Gaussian is also an extreme, like the ideal low-pass filter, but the opposite extreme.

Gaussian low-pass filter in two dimensions is given by

$$H(u, v) = e^{\frac{-D^2(u,v)}{2D_0^2}} \tag{12}$$

This is a zero mean Gaussian with standard deviation σ.

There is only one variable in the Gaussian, its standard deviation. To get more flexibility, a more flexible filter, the Butterworth, is frequently used.

$$H(w, z) = \frac{1}{1 + [\frac{D(w, z)}{D_0}]^{2N}} \tag{13}$$

When converted to spatial domain, filters become simple matrices that multiplied with original image gives some result. Here will be presented some examples of these matrices.

This is matrix with equal weights for all pixels

$$\begin{matrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{matrix} \tag{14}$$

but also, even more usually are used matrices with the biggest value in the center, which reflects as the resulting pixel have biggest impact from its original value:

$$\begin{matrix} \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \\ \frac{1}{10} & \frac{1}{5} & \frac{1}{10} \\ \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \end{matrix} \tag{15}$$

and, for example:

$$\begin{matrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{matrix} \tag{16}$$

All of these matrices are formed so that sum of all elements is 1. They all perform the same task of at

4.3 High-pass filters

Analog to low-pass filters that suppresses high frequencies, high-pass filters leaves high frequencies unchanged, but suppresses low frequencies. As low-pass filters allows blurring images, high-pass filters sharpens image, since edges and other abrupt changes in gray levels are associated with high-frequency components.

Because sharpening is basically just a reversed operation of smoothing, we obtain high pass filter from relation

$$H_{hp}(u,v) = 1 - H_{lp}(u,v) \tag{17}$$

where $H_{lp}(u,v)$ is the transfer function of the corresponding low pass filter. Result is that when the low pass filter attenuates frequencies, the high pass filter passes them, and vice versa.

Again, as for low-pass three kinds of high-pass filters will be covered: ideal, Butterworth and Gaussian.

Same principle works here, the Butterworth filter represents a transition between the sharpness of the ideal filter and the total smoothness of the Gaussian filter.

Having in mind 11 and 14 2-D ideal High-pass filter is defined as

$$H(u,v) = \begin{cases} 0 & D(u,v) \leq D_0 \\ 1 & D(u,v) > D_0 \end{cases} \tag{18}$$

where D_0 is again the cutoff distance measured from the origin.

The transfer function of the Butterworth high-pass filter of order n and with cutoff frequency D_0 from the origin is given by

$$H(u,v) = \frac{1}{1 + [\frac{D_0}{D(u,v)}]^{2N}} \tag{19}$$

As with low-pass Butterworth filter, we can expect its high-pass version to behave smoother than ideal filter.

The transfer function of the Gaussian high-pass filter with cutoff frequency D_0 from the origin is given by

$$H(u,v) = 1 - e^{\frac{-D^2(u,v)}{2D_0^2}} \tag{20}$$

It is also possible to construct high-pass filters as the difference of Gaussian low-pass filters. These difference filters have more parameters, which gives more control over the filter shape.

Resulting matrices that are used in actual implementation have negative values for pixels around center (which means that values of surrounding pixels have negative effect on resulting pixel), and high value in center.

Again, matrices are built so their sum is 1.

Example with equal impact of all surrounding pixels:

$$\begin{matrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{matrix} \tag{21}$$

A little bit different variant in which only left-right and upper-bottom pixels have impact, but diagonal ones are ineffectual:

$$\begin{matrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{matrix} \tag{22}$$

Possible variants are also matrices with something like this:

$$\begin{matrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{matrix} \tag{23}$$

Like with low-pass filters, different matrices will determine different cut-off frequencies as well as directional properties of the filter. Which one will be used depends only on question what are they used for. One of basic facts on computer graphic in general, is that it based on subjective opinions, so someone may find one matrix better for certain purpose, someone else may disagree.

4.4 Example

After presenting theoretical basics, here is one example that shows how it works in practice.

Original image 256x256 pixels (Fig. 1) is taken, Fourier transform is applied, and then in frequency domain it is filtered.

Figure 1: Original image

After performing high-pass filter with the 25 as the cutoff frequency image looks like in Fig. 2.

Finally, what original image becomes like after smoothing with low-pass filter with 50 cutoff frequency image is presented in Fig. 3.

That is just a brief example of what can be done with filters, since any experiments can be done with our software described in Section 2.

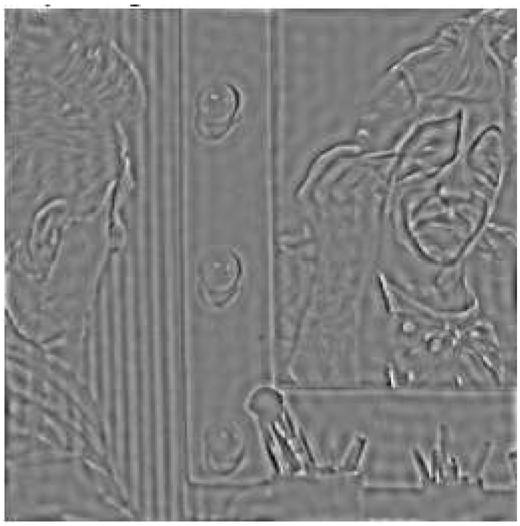


Figure 2: High-pass filtered image

Figure 3: Low-pass filtered image

5 Conclusion

Manipulating images in frequency domain is easier and more intuitive for many applications. Simple filters as illustrated before are sufficient for many purposes but more complex image enhancements are often required. Images from different sources (satellite, x-ray, ultrasound) have very different characteristics and require custom made filter combinations. Our object-oriented software system facilitates for easy addition of filters as well as for combining them.

References

[1] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, Pearson Prentice Hall, 2002

[2] Broughton, S.A.; Bryan, K., *Discrete Fourier Analysis and Wavelets: Applications to Signal and Image Processing*, New York: Wiley, 2008

[3] Guillon S, Baylou P, Najim M, Keskes N, Adaptive nonlinear filters for 2D and 3D image enhancement *Signal Processing*, pp. 237-254, 1998

[4] Chen Yangsheng, Yan Gangfeng, Optimum design of 2-D low pass FIR filters for image processing based on a new algorithm *2006 Chinese Control Conference, Vol. 1-5*, pp. 131-134, 2006

[5] Raiab Maher I., El-Benawy Talaat A, Al-Hazmi Mohammed W., Application of frequency domain processing to X-ray radiographic images of welding defects *Journal of X-ray science and technology,* pp. 147-156, 2007

[6] Adler Desmond C., Xu Chenyang, Petersen Christopher, Schmitt Joseph M., System design and image processing algorithms for frequency domain optical coherence tomography in the coronary arteries *Optical coherence tomography and coherence domain optical methods in biomedicine XIV* , 2010

[7] Li Wenjie, Chen Jie, Liu Yijin, Tian Jinping, Zhang Xiaobo, Liu Gang, Wu Ziyu, Tian Yangchao Image enhancement of x-ray microscope using frequency spectrum analysis 9^th *international conference on X-ray microscopy* , 2009

[8] Bartels M., Chen W., Bardhan R., Ke S., Halas N. J., Wareing T., McGhee J., Joshi A., Multimodal Optical Molecular Image Reconstruction with Frequency Domain Measurements *EMBC: 2009 annual international conference of the IEEE engineering in medicine and biology society, Vol. 1-20* , pp. 6655-6658, 2009

[9] Wang Yiping, Huang Xinsheng, Jia Peng, Direction Parameter Identification of Motion-Blurred Image Based on Three Second Order Frequency Moments *2009 international conference on measuring technology and mechatronics automation, Vol. I* , pp. 453-457, 2009

[10] Wu Jianghua, Wang Xiaoman, Wei Chaorong, Zhang Yixiang. Yan Donghui, Method for Medical Image Retrieval Based on Frequency Layer Feature *ITCS: 2009 international conference on information technology and computer science, proceedings, Vol. 2* , pp. 622-624, 2009

Marko Milosevic
University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: *markoub@gmail.com*

Jelena Tasic
University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: *jelenatasicsd@yahoo.com*

Prof. Milan Tuba
University Megatrend of Belgrade
Faculty of Computer Science
Bulevar umetnosti 29, N. Belgrade
SERBIA
E-mail: *tuba2@ieee.org*

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Building and programming a robotic arm

Kaloyan Mironov
Teacher Coordinator: Katalina Grigorova

Abstract

The paper describes the process of building and programming a robot arm. The paper contains a little research about the different types of robotic arms. It describes their hardware specifications. The paper describes the advantages of Java as language for the development of robot's software.

1 Introduction

The robotic arms are a major branch of robotics. The term "robotized hand" derives from the Czech word robota and is translated as „compulsory labour". The term describes well great part of robots in view of their purpose. [3]

The robotic arm is a robot manipulator, usually programmable, with functions close to the ones of the human hand. The robot is built out of joints that connect the separate components. This construction allows rotary as well as linear movement. The so called "end effector" is placed at the end of the robotized arm, which functionality resembles the human hand. The end effector is changed depending on the purpose of the robot. On Figure 1 you can see a robotic arm (right) and an end effector (left).

The robotic arm is one of the most useful technological inventions of the 20th century and rapidly turns into a benchmark for a lot of branches of industry. Robotized hands win recognition where the work is difficult or dangerous for humans.[1]

The first robotic arm has been developed in 1950 by a scientist named George Devol - Junior, until then robotics is mainly product of science fiction and imagination. Its development slows down during the next decades, because scientists are engaged to research outer space. The use of robotic arms in industry gains recognition in the 80s, when they are first integrated in automobile engineering and production.

Robotized arms production occupies the biggest relative part of robots production. After programming them, their function is reduced to performance of repeating movements, which are independent of human interference.

Another variety ofrobotic arms, concerning control, is the real time management by a human. This type of interaction, human-robot is used for solving problems in real time, in environment that is dangerous or hardly accessible for humans. This type of robots is used in outer space for bombs dismantling and in case of work in highly radioactive environment.

Fig 1 – Robotic arm and end effector

2 Hardware characteristics of the robotic arm

The standard robotized hand consists of seven segments and six joints. Computer managed step motors are used for joints, allowing movements in different areas. In contrast to the conventional motors, step motors move much more precisely by exact steps (one step can be a part of a degree). This allows programming the hand for precise and repeating movements. For better results, the robotized hand is equipped with movement censors, which watch for movement deviations.

An industrial robot of six joints looks much alike a human hand – his elements functionality is equivalent to human shoulder, elbow and wrist. Usually, the shoulder is mounted to a fixed base, not to moving objects. This type of robots has six degrees of movement freedom, which means they can move in six different areas. In comparison to that, human hand has seven degrees of freedom. This resemblance makes out of the robotized hand a perfect substitute of the human one. The architecture of a robotized hand is presented on Figure 2.

When moving, human hand moves from one position to another. By analogy, the robotized hand moves its end effector, through which it interacts with the surrounding environment. Robotized hands can be equipped with various kinds of tips, suitable for the relevant application. In most cases, one effector is a simplified option of a hand, being able to take and carry different objects. Robotized hands are often equipped with built in pressure sensors, which indicate the strength necessary to manipulate a certain object. This prevents the robotized hand from overloading or loss of balance. Other often used tips are: drills, burners or painting appliances.

In contrast to the human hand, the robotized one has a much bigger range of movements. The design of a robotic arm can ensure almost unlimited possibilities for movement and manipulation. The robot arm can be put at one and the same place – for example conveyor, or can be portable and moved from one place to another.

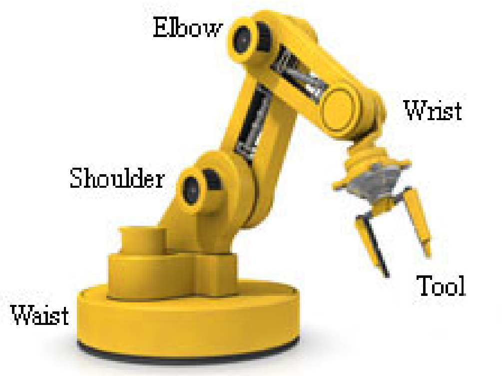


Fig 2 – Standard robotic arm architecture

3 Scope of use of robotic arms

The industrial purpose of the robotic hands is the repeating of one and the same action in controlled environment. Typical example is the conveyor processing. There are two methods for training of robotized hands – through software and through „manual regulator". The robot preserves in its memory the exact succession of movements and repeats them.

Greater part of the robotic hands work on automatic conveyors for assembly of automobiles. The robots perform this job much more effectively and quickly than man, which makes them much more preferred. The robots are positioned on the exact spot and use the exact power for performing the relevant job. Robots are independent of working hours, weather and they don't get tired. They occupy major place in computer industry – for assembling nano technologies.

4 Java as language for robots development

The language Java offers a lot of advantages, which makes it excellent choice in elaborating the project. The Java technology for object-oriented programming allows elaboration of easily understood and debugged applications. Java has the characteristics of a built in operational system. The language and the virtual machine maintain a lot of functions of a standard operational system, such as – multi-thread, synchronization and memory work. Another advantage of Java is the possibility for use of libraries. The Java technology ensures mobility of applications, which makes them platform independent. The developed software can be used with various operational systems.

5 The project

5.1 Introduction

The project follows the assembly of a robotized hand and the elaboration of software for its control. The application allows management of a robotized hand of up to 8 joints. The control is realized in real time, through the use of the interface graphic objects. For the purposes of the project, we use a robotized hand of 5 joints, 4 areas of movement and a standard end effector. The interface is totally developed in Java, in the environment of Eclipse. The hardware and software part of the project are presented on Figure 3.

Fig 3 – The hardware and software part of the

5.2 Hardware execution

The robotized arm consists of 5 joints and has a scope of movement 4 different directions. The arm is static and the movement in its bottom is realized through a rotating base. The project consists of three main components – a frame, a controller and servo motors - Figure 4. There are cramps, supporting elements and an end effector to the frame.[2]

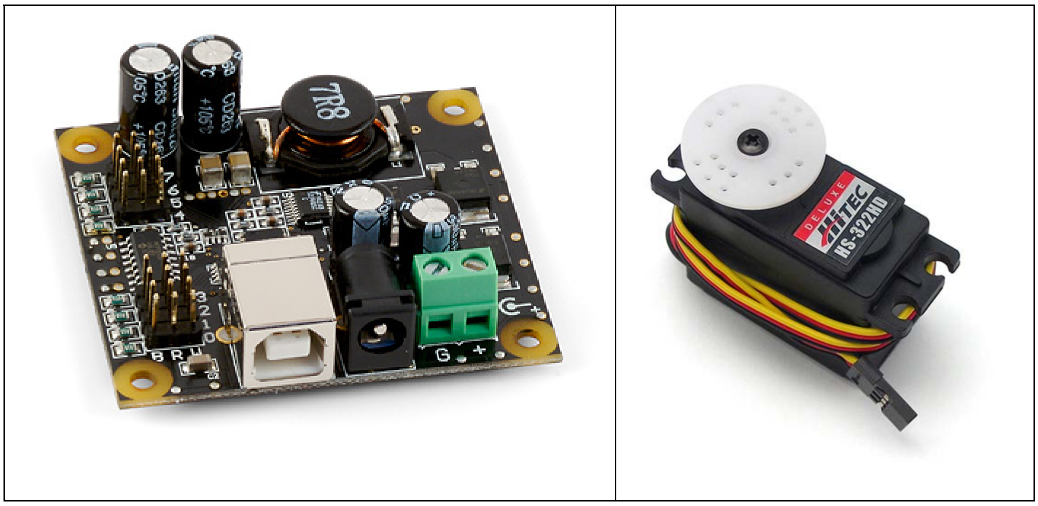


Fig 4 – Servo controller and servo motor

The number of servo motors is 5 and each plays the role of a joint in the hand. The measurement unit used to calculate movement in servo motors is degree. The minimum “step”, by which a joint can be moved, is 1 degree. The use of this type of motors sets limitation in the scope of movement. Each servo motor can move in the range from -90 to +90 degrees. This is presented in Figure 5.

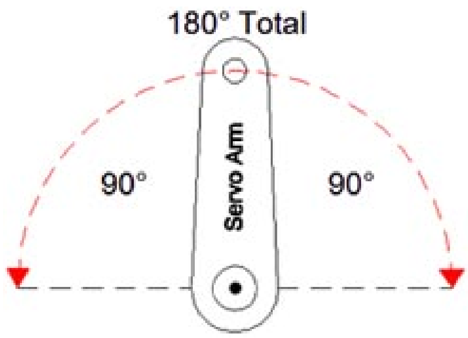


Fig 5 – Range of move

The controller maintains up to 8 servo motors and ensures control on the position, the acceleration and the speed of movement of each motor. The accuracy in positioning of a motor is 1/125 of a degree. The controller is connected to a PC through an USB port.

5.3 Software realization

The interface that manages a robotized platform should be easy for use but should also offer all necessary functions. The software should have simple and easily navigated interface that unites all components for robot control. The management of a robot in real time requires enough information on the managing parameters – position, acceleration, speed, etc.

The interface before and after its start is presented on Figure 6.

After the software start all components are inactive until the robotic hand is connected to the computer. The aim of this is indication of the connection between the hardware and the software part of the project and avoidance of mistakes.

5.4 Functionality

The software allows management of a robotized hand with up to 8 joints. The management is realized through selection of a motor from a falling menu. Condition of activity can be specified for each motor – switched on or switched off. Position and acceleration of the selected motor are set through sliders. The movement range is defined from the values of the minimum and maximum position, set through a slider. All current values responsible for the selected motor management are visualized in the right part of the interface. The software maintains function „Stepper”. In contrast to the real time movement, the function allows supple moving.

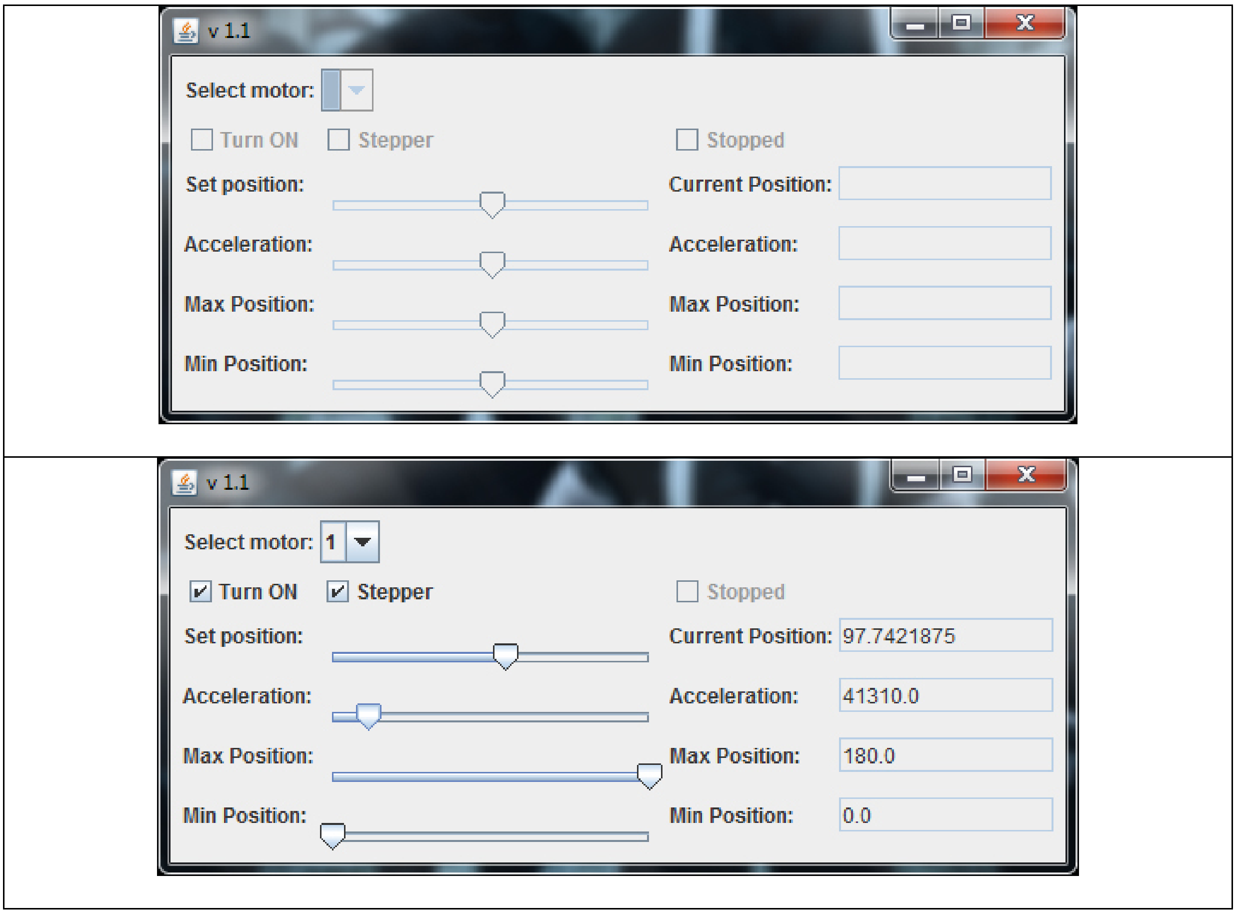


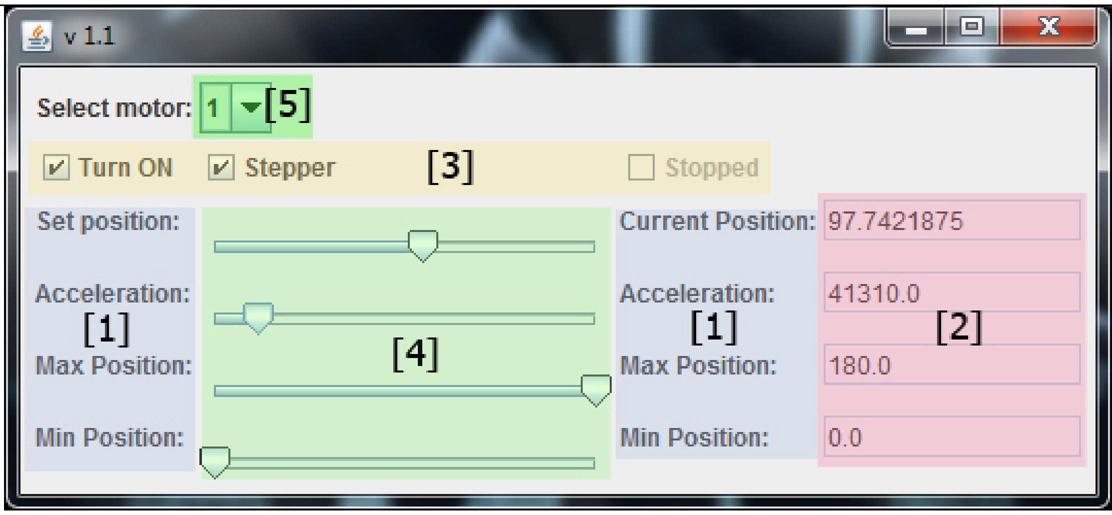
Fig 6 – The interface before and after its start

5.5 Components use

The interface has been developed in Java and uses the basic Swing components: JLabel, JComboBox, JTextField, JSlider и JCheckBox. Table 1 presents the Swing components, their placement in the dialogue box and the functions they perform. To facilitate, the components of each group are marked in different color.

Table 1 – Components placement in the dialog box

Placement



Swing component	Group number	Use
JLabel	[1]	It contains text that facilitates the interface use.
JTextField	[2]	It visualizes the actual value used by the management functions.
JCheckBox	[3]	It changes the working mode, too.
JSlider	[4]	It also gives the managing functions incoming data.
JComboBox	[5]	It offers motor selection.

The development of robotic arms requires the development of new software applications. The evolution of hardware is faster than the software. This is a precondition for shortage of software. The described in the paper software is a complex solution, because it can be used to control the main types of robotic arms.

References

[1] Robotic arm, http://www.en.wikipedia.org/wiki/Robotic_arm
[2] Phidget programming manual, http://www.phidgets.com/programming_resources.php
[3] How robots work, http://science.howstuffworks.com/robot2.htm

KALOYAN MIRONOV
University of Ruse
Informatics and information technology
#8 Studentska Str.
BULGARIA
E-mail: mail@kaloian.net

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Case-based Reasoning software framework extension

Andrei Neagu, Stelian Morariu
Coordinator: PhD. Ciprian Candea

Abstract

This paper describes the Case-based Reasoning (CBR) methodology, a young sub-domain of Artificial Intelligence, and its practical uses in real world problems. It focuses on the implementation of foundational concepts of CBR within an open source framework - jColibri2 – and the innovative work made by Ropardo S.R.L. in international research activity in order to extend the framework, to add functionality to it and to integrate it on a web infrastructure, enabling it to work with other larger systems. The primary objective is to provide an easy to use black-box layer for jColibri2, that will enable fast design, implementation and deployment of CBR systems, capable to reason in an unknown environment, with incomplete and disorganized data, learning on-the-fly, testing, adapting, extracting useful knowledge and developing in supervised and unsupervised manners.

Keywords: Case-based Reasoning (CBR), Artificial Intelligence (A.I.), jColibri2, Black-Box Framework, Knowledge-Intensive Systems, Ontologies, Textual CBR, GWT

1 Introduction

The main topic of this article is the Case-based Reasoning (CBR) methodology and it's usage for modelling and implementation of knowledge-intensive systems, capable of supervised and unsupervised learning. To be as straight-forward as possible, a visual representation of a CBR cycle is given here:

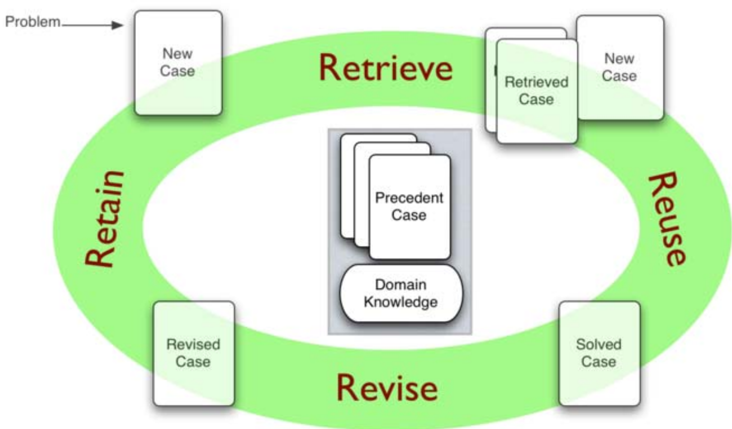


Fig. 1 CBR cycle

Our motivation is the fact that the CBR field is still a young one (the first prominent CBR systems appeared only after the 1990 year), it is very fertile and susceptible to future developments, yet, serious advancements have been made so far in this area. The applications that can take benefit from using this artificial intelligence sub-domain range from research projects and industrial process optimizations, to complex enterprise decision making. Practical implementation of the CBR concepts and operations is realized here using the jColibri2 open source framework. It has roughly 20,000 lines of Java code, and it uses more than 9 third-party software packages.

We also present here our efforts to extend and integrate the framework on a web infrastructure. At the present time, jColibri2 has only a white-box layer, therefore is available only to programmers, the construction of a good CBR system being a tedious and time consuming task; we aim at building a black-box layer for jColibri2 that will enable non-programmers to work with CBR concepts, to rapidly build state-of-the-art CBR systems. By doing so, the development process will be taking place on a more abstract level.

Similar researches[16]:

- UKIERI-CBRutils - A set of CBR methods designed to work in conjunction with the jCOLIBRI framework. Developed by: The Robert Gordon University & Indian Institute of Technology Madras
- myCBR Similarity Functions for jCOLIBRI - This contribution provides wrapper methods to use similarity measures generated with myCBR. Developed by: GAIA - jCOLIBRI developers & German Research Center for Artificial Intelligence DFKI GmbH
- Thunder - This contribution integrates a set of methods for the management of clustered case memory. Developed by: Research Group in Intelligent Systems - GRSI. LaSalle, Ramon Llull University.
- ALADIN - Abstract LAyer for DIstributed CBR Infrastructures. Developed by: GAIA - jCOLIBRI developers.
- ARFF Connector - This contribution provides a generic connector for (weka) ARFF files. Developed by: GAIA - jCOLIBRI developers.

The remaining content of the article is structured as follows:

- 2. *Case-based Reasoning* – short presentation of the CBR methodology; theoretical aspects
- 3. *Prominent CBR systems*
- 4. *jColibri2* – an overview of the jColibri2 framework and its capabilities
- 5. *Extending an artificial intelligence framework* – short description of our practical implementation; the extension and integration of jColibri2 in GWT context
- 6. *Conclusions and future developments* – possible integration of CBR with other A.I. fields, to obtain high-performance hybrid systems; distributed CBR
- References

2 Case-based Reasoning

2.1 The Case-based Reasoning methodology

To say that Case-based Reasoning is a technology is an understatement. First and foremost, it is a methodology that uses many other technologies from the Artificial Intelligence field: *Fuzzy Logic, Reducts, Rough Sets, Bayesian Networks, Artificial Neural Networks, Genetic Algorithms, Self-Organizing Maps (Kohonen Networks), Clustering Algorithms and others.*

CBR is used to extract useful knowledge from disorganized data, prediction of Web access patterns, medical diagnosis, weather prediction, legal inference, property valuation, corporate bond rating, color matching, design problems, pipeline process optimization, good practice identification, simulations etc.

Definition 1 *A case-based reasoner is an entity that solves new problems by using or adapting solutions that were used to solve old problems.*

The above given definition could very well apply to a representative of *Homo sapiens* . The CBR problem solving technique is very similar to the way humans solve new tasks.

2.2 The CBR cycle

The processes involved in CBR can be represented by a schematic cycle (see Fig. 1):

- RETRIEVE the most similar case(s);
- REUSE the case(s) to attempt to solve the problem;
- REVISE the proposed solution if necessary, and
- RETAIN the new solution as a part of a new case.

3 Prominent CBR systems

- KoBas – developed by Ropardo S.R.L.
- Experience Database – developed by Ropardo S.R.L.
- iDDesign Research Project – developed by Ropardo S.R.L.
- SMART: Support management automated reasoning technology for Compaq customer service
- Appliance Call Center automation at General Electric
- CLAVIER: Applying case-based reasoning on to composite part fabrication
- FormTool: Plastics Color Matching
- CoolAir: HVAC specification and pricing system

4 jColibri2

Fig. 2 jColibri2 logo [16]

jColibri2 is a framework developed in Java for building Case-based Reasoning (CBR) systems and it was constructed over the course of 10 years, mainly at GAIA research group, at Complutense University of Madrid http://gaia.fdi.ucm.es/projects/jcolibri/. It includes mechanisms to Retrieve, Reuse, Revise and Retain cases and it is designed to be easily extended with new components. [16]

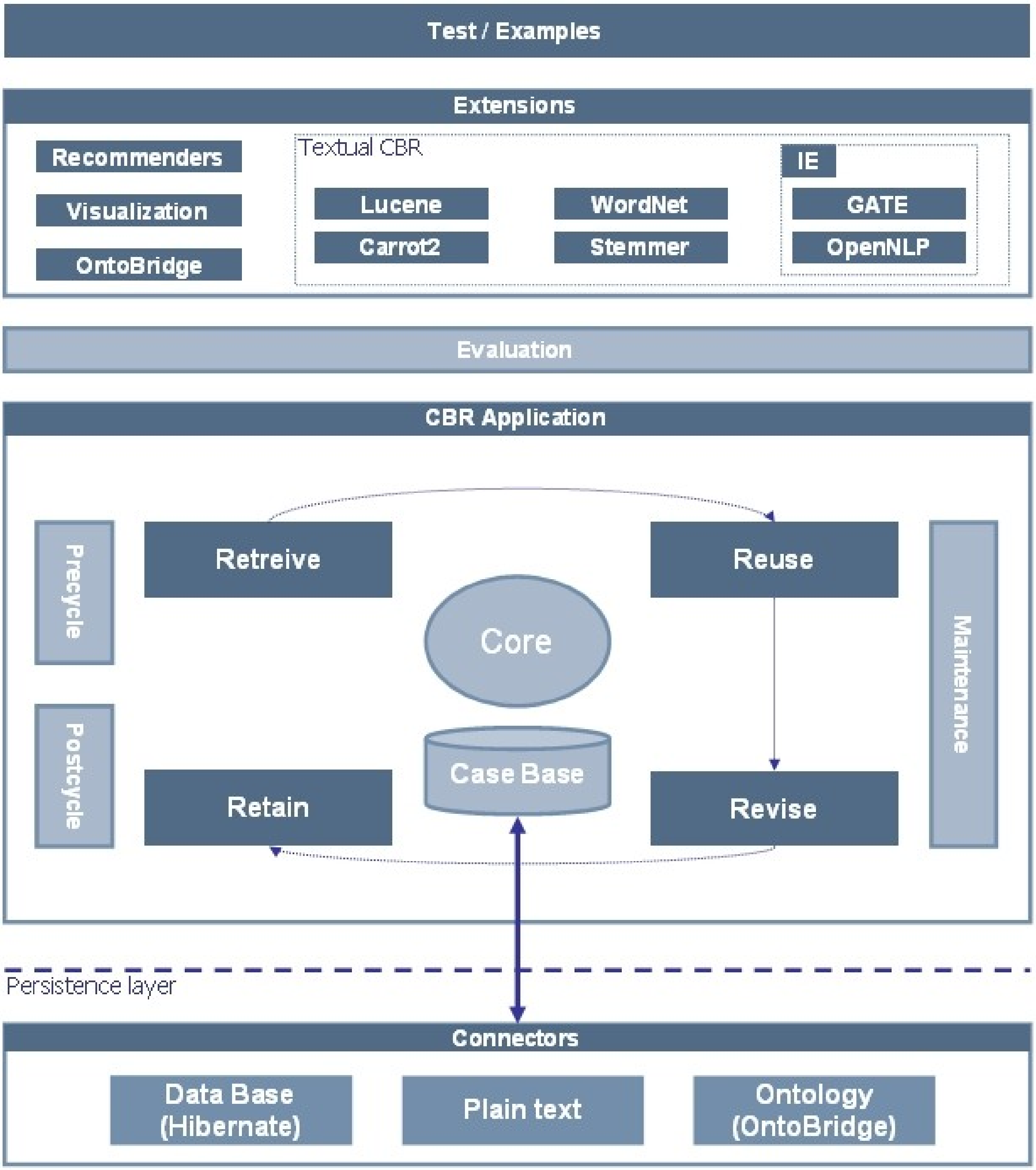


Fig. 3 jColibri2 architecture [16]

jColibri2 has a two-layered architecture:

- "The **bottom layer** has new features that solve most of the problems identified in the first version. It takes advantage of the new possibilities offered by the newest versions of the Java language. The most important change is the representation of cases as Java beans. A Java bean is any class that has a get() and set() method for each public attribute. Its modification and management can be performed automatically using a Java technology called Introspection (that is completely transparent for the developer). With this change, developers can design their cases as normal Java classes, choosing the most natural design. This simplifies programming and debugging the CBR applications, and the configuration files became simpler because most of the metadata of the cases can be extracted using Introspection. Java beans also offer automatically generated user interfaces that allow the modification of their attributes and automatic persistence into data bases and XML files. It is important to note that every Java web application uses Java beans as a base technology, so the development of web interfaces is very straightforward. The persistence of cases is going to be managed by the Hibernate package. Hibernate is a Java Data Objects (JDO) implementation, so it can automatically store Java beans in a relational data base, using one or more tables. It also supports XML files as storage media and has many tools that aid the persistence mapping. Hibernate offers a standard and well documented query language that could be used to express the CBR queries in a short future. Java Beans and Hibernate are core technologies in the Java 2 Enterprise Edition platform that is oriented to business applications. Using these technologies in jCOLIBRI 2.0 we guarantee the future development of commercial CBR applications using this framework".

- „Regarding the **top layer**, it is oriented to designers and includes several tools with graphical interfaces. It is the black-box version of the framework. This layer helps users to develop complete CBR applications guiding the configuration process. To perform this task these tools need semantic information about the Java components that compose the above layer. It is going to be done using Java Annotations, which is a new technology that allows to include meta-information in the source code. The annotations in the code are going to describe the behavior of a component with the corresponding information in CBROnto (our ontology of CBR). Then, the GUI tools can read the components information and reason using CBROnto to aid users in the design of the CBR application. An important feature of the designers layer is the representation of the case structure using OWL. Although cases are going to be Java beans, jCOLIBRI 2 is going to need semantic information about cases to aid in the composition of the CBR applications. Moreover, the CBR community is looking for a standard way for representing and interchanging cases so we are going to include an OWL based representation".

The information presented here was extracted from the paper "Lessons Learnt during the development of a CBR framework", SGAI-UKCBR Workshop, Cambridge 2006.

5 Extending an artificial intelligence framework

As it was stated before, our goal is making a black-box framework available to non-programmers, enabling non-IT professionals to work with A.I. Concepts. We are building the foundation(the basic blocks) that will enable development of more abstract concepts. Our practical implementation involves complex dynamic GUI, on-the-fly code generation and compilation, web integration.

The extension and integration of jColibri2 in GWT context is not an easy process. Implementation-wise, in order to enable jColibri2 to work with GWT, we had to rewrite parts of the framework, to make serializable versions of many components, to dynamically generate lightweight versions of objects etc.

We also had to employ advanced design patterns for GUI generation, because GWT does not offer support for reflection.

6 Conclusions and future developments

To conclude, our work will open jColibri2 and Case-based Reasoning to a much wider community, and this is of utmost importance.

Future development directions could be: integration of CBR with other A.I. fields, to obtain high-performance hybrid systems, distributed CBR computing, deploying the BB version of the framework on a cloud computing architecture (e.g. Google apps), CBR predictors, semantic web systems, integration of multi agent systems, integration of stochastic processes etc.

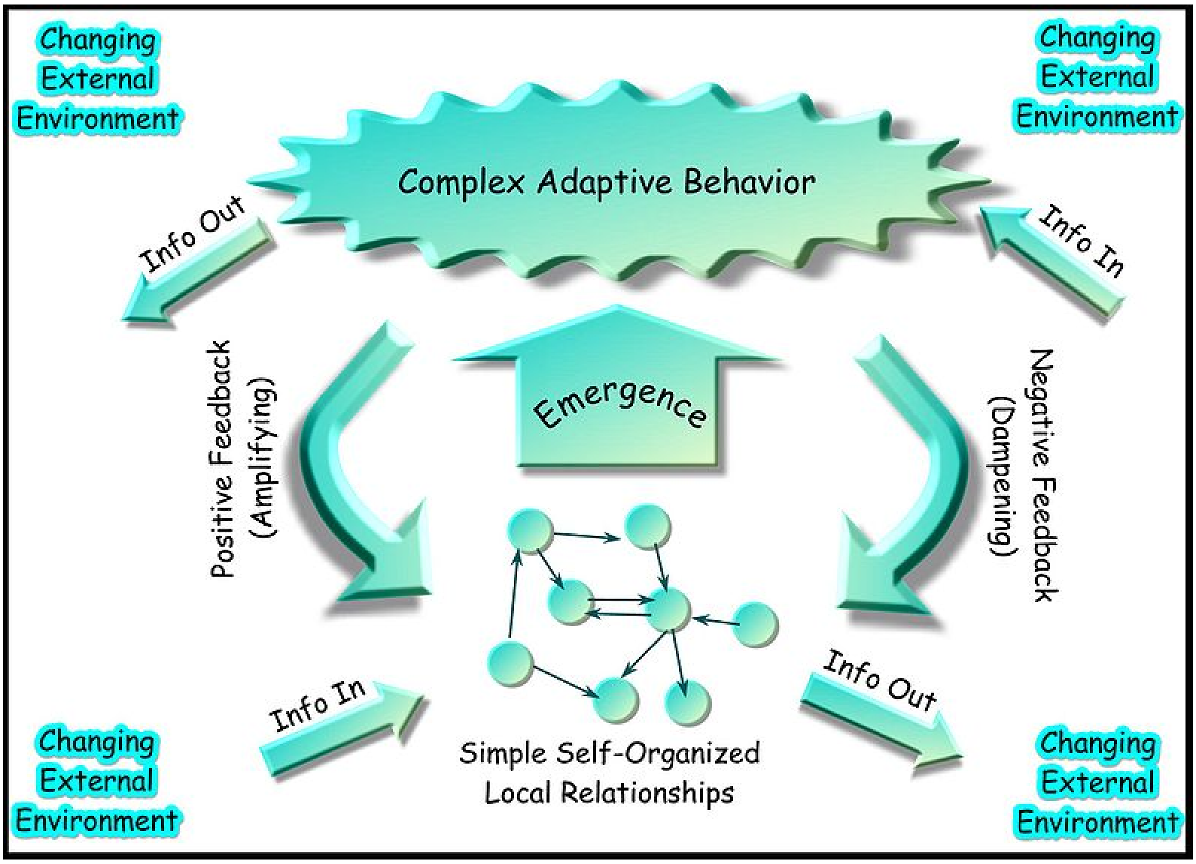


Fig. 4 http://en.wikipedia.org/wiki/File:Complex-adaptive-system.jpg

Acknowledgement: This paper is founded from the 7th Framework Programme research Grant.

References

1. Sankar K. Pal and Simon C. K. Shiu, *Foundations of Soft Case-based Reasoning,* John Wiley & Sons, Inc., Hoboken, New Jersey, 2004
2. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach (3rd Edition),* Pearson Education, Inc., Upper Saddle River, New Jersey, 2010
3. Belén Díaz-Agudo, Pedro A. González-Calero, Juan A. Recio-García and Antonio A. Sánchez-Ruiz, Building CBR systems with jCOLIBRI, in: *Special Issue on Experimental Software and Toolkits of the Journal Science of Computer Programming*, 69:1-3, 2007
4. Juan A. Recio-García, Belén Díaz-Agudo and Pedro A. González-Calero, jCOLIBRI 2 Tutorial, *Departamento de Ingeniería del Software e Inteligencia Artificial. Universidad Complutense de Madrid*, number IT/2007/02, 2007
5. Juan A. Recio-García, Belén Díaz-Agudo and Pedro A. González-Calero, Textual CBR in jCOLIBRI: From Retrieval to Reuse, in: *Proceedings of the ICCBR 2007 Workshop on Textual Case-Based Reasoning: Beyond Retrieval*., 2007
6. Juan A. Recio-García, Belén Díaz-Agudo, Pedro A. González-Calero and Antonio A. Sánchez-Ruiz, Ontology based CBR with jCOLIBRI, in: *Applications and Innovations in Intelligent Systems XIV.*

Proceedings of AI-2006, the Twenty-sixth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Springer, 2006

7. Juan A. Recio-García, Belén Díaz-Agudo, Antonio A. Sánchez-Ruiz and Pedro A. González-Calero, Lessons Learnt in the Development of a CBR Framework, in: *Proccedings of the 11th UK Workshop on Case Based Reasoning*, CMS Press,University of Greenwich, 2006
8. Albert Fornells, Juan A. Recio-García, Belén Díaz-Agudo, Elisabet Golobardes and Eduard Fornells, Integration of a Methodology for Cluster-Based Retrieval in jColibri, in: *ICCBR*, 2009
9. Juan A. Recio-García, Belén Díaz-Agudo, Derek Bridge and Pedro A. González-Calero, Semantic Templates for Designing Recommender Systems, in: *Proceedings of the 12th UK Workshop on Case-Based Reasoning*, CMS Press, University of Greenwich, 2007
10. Juan A. Recio-García, Derek Bridge, Belén Díaz-Agudo and Pedro A. González-Calero, CBR for CBR: A Case-Based Template Recommender System for Building Case-Based Systems, in: *ECCBR08*, 2008
11. Juan A. Recio-García, Belén Díaz-Agudo and Pedro A. González-Calero, Prototyping Recommender Systems in jCOLIBRI, in: *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, Lausanne, Switzerland, ACM, 2008
12. Juan A. Recio-García, Belén Díaz-Agudo and Pedro A. González-Calero, Semantic Templates for Case-Based Reasoning Systems, in: *The Knowledge Engineering Review*, 24:Special Issue 03, 2009
13. Juan A. Recio-García, Belén Díaz-Agudo, Juan M. Gómez-Berbis and Pedro A. González-Calero, The WINGS of jCOLIBRI: A CBR Architecture for Semantic Web Services, in: *Proceedings of the International Conference on Next Generation Web Services Practices*, IEEE Computer Society, 2007
14. Juan A. Recio-García, Belén Díaz-Agudo, Juan M. Gómez-Berbis and Pedro A. González-Calero, The WINGS of jCOLIBRI: A CBR Architecture for Semantic Web Services, in: *International Journal of Web Services Practices (IJWSP). Selected papers from the International Conference on Next Generation Web Services Practices Conference (NWeSP'07)*, 3:1, 2008
15. Sergio González-Sanz, Juan A. Recio-García and Belén Díaz-Agudo, D2ISCO: Distributed Deliberative CBR Systems with jCOLIBRI, in: *ICCCI*, 2009
16. http://gaia.fdi.ucm.es/projects/jcolibri/jcolibri2/index.html
17. http://www.ai-cbr.org/classroom/cbr-review.html
18. http://en.wikipedia.org/wiki/Case-based_reasoning

Andrei NEAGU
"Lucian Blaga" University of Sibiu
"Hermann Oberth" Faculty of Engineering
Department of Computer Science
4 Emil Cioran, Sibiu, Romania 550025
Creativ Solutions S.R.L.
2A Reconstrucției, Sibiu, Romania 550129
ROMANIA
E-mail: andrei.neagu@ropardo.ro

Stelian MORARIU
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Department of Informatics
5-7 Ion Rațiu, Sibiu, Romania 550012
Ropardo S.R.L.
2A Reconstrucției, Sibiu, Romania 550129
ROMANIA
E-mail: stelian.morariu@ropardo.ro

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Using Limited Influence Maps to Control Agents in Pac-Man

Dragos Nicolescu
Teacher Coordinator: Catalin Stoean

Abstract

In this paper we present the development of an influence map based controller for the Pac-Man agent and the results it yields when compared to a previous implementation. While influence maps have been used before to control an agent, the method presented in this paper differs by using a limited area of propagation for the influence. This particularity helps the agent better avoid enemies, leading to a longer lifetime. When conducting experiments in similar circumstances the resulting agent met our expectations, being competitive or better when compared to the previous implementation. Tests have also revealed the way certain parameters should be set and how they affect the model. Overall, the limited influence map proved to be an ideal solution in some situations, resulting in an adaptable agent that is very sensitive to the environment.

1 Introduction

Computer games have always been great test platforms for Artificial Intelligence (AI) methods, providing environments which present agents with challenges and offer them rewards depending on their behaviour. Pac-Man is a simple game, but it also offers the possibility of developing more complex strategies for playing it. Good human players often use a combination of speed, reaction-time and basic strategic thinking to complete the game. A computer-controlled agent will naturally have great speed and an almost immediate response-time. What such an agent lacks is strategy, and this is where AI comes in.

The influence map method is a good choice for controlling Pac-Man, being a technique specific to computer games and robotics. For some time influence maps have been used in complex real time strategy games to facilitate high-level decisions due to the fact that an influence map can present information about the state of the game in a concise way. The computer game industry is always trying to provide competitive computer-controlled enemies for human players, and an important step in this direction is developing agents that can sense subtle changes in the environment and can easily adapt to them. This fact, coupled with the recent trend of moving from hard-coded agents to intelligent agents that learn and adapt lead to the implementation of influence maps in controlling individual agents [1].

In this paper we present the implementation of an influence map based controller for Pac-Man. The main objective is to develop an agent that behaves better than its predecessors and to study the way certain parameters affect its performance. A secondary objective is to reduce the

computation time for calculating the influence map without compromising the relevance of the information it contains.
In the following section we will shortly describe the Pac-Man game, analyze the concept of influence maps and present a previous implementation of this technique in Pac-Man. Section 3 describes the purposed influence map model and its implementation. In section 4 we present the experiments carried out in order to compare the purposed model to its predecessor. In the last section we draw conclusions and suggest a few general directions for further research.

2 Influence maps and Pac-Man

In order to fully comprehend the implementation of the controller based on an influence map it is important to understand how the game works, what are the rules that shape the environment, and the way an influence map represents changes in the game world.

2.1 The Pac-Man Game

Pac-Man is an arcade-style game that was launched in Japan on May 22, 1980 by Namco. The game quickly achieved great success and is still played today on a variety of devices, ranging from mobile phones to game consoles and personal computers. The popularity of the game can be attributed to the fact that it is simple enough to be intuitive, yet hard enough to require some practice [2]. For this reason Pac-Man is an ideal platform for implementing some AI algorithms.
The game idea is very simple: the player controls an agent through a maze which he sees from the top perspective. The agent can only navigate along the corridors of the maze and must collect different reward-objects in order to score points. The agent must also avoid enemies called *ghosts* that also move through the labyrinth. There are two types of reward objects: pellets and power-pills. Power-pills are worth more points and also have an important effect on the agent: for a short amount of time the ghosts will become *edible*, meaning that the agent can *eat* them and gain extra points for doing so. The maze contains a great number of pellets, but only few power-pills.
When the game starts the player has a limited number of lives (usually 3). When the agent is touched by a ghost it loses a life and goes back to the starting position (this is not a set-back, as the mazes are usually easy to navigate). When an agent loses all of his lives the game is lost. In order to advance to the next level the agent must collect all rewards in the maze. When this task is completed the agent keeps his score and the remaining number of lives and the maze is reset (or a new maze takes its place).

2.2 Influence maps

Influence maps in CI are similar to potential fields in robotics, but they are used for more than just motion planning, being able to express detailed information about the environment.
An influence map is a function defined over the environment that expresses the desirability of the locations on the map according to some measure. We can consider an influence map as a grid that covers the environment. Each cell of the grid can contain one or more entities. If the agent is rewarded for collecting an entity then this object will have a positive influence. If the object is meant to be avoided it will have a negative influence on the agent. In the case of Pac-Man pellets, power-pills and *edible* ghosts have a positive influence, while *inedible* ghosts have a negative influence [3].
Positive or negative influence is propagated, meaning that an object will not only affect the “cell” which it occupies, but also neighbouring cells. This means that the desirability of a certain location in the environment is calculated as the sum of all influences that affect that location. For a location L in the environment this can be expressed through the following equation (1):

$$InfMap(L) = \sum_{i=1}^{n} \mathrm{Inf}(R_i, L) - \sum_{j=1}^{m} \mathrm{Inf}(E_j, L) \quad (1)$$

where R_i with $i \in \{1, \dots, n\}$ are the rewards and E_j with $j \in \{1, \dots, m\}$ are the enemies. The previous equation states that any location in an environment is influenced by every location in that environment. Because this influence cannot propagate as a whole, a function for influence propagation is required. The further an agent is from an object, the less it is influenced by that object, so it is logical that influence should decrease with distance.

Considering that the influence of an entity is strongest in the cell the entity occupies and it decreases as the distance from that cell increases we determine that there are two basic parameters to consider when building an influence map model:

- *Center Influence* – the influence which the object has on the grid cell in which it is placed; this is considered the "true" influence of the object;
- *Propagation Method* – the method which describes the way the influence propagates to neighbouring locations.

2.3 A Previous Approach (IMA)

A successful influence map agent (IMA) was developed in [4]. This implementation considers that every object influences every location in the maze, and as a result the propagation method covers the entire maze.

The authors consider that as the number of rewards gets smaller their importance should grow, and so their influence gets stronger, helping the agent find any leftover pellets or power-pills. The influence of ghosts, however, decreases as the number of rewards left in the maze gets smaller. This way an agent nearing the completion of the level (just few rewards left) will tend to be less careful when avoiding enemies and more determined to collect the last rewards.

Considering the position *i* with coordinates *x* and *y*, the value of the influence map for this position is described by the following equation:

$$influence(i_{(x,y)}) = \sum_{k=0}^{n_d} \frac{p_{dots} \cdot \bar{n}_d}{d(i_{(x,y)}, k_{(x',y')})} - \sum_{k=0}^{n_r} \frac{p_{run} \cdot n_d}{d(i_{(x,y)}, k_{(x',y')})} + \sum_{k=0}^{n_g} \frac{p_{chase}}{d(i_{(x,y)}, k_{(x',y')})} \quad (2)$$

where n_d is the number of uneaten dots (pellets and power-pills) in the maze and $\bar{n}_d$ is the number of eaten dots. $d(i_{(x,y)}, k_{(x',y')})$ is the distance between the location for which we are calculating the influence and the object influencing it ($k_{(x',y')}$). This distance was calculated using the Euclidean distance. The roles of the sums on the right-hand side of the equation are as follows:

- *First sum* – produces the positive influence of the uneaten dots;
- *Second sum* – introduces the negative influence of inedible ghosts;
- *Third sum* – produces the positive influence of edible ghosts.

p_{dots}, p_{run} and p_{chase} are parameters that act as weights for each sum in the equation.

Equation (2) treats pellets and power-pills equally, making no direct distinction between the two. The effect of power-pills it considered indirectly by taking into account edible ghosts. As the game progresses the number of dots decreases and the importance of the remaining dots increases. The influence of the inedible ghosts decreases with the progress of the game. Edible ghost influence is controlled using just p_{chase} , meaning that edible ghosts have the same influence throughout the game.

3 Limited Influence Map Agent (LIMA)

The Limited Influence Map Agent is the result of a development process that aimed to test the importance of the method for influence propagation. A secondary objective was developing a method for calculating the influence map which would be faster than previous implementations.
The main difference between IMA and LIMA, as the name suggests, is that LIMA uses a limited area of influence propagation, rather than having the influence propagate in the entire maze, as is the case with IMA.
The first step in developing LIMA was selecting the distance which would best fit the scenario of Pac-Man. Considering that the agent can only move parallel to the edges of the maze (no diagonal movement) the distance that best describes this movement is the Manhattan (or city block) distance.
Considering an object *O* with coordinates *x'* and *y'* and a location *L* with coordinates *x* and *y*, the influence *O* has on *L* is described by the equation (3):

$$influence\left(O_{(x',y')}, L_{(x,y)}\right) = \begin{cases} I_{center}(O) & if \;\; d\left(O_{(x',y')}, L_{(x,y)}\right) = 0 \\ I_{center}(O) - \dfrac{d\left(O_{(x',y')}, L_{(x,y)}\right)}{d_{propagation}(O)} & \\ \quad\quad if \; d\left(O_{(x',y')}, L_{(x,y)}\right) < d_{propagation}(O) & \\ 0 & otherwise \end{cases} \quad (3)$$

where $I_{center}(O)$ is the center influence of the object, and $d_{propagation}(O)$ is the maximum range at which the influence of *O* is propagated.
The purposed model considers the center influence and the propagation distance as parameters for each object in the game. Therefore, the total number of parameters for the LIMA method will be *2n* where *n* is the number of entities that influence the agent. The objects that influence the agent are: pellets, power-pills, inedible ghosts and edible ghosts; therefore the number of parameters for LIMA should be 8. This number is greater than the number of parameters of the IMA method (p_{dots} , p_{run} and p_{chase}) and therefore the method will be harder to optimize. In order to reduce the number of parameters we have considered avoiding inedible ghosts just as important as chasing and eating edible ones. Based on this judgement, edible and inedible ghosts have the same distance of propagation and their influences are calculated as follows:

$$I_{center}(EdibleGhost) = -1 \cdot I_{center}(InedibleGhost) \quad (4)$$

By using the above equation the LIMA method will have 6 adjustable parameters, which is still double the number of parameters IMA has, meaning that although harder to optimize, LIMA will allow for a greater flexibility in setting specific goals for the agent.
Limited influence means that at some point in the game (when there are only few rewards left) the agent might find himself in an area with a constant influence value of 0. In such a case the agent will chose the next step at random, without being allowed to turn back, in order to avoid situations where he moves repeatedly between two adjacent locations. Such behaviour means that levels take longer to finish. This is not necessarily a disadvantage because the limited and dynamic nature of the environment (ghosts constantly moving) means that such situations are rare and quickly resolve themselves.

4 Implementation

Previous implementations of controllers for Pac-Man have been done mostly in C or C++, but for the purposed method it was decided to use Java in order to simplify the implementation of the visual interface.

4.1 LIMA Implementation with Visual Component

Before implementing the LIMA method as a controller it was clear that some means of supervision was necessary. It was decided that the best way to do this is implement the controller in a standard Pac-Man computer game created for human players which naturally has a visual interface.

The game level that was used is one of the classic Pac-Man levels containing 131 pellets and 4 power-pills. The maze can be seen as a grid with an area of 15×15 cells. The environment, along with the imaginary grid can be seen in Fig. 1.

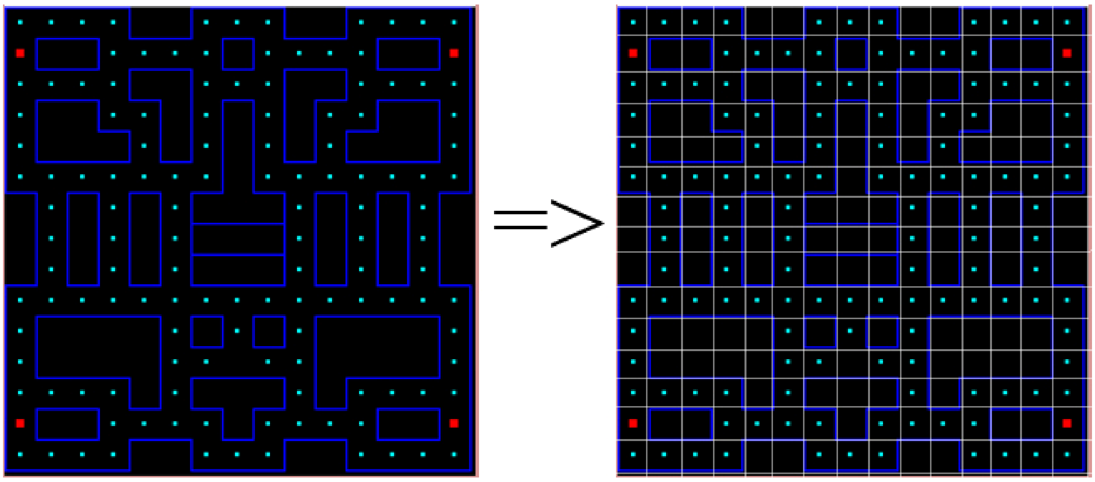


Fig. 1 The game level (left) and the grid that divides it into cells (right).

Using the grid to divide the environment into cells means that a two dimensional array can be used to store information about the maze. The first element of the array (line 1, column 1) represents the cell in the upper, left-hand side corner of the grid.

In order to code information about the game world in the array the basic characteristics of a cell were denoted by prime numbers. For example a left wall is represented by 7; a right wall is represented by the number 3; the characteristic *"contains a pellet"* is represented by 11. A cell that is limited by a left wall and a right wall and also contains a pellet will be represented in the array by the product $3 \cdot 7 \cdot 11 = 231$. This information can be dynamically updated with ease: if the agent collects the pellet from a cell then the number representing that cell will be divided by 11. Considering *leveldata(i,j)* a cell in the *leveldata* array that represents the maze, the essence of coding information about the environment is represented by equation (5):

$$leveldata(i,j) = \prod_{k=0}^{nrcaract} code(X_k) \tag{5}$$

where $1 \le i, j \le 15$, *nrcaract* is the number of basic characteristics of the cell and $code(X_k)$ is thefunction that assigns a prime number to each basic characteristic X_k.

Supervision of the method and analysing the actions of the agent means there is also a need for observing the influence map. The influence map is represented by another two dimensional array with the same dimensions as the one coding the maze: 15×15 cells. In this array each cell contains the value of the influence for the location in the maze which it represents. Due to the dynamic nature of the game, the influence map is updated very often. This is a problem when trying to analyze it during a game. Because of this a new way of observing the influence map had to be found, one that would present the information it contains in an intuitive way.
The solution was to represent the influence map as a grid in which each cell is coloured based on the value of the influence in the location it represents. In order to make this representation even more intuitive we decided to use the colour red for negative influences and green for positive ones. As the value of the influence in a cell decreases, the value of red increases in the RGB colour model for that cell. The higher the influence is in a cell, the greener the colour representing it will be. Cells that have an influence value of 0 are depicted using grey. The result of colour coding the influence map can be observed in Fig. 2.

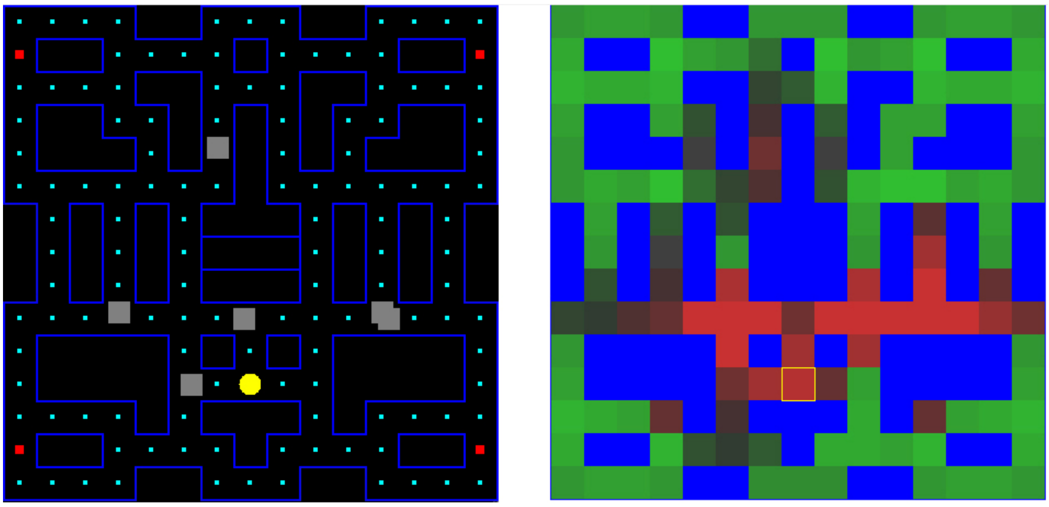


Fig. 2 A situation in the game, with the agent being represented by a yellow disc and the ghosts as grey squares (left) and the same situation colour coded for easy analyzing (right)[1].

4.2 Implementation of a test Platform for LIMA and IMA

Testing LIMA and making a direct comparison between it and IMA means implementing a test platform where both methods are run in the exact same conditions.
As with the LIMA implementation with visual component, the test platform was developed in Java. The application consists of 2 classes that model the agent and the ghosts and a class that runs the actual game without having a visual component. The tests are run for hundreds of times, at high speeds and can sometimes take few hours to complete. In these conditions a visual component would have been useless, because it is impossible to draw any conclusions by watching the game.
In order to test multiple parameter sets for both methods the program input is done via .in files that contain a header specifying what each column represents. After reading a row (representing a set of parameters) from the input file, the application will run a number of games (usually 100 or 1000) using those parameters to build the influence map. After the games are run, the results are saved as a row in an Excel file. This row contains various information about the results for a set of parameters; for example the number of ghosts, pills and pellets the agent ate, the number of times

[1] This application can also be observed online at http://inf.ucv.ro/~cstoean/students_projects/2010/PM.html

the agent won a game, the number of moves the agent made, etc. After the application completes the test, the output Excel file will have the same number of rows as the input file, each row of results corresponding to a row (or a set) of input parameters.
Using the method described above an observer can clearly analyze the output of both methods and conduct various comparisons between them which are described in the next section.

5 Experimental Comparison

A comparison between to the two controllers was needed in order to determine the strong points of each method and to better understand the way influence propagation affects the results of the game.

5.1 Setting up the Experiments

The direct comparison between the two methods had to be performed in similar circumstances, without giving any method an advantage or disadvantage. Therefore, several aspects had to be considered, most important of which was the way to set the parameters for both methods. In order to perform a fair test, we could not use only one set of parameters for all the tests, so it was decided to use 50 parameter sets for each method and then compare the best results, the worse results and the average ones.
A set of LIMA parameters consists of 6 numbers for setting the center influence and propagation distance for pellets, pills and ghosts. IMA only uses 3 parameters: p_{dots} , p_{run} and p_{chase}. Because the latter has a smaller number of parameters and the tests will be carried out on the same number of parameter sets, IMA will have the advantage of covering more of the exploration field for its parameters.
The next phase in building the parameter sets for the two methods was setting the intervals for each parameter. Setting these intervals means taking into consideration various other parameters, like the dimensions of the maze or the importance of each task the agent has to accomplish. The final definition intervals for the parameters of the two methods are as follows:

1. LIMA:
   - 1.1. Pellet:
     - Center Influence $\in (0, 5)$
     - Propagation Distance $\in \{1, 2, 3, 4\}$
   - 1.2. Power-Pill:
     - Center Influence $\in (0, 15)$
     - Propagation Distance $\in \{1, 2, 3, ..., 8\}$
   - 1.2. Ghosts:
     - Center Influence $\in (-50, 50)$
     - Propagation Distance $\in \{1, 2, 3, ..., 8\}$
2. IMA:
   - 2.1. PChase $\in \{5, 6, ..., 45\}$
   - 2.2. PDots $\in \{1, ..., 30\}$
   - 2.3. PRun $\in \{2, ..., 50\}$

The IMA parameter intervals respect the suggestions in [4], being adapted for the size of the maze used in this experiment.
After the parameter intervals were established the 50 parameter sets used for testing each method were constructed by using the Latin Hypercube Sampling (LHS) statistical method. This method generates equally distributed configurations, and has been successfully used before to set values for parameters [5].

Next issue was setting up the reward system for the agents. The reward point value of each task has to reflect its difficulty, so it was decided to adopt the following values:

- collecting a pellet brings 10 points;
- collecting a power-pill brings 50 points;
- capturing an edible ghost brings 100 points;
- finishing a level brings 500 points.

For both methods we tested the 50 sets of parameters by running 100 rounds of the game for each parameter set. The results that are collected for the 100 rounds/parameter set are: the number of collected pellets, the number of collected power-pills, edible ghosts killed, number of finished rounds (or *wins*), the number of moves made by the agent, and the total score. Each of those results represents the total over 100 rounds.

5.2 Results and Observations

Experimental results are presented as a mean over the 100 rounds run on a specific parameter set (in the case of the best and worst configurations) or as an average for all 50 LHSs (Table 1) [3]. Also included in the table is a row showing the standard deviation for the results. The best and worst configurations are designated as such due to the achieved score. Statistical tests (t-test and Wilcoxon rank-sum test) are run on the presented results in order to determine significant differences (values written in bold in the table).

Measure	**Best**		**Worst**		**Average**		**Std. dev.**	
	IMA	**LIMA**	**IMA**	**LIMA**	**IMA**	**LIMA**	**IMA**	**LIMA**
Pellets	120.7	121.5	33.1	**38.1**	83.3	**89.6**	21.6	17.6
Power-pills	3.68	3.67	0.02	**1.06**	1.32	**3.03**	1.01	0.63
Ghosts	**1.55**	0.91	0.02	0.31	0.6	0.57	0.43	0.14
Wins	0	0.08	0	0	0	0.01	0	0.02
Moves	**618.2**	556.9	133.4	**241.6**	294.4	**405.2**	112.2	61.7
Score	1532	1515	344	**470**	960	**1116**	302	211

Table 1: Test results with major differences written in bold.

By observing the significant differences in the table it can be noted that for the best configurations IMA behaves better regarding eaten ghosts and the number of moves the agent made. When considering the worst parameter set and the average results it is clear that LIMA has significantly better performance regarding the number of eaten pellets and power-pills, the number of moves the agent makes and the overall score. In fact, for the average results, IMA is only slightly better than LIMA considering the number of eaten ghosts.

IMA has a better performance at its best but this performance is quite different from the ones achieved by the worst and average configurations. This fact is also proven by the IMA standard deviation which is constantly higher when compared to LIMA .

A striking difference between the two methods is the number of wins achieved by the agent. IMA did not achieve a single win for any of the 100 runs for each of the 50 parameter sets. A possible explanation for this behaviour will be discussed in the following subsection. LIMA managed to win 8 of the 100 rounds for the best configuration, having an average win rate of 1 out of 100.

5.3 Discussion

As stated before, IMA has a smaller number of parameters to set than LIMA. An equal number of configurations means a far better parameter tuning for IMA. Although setting the parameters for LIMA is more intricate the results prove that the method is much more dependable when assigning the parameter values. For the configuration with the worst results LIMA performs better than IMA on all accounts, except wins (neither method won a round). This, along with the standard deviation which is greater for IMA for all results, proves that the purposed method is better balanced compensating for the larger number of parameters.

Next analysed are the average results which represent the mean of the results of all configurations and as such are the most relevant when it comes to the general behaviour of the controllers. Considering the average results, LIMA performs significantly better in all aspects apart from eaten ghosts.

An analysis must also be conducted on the number of wins, specifically to explain why LIMA does not win any rounds. This issue is also connected to the longer lifetime (see number of moves) of the LIMA agent in most cases. When approaching the end of a round the two methods have different behaviours:

- **IMA** – the influence of the remaining rewards increases and the influence of the ghosts decreases, “blinding” the agent and making him go straight for the rewards without paying any attention to the ghosts which probably kill him on the way. This explanation is also supported by the fact that when approaching the end of the game the IMA agent always dies when there are between 3 and 5 pellets left.
- **LIMA** – the agent is usually situated in an influence of constant null influence, meaning that he navigates at random for some time (until he is near a ghost or a reward). This random movement is the cause of the larger number of moves made by the LIMA agent.

The fact that even though the LIMA agent manages to survive and sometimes win the final part of the round (even if this means spending some time navigating at random) proves that the purposed model is a balanced controller which adapts to most situations in the game.

IMA performs better when considering the number of eaten ghosts. This happens due to the fact that for the LIMA method edible ghosts have a positive influence equal to -1 × (inedible ghost influence). This helps reduce the number of parameters the method has, but also represents a disadvantage when compared to IMA (for which the influence of the edible ghosts can be set independently).

6 Conclusions and Further Research

This paper explores the concept of influence map based controllers for Pac-Man and also introduces the idea of limiting the propagation of influence in such a controller. The results of this approach are promising and it was proved that the method behaves equal or better when compared to other existing controllers. Limiting the influence propagation increases the number of parameters used by the method and by doing so helps build a balanced agent that can easily adapt to specific situations inside the game.

An influence map based controller for an agent has many applications in fields like computer simulations, computer games and robotics. The method developed in this paper is especially suited for situations that require a short reaction time. With regard to this, our tests have shown that when it comes to computing the entire influence map the purposed method is 89% faster than its predecessors. This large difference comes from limiting the area of influence of and therefore computing fewer values for each entity.

The only aspect where the purposed agent fell behind the previous implementation is capturing (or *eating*) enemy agents (*ghosts*). This is a setback determined by the use of a single value for setting the influence of both harmless and harmful agents. While it is not a significant flaw in the method this aspect can easily be improved by using separate values for the 2 types of agents mentioned before. This increases the number of parameters for the method but might prove to be what makes the purposed agent clearly superior to previous implementations.

The Limited Influence Map Agent (LIMA) method could also be improved by finding a new strategy for the scenario in which the agent is surrounded by constant null influence. In such a case the current agent chooses the next step at random (without being able to turn back). This is not a great issue for the method, and sometimes it proves to be an effective strategy, but a better approach is sure to shorten the time it takes the agent to complete a level.

In the near future we plan to develop and employ an automated tuning method for improving the parameters used by the method and therefore optimizing its behaviour.

References

[1] S. Lucas and G. Kendall. Evolutionary Computation and Games, *IEEE Computational Intelligence Magazine*, vol. 1, pp. 10-18, 2006.

[2] N. Beume, H. Danielsiek, C. Eichhorn, B. Naujoks, M. Preuss, K. Stiller, S. Wessing. Measuring Flow as Concept for Detecting Game Fun in the Pac-Man Game, *Proc. of the 2008 IEEE Congress on Evolutionary Computation (CEC 2008),* pp. 2447-3454, 2008.

[3] D. Nicolescu and C. Stoean. An Efficient Influence Map Based Controller for the Pac-Man Agent, submitted to *Journal of Knowledge, Communications and Computing Technologies (JKCCT),* 2011.

[4] N. Wirth and M. Gallagher. An Influence Map Model for Playing Ms. Pac-Man. *Proc. of the 2008 IEEE Symposium on Computational Intelligence and Games,* pp. 228-233, IEEE Press, Perth, Australia, 2008

[5] C. Stoean, M. Preuss, R. Stoean, D. Dumitrescu. Multimodal Optimization by Means of a Topological Species Conservation Algorithm, *IEEE Transactions on Evolutionary Computation,* vol. 14, no. 6, pp. 842-864, 2010.

Dragos Nicolescu
University of Craiova
Faculty of Mathematics and
Computer Science
Department of Computer Science
Str. A.I. Cuza, nr 13, Craiova
ROMANIA
dragosm.nicolescu@yahoo.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

MathCast – a universal prediction application

Dragoş Iulian Obancea
Teacher Coordinator: Lect. Dr. Vlad Monescu

Abstract

Since ancient times, people felt the desire to find out what the future holds for them. Although there are many contradictory opinions in this domain, one thing is for sure: there is no model or algorithm that can accurately predict future events and values. However, throughout history, people have developed various mathematical models, which were subsequently implemented into specialized software in order to give approximate numerical values, more or less close to real values, on certain aspects of the future. Creating a similar software would not bring anything new in this field, so *MathCast*, beside the provided support for adding and using mathematical models to predict the future, it also offers the possibility for users to create various scenarios and to build predictions in the form of a text, describing events that may occur with a certain probability on certain periods of time. Also, during the creation of this software, I have developed new mathematical models. These models are unpolished yet, but in some cases the values resulted surpass other possible values, resulted from any other type of algorithm.

1 Introduction

In the past, it was believed that predicting the future represents an art that only magicians can have access to, it usually occurs during a full moon, with the help of incantation and / or combinations of chemicals. With the advent of more advanced civilisations (e.g. Maya Civilisation), the filed had become „more scientific” by introducing some basic mathematical models for calculating, in most cases, the positions of stars and celestial objects. In this way the astrology was developed. The result was a complex calendar created by the Maya civilization, which describes with great accuracy, every important astral event that will take place even in 5000 years time.

Nowadays, there are many areas where people are using prediction/forecast methods, from classical weather forecast, to population and economic statistics of how the world would be in the next 50-100 years from now, or even more. Even personal life predictions are now very popular, but this is an area that does not belong to our interest.

Mathematics models achieve the objective of finding credible statements about certain values, that will describe different aspects in a given area, over a certain period of time we use. Over the past 300 years people have developed a series of such methods, which have the purpose of accurately approximate data sets that describe different future aspects. Unfortunately, the uncertain and unpredictable nature of the future (from our point of view) makes it very difficult for us to give a classification of these models. Also, it is impossible to write a demonstration that would prove, that a model is better or not than another. The practical field shows us that in different situations, we can have more exactitude than usual. The only sustainer of this claim is time, that will offer us the possibility of comparing actual data with the one predicted.

Although predicting the future is not an exact science and the results can have various interpretations, these models can lead to clear and concise statement, which cannot be contradicted, such as: *”Taking into account the amount of data recorded over the past 300 years and including our rate of growth, in 50 years we will reach the next value, which can lead to ...”.*

It can be anticipated the idea, that *MathCast* is a software that makes predictions and forecasts in almost any field. Therefore, when created, it was necessary for *MathCast* to be a framework for achieving, adding and developing, and using mathematical models for predicting the future. These would be all the similarities between MathCast and other software from this area.

Albert Bartlett [1], a well known professor from the University of Colorado described a new concept based on the following three aspects:

- Let us assume a situation where the data set reveals increasing/decreasing values.
- These values cannot exceed certain limits (upper / lower).
- Changing the rate of growth/decay leads to certain consequences, each of them with its own probability.

MathCast contains a component that, in my opinion, no other software in this area contains, which is a combination of these three aspects for making a text prediction, describing events that may occur in certain periods of time with a certain probability.

MathCast also provides opportunities for building scenarios based on data sets for which we know future values. These methods vary from changing the values to changing the percentage between the values.

Another noteworthy aspect of this software is that it also includes the possibility of adding new models for making predictions on data sets that depends on other data sets, for which we know future values (calculated values).

Displaying and saving data, as well as other important aspects will be described more fully in the chapters that follow.

2 Software Requirements Specification

The original objective of this project was the following: *The creators have to make a software that can combine all three aspects of the concept developed by Albert Bartlett, presented in the introduction.* This requirement implies that the following problems have to be solved:

- Establish an easy way to add algorithms that would contain mathematical models for predicting the future. They will also allow users to test the models on real data and view/save the results. For the moment, it is desired that the application should contain a reasonable number of prediction algorithms.
- Establish a way for adding further limits required for calculation. These limits may be constant or not.
- Establish a way for the introduction of the consequences in a table, with or without probability.
- Establish a way to combine the three modules mentioned above and return, a detailed text file containing a list of events that may occur at a certain time, with a certain probability.
- Establish a way to combine multiple files, from the previous module, and return another text file containing lists of multiple events that may occur at several intervals, with the associated probabilities.

To solve the above problems, there will be given the following input:

- Files (xml, txt, etc.) containing the data nodes known so far. Each node will be associated with a date, value and description. The nodes will be arranged in chronological series with equal time intervals.
- Files (xml, txt, etc.) containing data nodes (known or predicted), that represents a limit for the values presented in the preceding paragraph.

- Files (xml, txt, etc.) containing two types of events that may occur and the probabilities associated with them. In the first category there are events embedded that may occur if the last increasing/decreasing percentage will be ascending/descending, and the second category includes events that may occur if the percentage is on an descending/ascending line. In other words, in the first category, events are embedded for the cases where:

$$oldPercent < newPercent$$

and in the second category events are embedded for the cases where:

$$oldPercent > newPercent$$

In addition to these initial objectives, in the course of the development of the application, new targets have been introduced, among the most important we mention:

- A module that allows users to build scenarios based on data sets, that contains or not future data. Also, more algorithms expressing mathematical models can be easily added. A reasonable number of algorithms is required, from one that calculates the double of limits, to a framework for changing the values manually.
- A module that allows user to make predictions on data sets that depend on other data sets, for which future values are already known. To achieve these goals we need mathematical models transposed into algorithms. Of course, the usage of these algorithms must be easy and the software must contain at least two such methods.
- For each module that works with data set, both graphical types of representation are required (the simplified and the detailed modes). The detailed representation must be organised so that it can be transposed into an image that can be saved on the hard-disk.

The application is to be built in modules, so we anticipate that we need a „container" to incorporate these modules. This container will work with projects. Each project is built from files resulted from the modules mentioned above. The user interface and operating principle of the container will be similar to Microsoft Visual Studio 2010 (the framework in which this application was developed).

Given the fact that this software is still in the developing process and that some objectives were specified during its completion, we may say that as long as the evolution process continues, new requirements will be added and so the complexity of this application will increase.

The professional requirements of the software requires that, beside the objectives mentioned above and beyond those which will probably appear, a number of qualities must be insisted upon. The most important are: correctness, robustness, extensibility, reusability, performance, easy usage, universality and accessibility.

2.1 Correctness

Correctness is a very important property of software systems [2]. Testing the accuracy of prediction algorithms can be done only by comparing their results with real data, but we cannot put our finger on a specific criteria of comparison, because the phenomenon itself is not an exact science. Thus, some algorithms may be good in certain situations and less good in others. For this application, it is required to test the algorithms and mathematical models in different situations. Also, a documentation is needed, that presents the advantages and disadvantages.

2.2 Robustness

Clearly enough the robustness is the completion of correctness [2]. For this application it is required to test the component of each module in different situations in order to avoid further problems.

2.3 Extensibility

Extensibility is the ability of a software to easily adapt to changes in the specification phase [2]. This target has already been achieved by the fact that after expressing initial requirements, new objectives are developed.

2.4 Reusability

Reusability is the ability of software components to allow the creators to use them in other projects [2]. In this manner, even specifying the problem gives us a hand because it requires an application based on modules. So the modules have to be independent, in order for them to be used in other applications.

2.5 Performance

Although C# (the chosen language) is not the fastest one, speed is required that the algorithm used to be as fast as they could be and the process of drawing graphics to be very efficient.

2.6 Easy to use

There can be many types of users who can use this software. This application is designed for all social categories, from IT specialists, doctors and mathematicians up to journalists. Therefore, the success of such a system is directly proportional to the market success, and this aspect can be improved only through a friendly interface for the user. All in all, there is a requirement for this application to provide an interface, that needs minimal computer knowledge and that allows the user to be quicker and more efficient. It also requires the application to contain a text support, that gives explanations on how each component works.

2.7 Universality

Predictions can be made in any field, therefore it is necessary for the application to accept different types of data with different measuring units and different time intervals.

2.8 Accessibility

Achieving such a system that can calculate data quick and efficient, automatically leads to lower costs than when hiring mathematicians and IT specialist. Also, the results can be obtained in a shorter period of time.

3 Main results

It is a widespread idea that a systematical presentation of a paper, that contains short and concise sentences, determines the listener/reader to be more receptive to the concept described, as compared to a long and tedious presentation. Therefore, further in my paper I will divide it into small sections. So, we have our work divided into two different parts:

- The theoretical description presents mathematical models and their related algorithms.
- The practical perspectives presents the application in accordance with the requirements set out in Chapter 2.

3.1 Theoretical description

In order to fulfill the initial requirements, mathematical models were introduced that are meant to describe the algorithms, designed to calculate the values regarding different aspects of the future.

As I mentioned in the introduction, it is impossible to give a criteria for classifying these methods because predicting the future is not an exact science, so some methods might be appropriate in certain conditions, but others may be less appropriate.

Taking into account the ideas promoted by Albert Bartlett [3] I have introduced classical models taken from specialized books (about general Economy [4]) as well as new methods (see Chapter 6). For the moment, the new methods are not fully developed, but in some cases they have proved to be much closer to reality than the previous results.

Further on, I will talk about the models used in the application, without going into details in the case of the classical ones because for them it already exists an adequate documentation. This application contains:

- Models for the determination of future values in a data set
  - Lagrange interpolation: The polynomial function is determined, and with it we can calculate future values.
  - Last percent method: It is based on the last value in the data set and the percentage determined by the last two values.
  - Last of the last percent method: the last two percentages are calculated and then the percentage determined by those. The method is based on this value, the last value in the data set and the percentage determined by the last two values.
  - Pyramid method (section 3.1.1)
  - Extreme pyramid method (section 3.1.2)
  - Absolute Chronological Extrapolation [4]
  - Index Chronological Extrapolation [4]
  - Analytical Chronological Extrapolation [4]
- Models for determining future values of a data set, that depend on other data sets, in which we already know the future values.
  - System method: it creates a system from the last n values (n being the number of data sets) from each set of data, and so it determines how the objective data set depends on other sets. Next, this calculation becomes common.
  - Revised simplex method: it creates a system from the last n-1 values (n being the number of data sets) from each set of data. The objective function coefficients are the first set of future values for which they are known. This calculation algorithm gives the name of the method(revised simplex).
- Models for the development of scenarios based on a data set
  - Constant method: all future values are equal to the last known value
  - Double limit method: the limit is doubled for all future data
  - Manual data method: allows the user to manually change the values:
    - Value change
    - Percent change
    - Percent change from previous value
  - Manual percent method: allows the user to manually change the percent values:
    - Percent value change
    - Percent percent change
    - Percent change from previous percent value

All the presented methods for determining future values from a set of data have been adapted, to be used in the cases where the intervals are less than a year, when there is a periodicity in the data set (For example, when patients have got a flu, there are fewer cases in summer than in winter). Also, the last three models were once again adapted to calculate a score line of data values.

In regard with the development of the application, I will introduce other models and I will try to improve the existing ones (especially the *Pyramid method* and the *Extreme Pyramid method*).

3.1.1 Pyramid Method

The mathematical model of this method (see figure) is as follows:

Let *n* inputs:

$$a_1, a_2, a_3, \dots, a_n$$

Each of these values are associated with dates in chronological series:

$$t_1, t_2, t_3, \dots, t_n$$

It requires finding the value a_m associated with t_m , $m>n$. First, the values $a_{n+1}, a_{n+2}, a_{n+3}, \dots, a_m$ will be calculated.

In order to find the value of a_{n+1} the next steps will be followed:

We build the matrix *X* with *n-1* lines. The items on each line will be added as follows:

- On the first line we will add *n-1* elements following the next formula:

$$x_{1j} = \frac{100 \times (a_{j+1} - a_j)}{a_j}$$

- On each line *i, i>1* we will add *n-i* elements following the next formula:

$$x_{i,j} = \frac{100 \times (x_{i-1,j+1} - x_{i-1,j})}{x_{i-1,j}}$$

Next, for each line in the matrix X, beginning with the last one till the first one, we will add an element as follows:

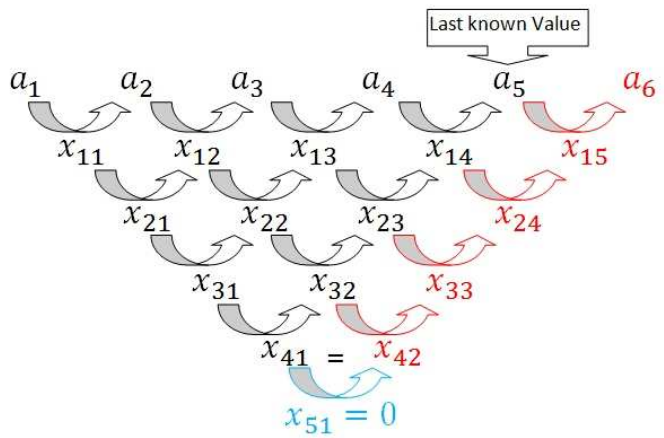
- In the first iteration (on the last line), the item added will be equal to the last item on that line (which is unique).
- On each line *i, i<n* we will add one element following the next formula:

$$x_{i,j} = x_{i,j-1} + \frac{x_{i+1,j-1}}{100} \times x_{i,j-1}$$

Finally, a_{n+1} will be calculated as:

$$a_{n+1} = a_n + \frac{x_{1,n}}{100} \times a_n$$

Similary, other values are calculated until a_m is reached.



In the tests I have made using this method, I have reached the conclusion that, although in some cases the method is unstable, in other ones, the resulting values are much closer to real values than in any other method I have used. In case of further development, I will try to stabilize this method, or at least to detect data sets for which the model does not work correctly.

3.1.2 Extreme Pyramid Method

The mathematical model of this method (see figure) is as follows:

Let n inputs:

$$a_1, a_2, a_3, \dots, a_n$$

Each of these values are associated with dates in chronological series:

$$t_1, t_2, t_3, \dots, t_n$$

It requires finding the value a_m associated with t_m , $m>n$. First, the values $a_{n+1}, a_{n+2}, a_{n+3}, \dots, a_m$ will be calculated.

In order to find the value of a_{n+1} the next steps will be followed:

We build the matrix X with $n-1$ lines. The items on each line will be added as follows:

- On the first line we will add $n-1$ elements following the next formula:

$$x_{1j} = \frac{100 \times (a_{j+1} - a_1)}{a_1}$$

- On each line i, $i>1$ we will add $n-i$ elements following the next formula:

$$x_{i,j} = \frac{100 \times (x_{i-1,j+1} - x_{i-1,1})}{x_{i-1,1}}$$

Next, for each line in the matrix X, beginning with the last one till the first one, we will add an element as follows:

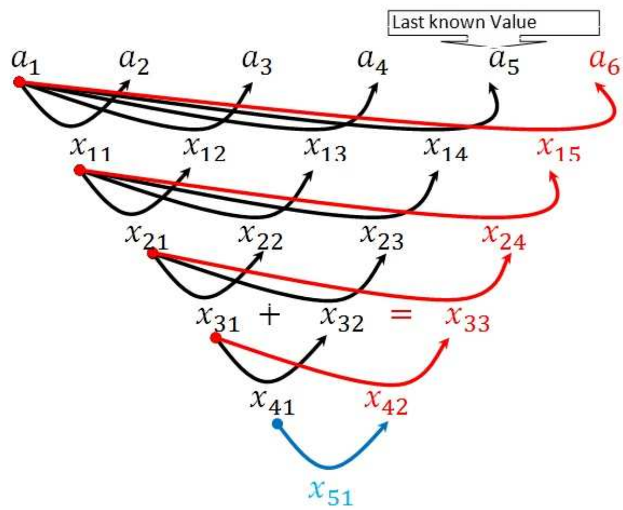
- In the first iteration (on the last line), the item added will be equal with the sum of the first two items on that line.
- On each line i, $i<n-1$ we will add one element following the next formula:

$$x_{i,j} = x_{i,1} + \frac{x_{i+1,j-1}}{100} \times x_{i,1}$$

Finally, a_{n+1} will be calculated as:

$$a_{n+1} = a_1 + \frac{x_{1,n}}{100} \times a_1$$

Similary, other values are calculated until a_m is reached.



In the tests I have made using this method, I have reached the conclusion that, although in some cases the method is unstable, in other ones, the resulting values are much closer to real values than in any other method I have used. I also noted that the method detects the frequency values to a certain degree. Contrary to how other methods work, this one doesn't tend to eliminate it. In case of further development, I will try to stabilize this method, or at least to detect data sets for which the model does not work correctly.

3.2 Software Description

3.2.1 Modules

According to the objectives outlined in Chapter 2, the application was built on modules. Each module is actually a class that implements the *Form* interface (a window). The application has, so far, seven such components, each with different functionality:

- *Data Interpolation* – is probably the most used module, and it contains a series of tools and methods of prediction based on a single data set.
- *Multi cast* – contains a number of tools and methods of prediction based on a set of data that depends on other sets.
- *Scenario* – contains a series of tools and methods that can be applied on data sets to achieve scenarios.
- *Consequently Table* – contains a series of tools and methods available to the user for making events and associated probabilities table, specified in the objectives.
- *Limit Forecast* – contains a series of tools and methods available to the user for combining three components: a prediction, a limit and a table of consequences. This module returns a text file containing a list of events that may occur at a certain time, with a certain probability.
- *Combined Limit Forecast* - contains a series of tools and methods available to the user for combining two or more files from the previous module. This module returns a text file containing one or more lists of events that may occur at several intervals of time, with certain probabilities.
- *Multi Chart* – allows users to list one or more files, results of the first mentioned module, for their graphical representation.

In addition to the functionality mentioned above, each module offers the possibility of saving the results in different types of files and allows you to view data in a graphical representation.

3.2.2 Modules Container

Since the application was built in modules, it is anticipated that the container is required to incorporate them. This provides a framework in which each module behaves like an independent window of Windows. Besides this, the module container also offers the following possibilities:

- Creating a new project. A project is a collection of files in the same directory and is represented by a file with *".mtc" (MathCast)* extension.
- Saving, deleting and editing existing projects.
- Solution explorer. The software contains a tool that detects file types and displays them in a *tree-view* structure. Each node of the structure, representing a file, can be opened if the application knows that type of file. For example, if the file is the result of one of these modules, then the module will be detected and used to open it.
- Opening and closing modules.
- View a calendar.

- All the above components can be „shifted” in the main window, they can be modified or removed, depending on the users preferences.

The module container was designed to resemble the software that was used to create this application.

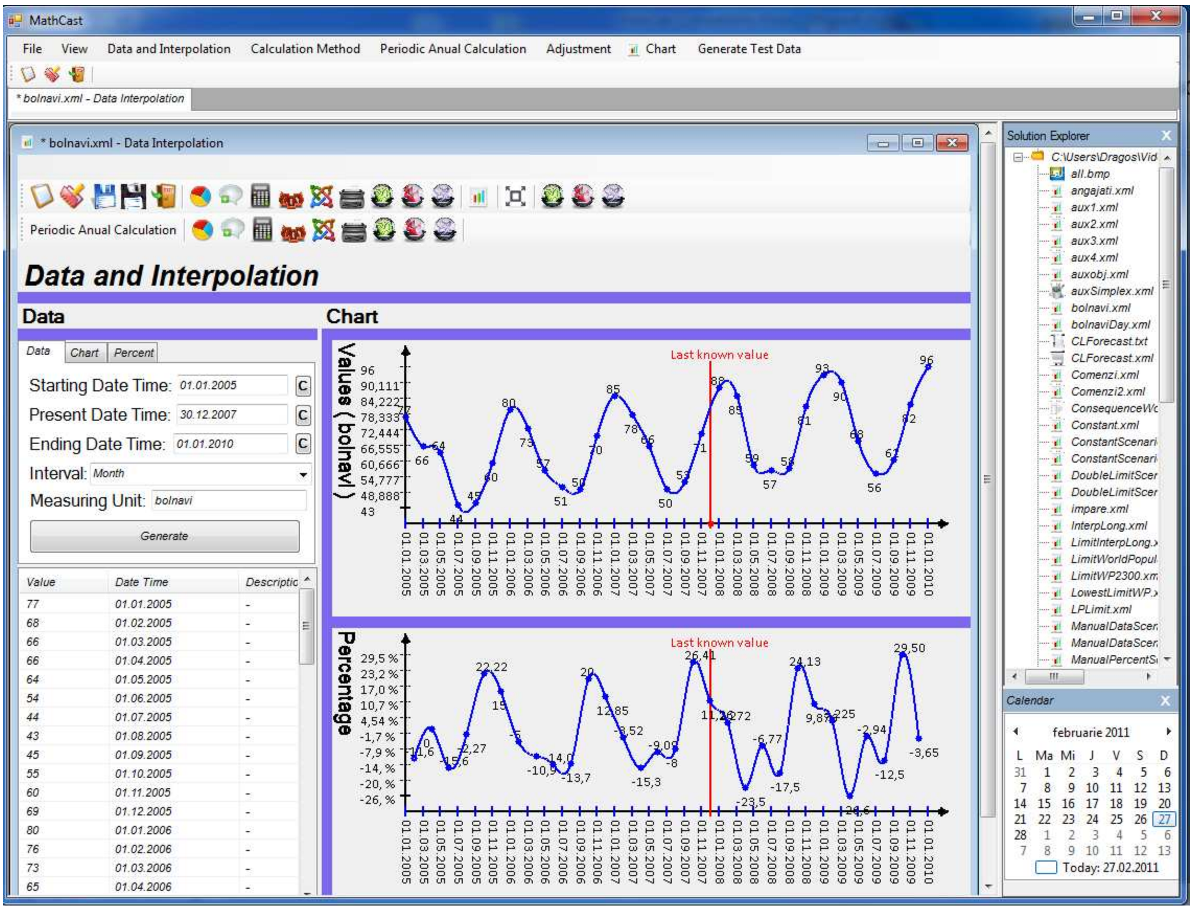
3.2.3 User

From a practical perspective, the user horizont is as wide as the different fields in which this application can be used. From IT professionals to doctors, economists and journalists, the software tools are dedicated to whoever may need them. For this reason, great attention was paid to the process of implementing the user interface.

From a theoretical point of view, there is only one type of user, the final user, who may use the application without requiring a login. To add new algorithms and mathematical methods for calculating data, it is necessary for the intervention of a programmer, even if the code’s structure is as easy as 123.

3.2.4 User Interface

I have opted for a less complicated user interface, because simplicity is one of the most important factors contributing to the success of introducing a new product on the market.



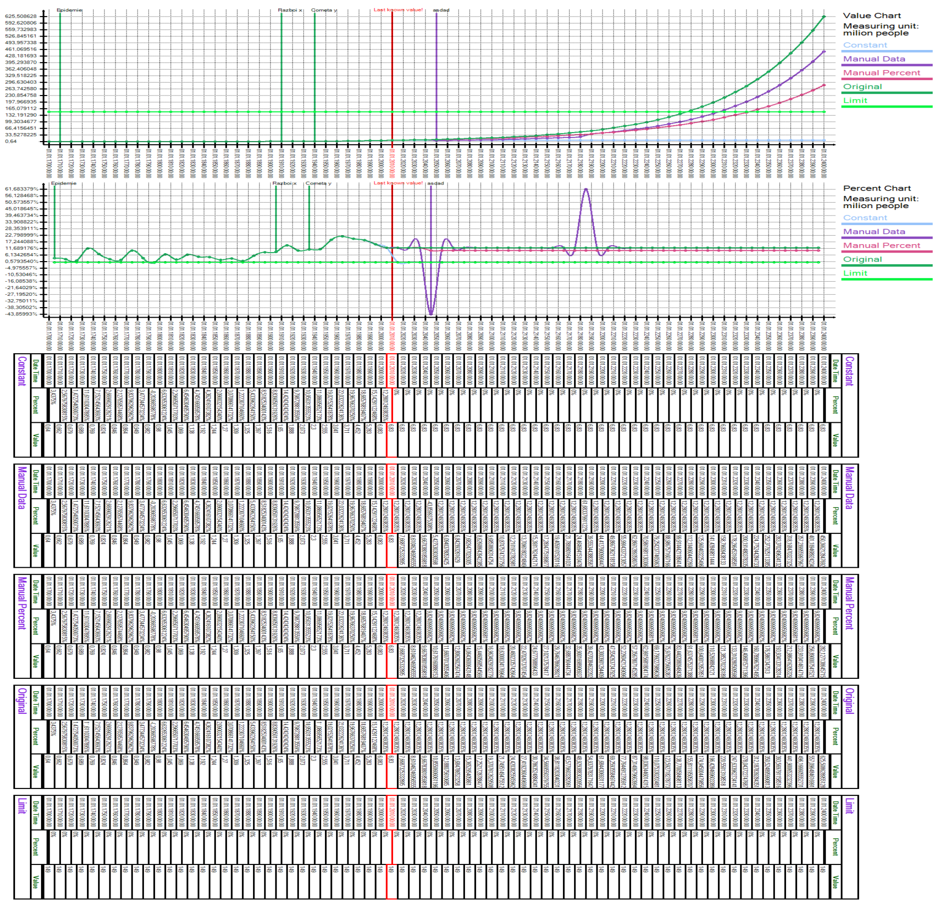
3.2.5 Saving data

As we noted in section 3.2.1, each module offers the possibility of saving data. Data saving begins when creating and saving a project, by the automatic building of the directory and file with *".mtc" (MathCast)* extension. With the help of the container and its modules, the application can detect and save three different types of files:

- Files with *".txt"* extension. These files are text forecasts and are generated by the modules: *Limit Forecast* and *Combined Limit Forecast*.
- Files with *".xml"* extension. They are generated by the other modules. For each module, there is another type of *".xml"* file (different type of data contained and different root names); such types are automatically detected. There will be no list of these types mentioned, because of the limited space in this paperwork.
- Image files. These are graphical representations of data and can be opened with a specialized component (Graph) which is the true graph generator.

3.2.6 Graphic representation

A particular problem was creating a proper representation of graphics, that can be understood by any user. The degree of difficulty of this objective was high because of the demand of detailed representation of data sets, that may or may not have the same length of time and measure. So far, the results have been above expectations. As we can observe in the following chart, the results are „drawn" in an image that can be saved in various formats. Also, in the process of drawing the graph, you can select the components that you want to be visible.



3.2.7 C# Language

C# language was preferred because it is an object-oriented programming language, that is based on C++ syntax and includes aspects of the languages *Delphy, Visual Basic* and Java with a particular emphasis on simplification (less symbols then in C++, and less decorative requirements than Java, etc.) [5, 6].

Each module presented in section 3.2.1 is a class that implements the *Form* interface.

Mathematical models, that have been transposed into algorithms, have been collected in *Tools'* class.

Other methods and models, necessary for different actions related to the functionality or appearance of the application, were placed in classes such as: *FileExplorer, DockExtender, Nodes,* etc.

Because of the limited space in this paperwork, examples code will not be given.

4 Conclusions and future developments

Predictions is a vast field that can be applied in many aspects of daily life. Therefore, the possibility of developing such a computer software is infinite, as well as its usefulness. If we are to think about it, even the science that deals with this aspect of mathematics/computer science is actually taking its first steps. So, because of the growing desire of people to find out what the future holds, development prospects are bright.

Through the software that was build and presented in this paper, we managed to achieve the objectives mentioned in chapter two, and also to introduce other features designed to facilitate the work of the user.

As mentioned in the paragraph above, „the power" of this application is given by the possibility of continuous development. This possibility is encouraged by the modularity of components. From my point of view, the current application represents only a starting point in shaping a complete computer software, that will be successful on the market. If this application is appreciated and if there are opportunities for further development, I propose myself the following goals:

- Developing new modules with new functionality;
- Developing of the existing modules;
    - Developing of new algorithms and calculation methods;
    - Intensive testing and error correction that may occur during testing;
    - Creating a more professional and easier interface to use;
- Making an interactive Help;
- Developing a more complex graphic representation;
- Developing and improving the module container interface;
- Usage of the application in real situations, made by specialists in different fields (medicine, meteorology, economics, etc.)

5 Notes

This paperwork was completed on 27/02/2011. Therefore, there is a high probability that the application have been changed. Generally, these changes may include adding new modules or computational models.

The algorithms presented in previous chapters (*Pyramid* and *Pyramid Method*) are my own ideas, but this does not exclude the possibility that they might have been formulated before. After a thorough search, I have not found yet, other algorithms like my own, but I will keep on searching.

References

[1] Albert Bartlett, *Arithmetic of Growth Methods of Calculation, Population and Environment,* March, 1993

[2] Dorin Bocu, *Ingineria Softului,Cur*s, 2008

[3] Albert Bartlett, *The World's Worst Population Problem – Is There a Population Problem?,* 1998

[4] Doina Maria Simion, *Serii cronologice. Indici statistici.,* "Lucian Blaga" University of Sibiu Press, 2001

[5] Dragoş Iulian Obancea, *Primii paşi cu C#*, Editura Noua Bucureşti, 2009.

[6] Lucian Mircea SASU, *Introducere în limbajul C#,* Transilvania University of Bra□ov Press, 2009

DRAGOŞ IULIAN OBANCEA
Transilvania University of Braşov
Faculty of Mathematics and Computer Science
Computer Science specializaition
Street Iuliu Maniu, Nr. 50, Braşov
ROMÂNIA
obancea_dragos@yahoo.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Companion – An intelligent agent architecture

Adrian Tudor Pănescu
Teacher Coordinator: Dan Cristea

Abstract

This paper presents the main aspects regarding the architecture of an intelligent agent named Companion. It is intended to support its user (Master) as an adviser for several daily life aspects. To obtain a flexible smart operation more methods and tools from Artificial Intelligence are used. Two main components of the proposed layout are discussed, namely the knowledge base and the decisional system.

The knowledge representation scheme uses an ontology that considers both the information specific for the Master, and certain issues that are necessary to support the decisional mechanism. Thus, taking into account that planning is a key aspect for Companion, the ontology and its corresponding transposition into a knowledge base are designed to provide the pieces of information needed by the planning mechanism. Various planning aspects can be taken into account, the paper describing the basic points for a temporal planning method. This is based on a proper time representation and an adapted planning procedure that operates in the plan space. Such an approach is favored by the combination between rule based and object oriented programming, which is supported by the chosen programming tool, CLIPS. Moreover, this solution facilitates an easy link between the ontology (built in Protégé) and the knowledge base the inference engine operates on. The paper underlines the advantages of the proposed solution and different possibilities of extension.

1 Introduction

Artificial Intelligence (AI) is a research field that is still expected to provide solutions for several application areas. One such field regards the intelligent assistance requested by human users. Though from the hardware point of view a continuous improvement can be noticed, the software part is not as advanced, yet. A possible path for improvement is represented by the combination of multiple methods and tools specific for AI.

This contribution proposes an architecture for an intelligent agent (called Companion) capable of helping the user (this will be further named Master) in different scenarios in his daily routine. These can include planning various activities, helping in taking certain decisions (suggesting a route, a show to watch, a book to read), resource management (financial issues, inventory of household items, etc.), or keeping a journal regarding different aspects of Master's life, like health issues. Even if some solutions for assistant agents have been already proposed (see [1], [2]), they are focused on certain aspects like the interaction within the Internet, or some well delimited decisional issues. In this way the problem of designing and implementing a software assistant is still open, and the main contributions of this paper regard the organization of the scheme for knowledge representation, the operation of the planning component and the link between these two subsystems.

The fundamental part of the Companion is represented by the decision system pictured in Fig. 1. The main components are an ontology, a knowledge base and multiple planning modules. The ontology was built using Protégé, an open-source Java-based platform for modeling ontologies, developed at Stanford University [3]. Because the planning modules rely on the rule based programming paradigm, the knowledge base was programmed in CLIPS. This is a programming environment developed at the Software Technology Branch (STB), NASA/Lyndon B. Johnson Space Center, providing a forward chaining inference engine [4]. Another reason for choosing CLIPS is the way it combines rule based and object oriented programming, facilitating the connection between the ontology and the planning modules. The link between the ontology and the knowledge base is assured by an auxiliary CLIPS Converter. There are also some complementary components, like outside sources (these can be connected with various sensors, not represented in Fig. 1) that supply the ontology with information by also using, when required, a Natural Language Processing module and a User Interface.

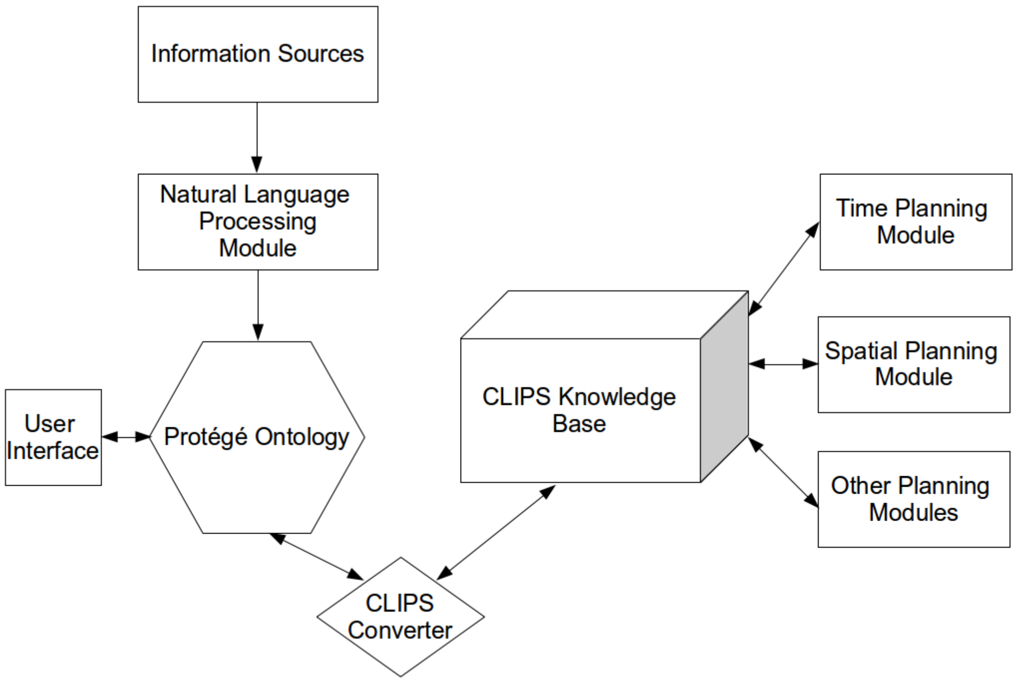


Fig. 1. The general architecture of the Companion's decision system

This paper is organized as follows. Section 2 details the used ontology, Section 3 the CLIPS knowledge base and Section 4 a planning module, namely the one regarding the temporal inferences. The paper ends with few conclusions and further development possibilities.

2 Design of the ontology

In Computer Science an ontology is a formal representation of knowledge as a set of concepts within a domain and the relationships between those concepts. Ontologies model concepts and relationships in a high level of abstraction, providing rich semantics for humans to work with and the required formalism for computers to perform mechanical processing and reasoning [5].

To describe the used ontology we will consider several notions similar to those in Object Oriented Programming. The concepts are represented using classes, which can have zero or several slots (properties, fields); these slots can be primitive types like strings, integers or instances of other defined classes.

The “root” concept is the *Person* class, the instances of which are representing the Master, the user of the Companion, and other entities that are in various relationships with him. This class has several slots like *profession***,** which is an instance of the *Profession* class, *hobby***,** which is an instance of the *Leisure Activity* class and *agenda*, an object which will store the schedule of the considered person, instance of the *Agenda* class. It should be noted that the *profession* and *hobby* slots can have zero or more values (as a person can have zero or more professions and hobbies), but the *agenda* slot is required

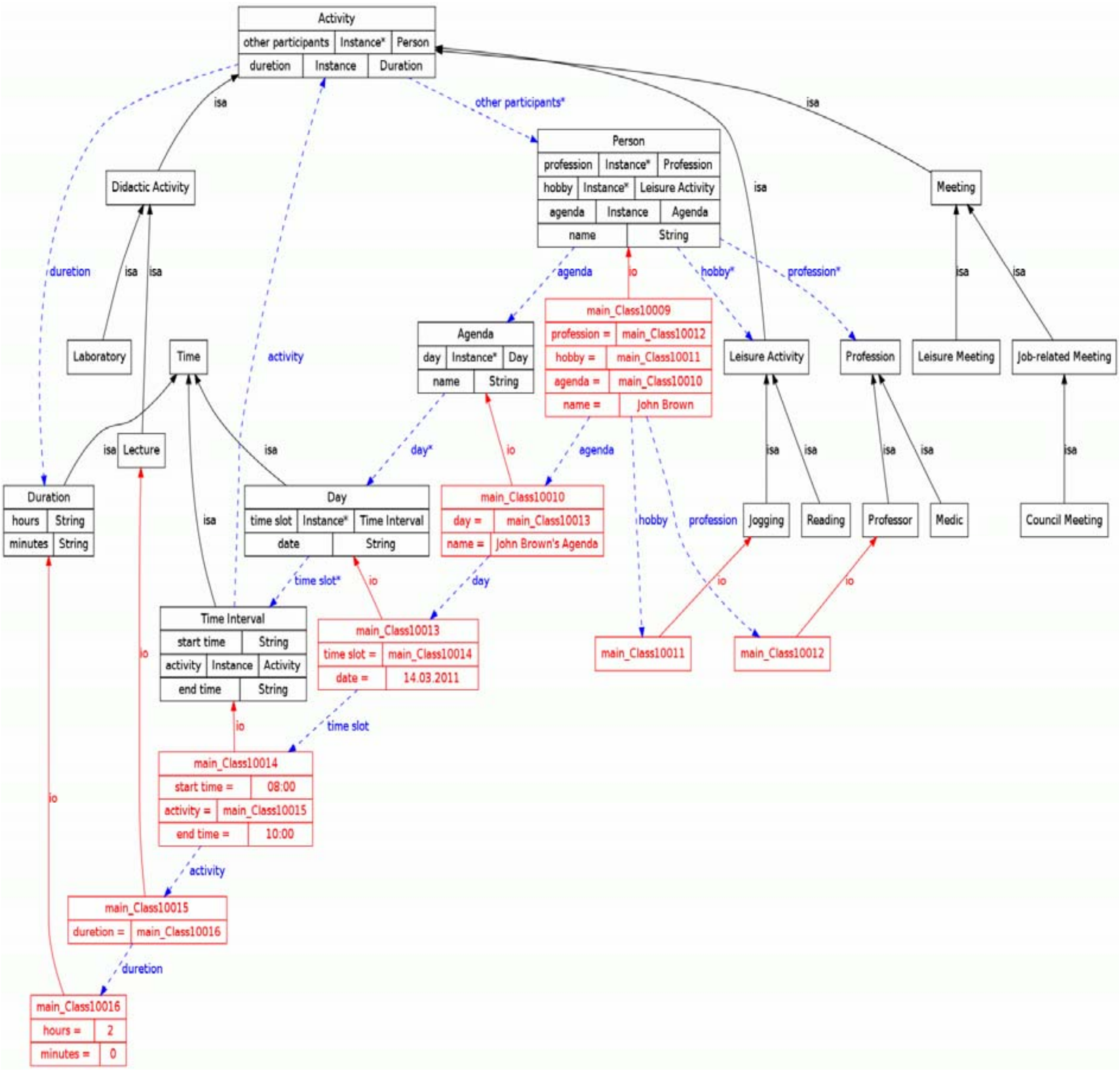


Fig. 2. Part of the Companion ontology

and unique. Slots that can have multiple values are marked in Fig. 2 with a star. The *Recreational Activity* class is a subclass of the more general concept *Activity* (this kind of relationship is named "*is a*" and it is marked accordingly in Fig. 2). At the same time *Jogging* and *Reading* are subclasses of *Recreational Activity*, as they are derived particular concepts.

In order to specify the Master's agenda using the *Agenda* class, and to be able to conduct temporal planning, some concepts about time are also included. The main class *Time* has several subclasses like *Day* (an agenda consists of an unlimited number of days), *Time Interval* (in a day there can be multiple *Time Intervals*) and *Duration* which describes what is the needed amount of time for each activity; of course, there can be activities with no specified duration.

In order to better understand this hierarchy and the relationships between classes a use case is described using appropriate instances, as showed in Fig. 2. This scenario can be summarized as follows: *John Brown*, the Master of the Companion, is a *Professo*r who has *Jogging* as a hobby. His agenda records that on the *14th of March*, between *08:00* and *10:00* he has a *Didactic Activity* of type *Lecture*.

3 Ontology conversion into a knowledge base

In order to apply the needed time planning techniques described in the Section 4, the ontology had to be converted in a format specific for CLIPS. Protégé already includes an option to export ontologies to the CLIPS format by creating 2 files. The first one, with the *.pont* extension contains all the definitions of classes ready to be loaded in the CLIPS engine. For example, the *Person* class as modeled in the ontology is translated to the following definition:

```
(defclass Person
        (is-a USER)
        (role concrete)
        (single-slot name_
                (type STRING)
;+      (cardinality 1 1)
        (create-accessor read-write))
        (single-slot agenda
                (type INSTANCE)
;+      (allowed-classes )
;+      (cardinality 1 1)
        (create-accessor read-write))
        (multislot profession
                (type INSTANCE)
;+      (allowed-classes Profession)
        (create-accessor read-write))
        (multislot hobby
                (type INSTANCE)
;+      (allowed-classes Leisure+Activity)
        (create-accessor read-write)))
```

In the above example certain slots in the original model were excluded (lines starting with a semicolon) as there is no CLIPS equivalent for them. Moreover, a distinct need regards the matching of instances on the rules' patterns, as shown in the next section. This supposes a certain specification for the CLIPS classes whose instances have to be involved in the rules' matching process, and such a condition is not provided by the Protégé ontology [4], [6]. These special cases need to be treated separately.

The instances of classes are exported in a file with the *.pins* extension. Unlike the definitions for classes, those for instances need a certain transformation before loading them into CLIPS, namely wrapping them in a *definstances* construct. As an example, the "*John Brown*" instance of the *Person* class is translated to the following definition:

```
([main_Class10009] of  Person
      (hobby [main_Class10011])
      (name_ "John Brown")
      (profession [main_Class10012])
      ( [main_Class10010]))
```

4 The temporal planning module

As already explained, a planning component of the developed system aims at obtaining a temporal reasoning mechanism. For such a purpose a specific approach must be devised, because neither of the classical state space or plan space planning methods are entirely appropriate [7]. The starting point has to be a suitable representation scheme for the temporal entities. By making use of the symbolic calculus provided by the Point Algebra, one can organize the pieces of knowledge containing time information under the form of a temporal knowledge base. According to [7], this can be a pair:

$$\Phi = (F, C) \tag{1}$$

where F is a finite set of temporally qualified expressions (tqes) and C is a set of constraints; these can be both ordinary constraints, as usually considered in Constraint Satisfaction Problems [8], and temporal constraints. Making the connection with the previously presented ontology, one tqe is:

$$Jogging(Person1)@[t_0, t_1] \tag{2}$$

meaning the activity named *Jogging*, performed by the entity *Person1* takes place in the interval between the time points t_0 and t_1. Logically, the expression (2) says that $\forall t$ such that $t_0 \leq t \leq t_1$ the relation *Jogging(Person1)* holds at time *t*. *Jogging()* is to be understood as a first order predicate. Examples of constraints within the set C are: $Person1 \neq John\_Brown$, $t_0 < t_1$ with the second one being a temporal constraint. The instances and the slot values in the constructed ontology allow the obtaining of the corresponding temporal knowledge base Φ, as needed for the temporal planning component.

Temporal planning operators are needed; they may be derived and adapted from the classical planning theory. For example, the temporal planning operators can have a STRIPS like form [7]. This means a planning operator is an activity to be included in a plan in order to achieve a goal, and that is characterized by its preconditions and effects. In our case these have to be considered as tqes too, as exemplified below for the *Lecture* operator. It refers to the activity of lecturing, conducted by a certain professor (*P*), to a class of students (*S*), into a certain location (*H*), and which is to be carried out in the interval between the time point t_s and t_e.

$$
\begin{array}{lll}
\multicolumn{3}{l}{Lecture(P, S, H, LC, T)@[t_s, t_e]} \\
& Precond: & At(P, H)@(t_1, t_s] \\
& & At(S, H)@(t_2, t_s] \\
& & Free(H)@(t_2, t_s] \\
& & Include(T, LC) \\
& & Assigned(P, LC) \\
& Effects: & At(P, H)@[t_s, t_3) \\
& & At(S, H)@[t_s, t_4) \\
& & Unfree(P)@[t_1, t_3) \\
& & Unfree(H)@[t_1, t_4) \\
& & Unfree(S)@[t_1, t_4) \\
& & Lecture\_Delivered(T, LC)@[t_e, t_3) \\
& Constraints: & t_2 < t_1 < t_s < t_e < t_3 < t_4
\end{array}
\tag{3}
$$

It is to remark that besides the usual aspects of the STRIPS approach the temporal information is present both in the tqes used to describe the preconditions and effects, and into the added planning operator component, i.e. into *Constraints*. These are necessary to carry the information on the chronology to be complied with, so that the *Lecture* operator should be included in a plan to fulfill a goal. In our case the goal regards that the lecture with the code *LC*, included in the students' timetable *T*, and assigned to the professor *P* must be already delivered at the time point t_e. The planning methodology combines tqes with normal expressions. For example, it is needed that the professor *P* should be at the hall *H* sometime before starting the lecture, while the fact that the lecture with the code *LC* is included into the student timetable *T* is an expression of an un-temporal kind. As temporal planning avoids using negated atoms (it relies on the closed world assumption [9]), it was necessary to explicitly state the information on the conditions of not being free for the various entities involved in the modeled activity, during certain time intervals. As expression (3) shows, both closed and semi-open intervals are used; the starting of the lecture is a sharp time point, while the moment when professor leaves the lecture's location is loosely specified, being a time point after the lecture end (t_e) and before the moment when students leave the location (t_4). Again is to observe how the considered planning operator is consistent with the developed ontology, the Companion agent's knowledge base providing the needed pieces of knowledge.

Without giving all the theoretical details, the above introduced concepts allow us to define a temporal planning problem as:

$$\mathsf{P} = (D, \Phi_0, \Phi_G) \tag{4}$$

where *D* is the domain knowledge base that contains general information on the universe of discourse, including all the possible temporal planning operators, Φ_0 is the temporal knowledge base containing data on the initial state of the world, and Φ_G is the temporal knowledge base specifying the information on the goals to be undertaken. The planning procedure must find the set of actions (these are instances of the planning operators) so that when they are applied in Φ_0 this entails Φ_G. The applicability of an action in the conditions defined by a temporal knowledge base is to be judged in the same way as in the STRIPS methodology, by further considering the temporal constraints included in the tqes. The planning procedure results as generalization of the plan space planning scheme [7], conducting to a nondeterministic mechanism, known to raise an NP-complete problem. Nevertheless, with some adaptations a suitable implementation is expected to deliver satisfactory results for certain applications.

The present proposal is for applying a forward chaining mechanism, which is possible in a CLIPS implementation. As already presented, the initial Protégé ontology is transformed into a CLIPS knowledge base with classes and instances. Benefiting from the way the CLIPS rules can be activated by instances, the rules to solve the agent planning component can be written. Due to the fact that CLIPS has a forward chaining inference engine and the planning method is supposed to start from goals, these will be included into the conditional rules' parts, together with the preconditions of the planning operators. As an illustrative example, the following rule is activated when there is a goal to plan a lecture, identified by its title, whose code is found in an instance of the *TimeTable* class, being assigned to a professor identified by his name. All the restrictions on the professor, students, hall availability and their time conditions are easily checked through the pattern matching mechanism of the rules on the instances existing in the CLIPS knowledge base.

```
(defrule R
    (object     (is-a Goal)
                (type lecture)
                (title ?t)
                (interval ?t1 ?t2))
    (object (is-a Professor)
                (name_ ?n)
                (at ?nh)
                (interval ?x1&:(< ?x1 ?t1) ?))
    (object (is-a Lecture)
                (title ?t)
                (code $?c))
    (object (is-a Hall)
                (name_ ?nh)
                (state free)
                (interval ?y1&:(< ?y1 ?x1) ?))
    (object (is-a Student_Class
                (class_id ?id)
                (at ?nh)
                (interval ?w1&:(< ?w1 ?x1) ?))
    (object (is-a TimeTable)
                (cc $?z)
                (assigned $?  ?n ?t))
    (test (subset $?c $?z))
 =>
    (printout t " Professor name " ?n " Lecture title " ?t " Class identifier "  ?id
     " Interval " ?t1 "-" ?t2 " Hall name  " ?nh crlf) )
```

The correspondence between the expression (3) and the above rule is evident, except for the effects of the activity that, for the sake of simplicity, were not included within the rule. Of course, the activity effects must be considered in order to keep the agent knowledge base updated and to keep track of the time evolution. As mentioned, one must take into account the nondeterministic character of such a planning procedure. When implemented in a rule based fashion this can be handled by an appropriate control of the conflict resolution phase (this can conduct the search until traversing a whole branch), together with a proper control mechanism to determine the passing to the next branch. The knowledge base consistency checking is also related with the planning scheme. This raises another NP-complete problem which regards the checking on the variable bindings (mainly, the actions' parameters are to be checked). Even so, the developed planning mechanism and the ontology it uses are focused on the temporal aspects; as these are modeled by the means of the Point Algebra it is possible to obtain a consistency checking procedure of complexity $O(n^3)$ for the temporal variables, that may be acceptable for the common tasks of the Companion.

5 Conclusions and development possibilities

The developed architecture presents enhanced reasoning capabilities as resulted from the combination between an ontology, a knowledge base and an appropriate planning mechanism. The rule based environment allows an efficient implementation of the theoretical background for temporal reasoning. Both the Protégé ontology and the CLIPS definitions are general enough, allowing the modeling of a wide range of scenarios, as needed by the Companion's users. Even so, in case the user may want to register some data that would not fit under any of the already defined concepts, the system will place them in a special area in the ontology that does not take part in the planning process. As an improvement, there could be developed a procedure to transform this additional information into the patterns acceptable for the CLIPS knowledge base. For the decisional system, the fusion between the plan space approach, the suitable representation of time and a forward chaining inference engine is a solution that can manage complex use cases. The original points of this contribution regard the developed ontology, the way it contains data (facts and relations) needed for temporal reasoning, the efficient choice for the knowledge base (a mixture between rules, facts, objects and instances, as possible in CLIPS) and the specific design and implementation of the temporal reasoning scheme. This shows that an improved software performance is obtained when more AI instruments are properly connected and used.

The presented system can be extended to include various information sources (like audio/video channels), or other planning modules that may use these types of information. Moreover, the communication between several such Companions can be considered. In such a case, the decisional system should include a negotiation module, to be connected with the planning units. The discussed architecture can support such improvements, as the flexible design can easily integrate them.

References

[1] R. J. Keeble and R. D. Macredie. Assistant agents for the World Wide Web intelligent interface design challenges, *Interacting with Computers,* Volume 12, Issue 4, 357-381, February 2000.

[2] J. Oh, F. Meneguzzi, K. Sycara and T. J. Norman. An agent architecture for intelligent information assistance, *Annual Conference of ITA*, London, 2010.

[3] http://protege.stanford.edu

[4] J. Giarratano and G. Riley. *Clips Manuals*, version 6.0, Software Technology Branch, Lyndon B. Johnson Space Center, NASA, 1993.

[5] M. Golemati, A. Katifori, C. Vassilakis, G. Lepouras and C. Halatsis. Creating an Ontology for the User Profile: Method and Applications, *Proceedings of the First IEEE International Conference on Research Challenges in Information Science (RCIS)*, Morocco, 407-412, 2007.

[6] K. Pooloth and R. Jetley. Integrating the CLIPS Rule Engine with Protégé, *AIML 05 Conference*, Cairo, Egypt, 134-140, 2005.

[7] M. Ghallab, D. Nau and P. Traverso. *Automated Planning. Theory and Practice*. Morgan Kaufmann, Amsterdam, 2004.

[8] E. Tsang. *Foundations of Constraint Satisfaction*, Academic Press, 1993.

[9] R. Brachman and H. Levesque. *Knowledge Representation and Reasoning*, Elsevier, 2004.

ADRIAN TUDOR PĂNESCU
"Alexandru Ioan Cuza" University of Iaşi
Department of Computer Science
16, General Berthelot, Iaşi, 700483
ROMÂNIA
E-mail: adrian.panescu@info.uaic.ro

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Parallel algorithms for snow mapping based on satellite remote sensing

Petar Marinov
Teacher Coordinator: Milena Lazarova

Abstract

The paper presents parallel models for snow cover detection based on satellite remote sensing of the Earth. Two different algorithms are selected for parallelization that use normalized difference visible/infrared snow cover index and microwave scattering indices calculated based on satellite multi-spectral data. Both data and functional decomposition using message-passing (MPI), multithreading (OpenMP) and hybrid (MPI+OpenMP) parallel models are suggested and implemented. The experimental evaluation of the performance parameters is made on a heterogeneous compact cluster and the speedup results achieved show good scalability of the parallel algorithms in respect to both the processed data and the high-performance computer platform.

1 Introduction

Earth and its atmosphere are objects of observation by mankind so that he can increase his knowledge of the world in which he lives. A great number of individuals, scientists, organizations, governments, and members of academia are working independently and collaboratively, utilizing state-of-the-art technology and equipment such as Space Remote Sensing. Acquiring a greater understanding of the planet is due in order to find that proper and effective solutions to problems can be sought. Having more accurate and detailed information at our disposal allows finding a more legitimate solution. Remote Sensing from space satellites provides such detailed information [1, 2, 3, 9].

The recent use of latest-generation sensors in airborne and satellite platforms is producing a nearly continual stream of high-dimensional data, which, in turn, is creating new processing challenges. Computational requirements of time-critical applications are addressed by incorporating high performance computing models in remote sensing missions [4-7].

As the most important and most complex Geo-Information, remote sensing data can be fast processed with cluster-based parallel computation technologies. Parallel computation is in order because of the great number of satellite sensors, the multitude of different problems requiring different processing, different applications and objectives [5].

Geo-Information is used for Early warning systems for natural calamities, observation and valuation of the environment and solving environmental problems, climatic processes, etc.

This paper presents parallelization of different algorithms for detecting snow cover from satellite multi-spectral data. Satellite snow detection is practice for decades, but new high-dimension sensors provide greater amount of data, thus the need of speed up.

2 Parallelization of snow mapping algorithms

The study of atmospheric and climatic variability on a global scale requires monitoring of the snow cover component, which for the climatologists and the meteorologists remains a challenge. Knowledge of the snow cover is important, because through its inherent physical properties, snow cover significantly influences the evolution of weather on a daily basis as well as climate variability on a much longer time scale. Snow modifies the radiative balance and energy exchanges between the surface and the atmosphere, and thus indirectly acting on the atmospheric dynamics [10].

Satellites are well suited to the detection of snow cover because the high albedo of snow presents a good contrast with most other natural surfaces (except clouds) [11].

The purpose of snow mapping algorithms is to generate global snow-cover products from space satellites sensor data [8, 11]. The snow maps will augment the valuable record of global snow cover and will provide a global and improved source of information for the study of land surfaces.

The parallelization of the selected algorithms in this paper is made on pre-processed, rather than raw, sensor data in order to see easier the results of the parallel processing.

2.1 Snow mapping algorithms

The reviewed algorithms, use as base for the parallel realization, are proven to be reliable and are used for creation of daily snow maps in different space centers. They were selected to cover the two fundamental satellite-based snow mapping techniques, depending on the type of the used sensors: visible/infrared and microwave.

2.1.1 Visible/infrared snow detection

Detecting snow in visible/infrared part of the electromagnetic spectrum consists of satellite detection of Earth surface reflection of waves in that part of the spectrum (10^{12} – 10^{15} Hz). The data registered is processed to specify snow cover by detecting certain characteristics of snow. Such characteristic that is often used in snow mapping analyses is the Normalized Difference Snow Index (NDSI).

Daily snowmap cover product, released by Goddard Space Flight Center at National Aeronautics and Space Administration of USA, are based on an algorithm using NDSI [11]. The product contains everyday, weekly and monthly maps of Earth surface snow cover, produced by data from the Moderate Resolution Imaging Spectroradiometer (MODIS) instrument on board of EOS Terra и EOS Aqua satellite series. That serial algorithm was used as basis for parallelization efforts in this paper.

NDSI is index used to distinct snow cover from most of other Earth covers. The NDSI is a measure of the relative magnitude of the characteristic reflectance difference between the visible and short-wave IR reflectance of snow. The index is normalized (between -1 and 1) and is dependant on reflection from multiple spectral bands.

$$NDSI = \frac{RED - IR}{RED + IR} \quad (1)$$

Although this evaluation is independent for each pixel of the input data, the space resolution of the MODIS instrument is high.

2.1.2 Microwave snow detection

To detect snow in microwave part of the electromagnetic spectrum, Earth surface reflection of waves in the GHz range is captured by satellite sensors. Snow cover is identified by the scattering of high frequency microwaves from ice particles differing to the lower frequency measurements. The difference is represented by scattering indices.

Algorithm based on scattering indices is underlying for a daily snowmap product offered by the Center for Satellite Applications and Research (STAR) at National Oceanic and Atmospheric Administration of USA [13]. The data used are from the instrument Advanced Microwave Sounding Unit (AMSU-A) on board of NOAA KLMNN satellite series. That algorithm was used as basis for parallelization implementations in this paper.

Registered reflected microwaves are transferred to brightness temperatures of the surface. The brightness temperatures differ for different wave-lengths.

Cover detection is found on the fact that ice particles in snow reduce the high frequency brightness temperature of high frequency microwaves relatively to the lower frequency. The scattering indices Ω are calculated in the following way based on the brightness temperatures TB of different frequency sensors:

$$\Omega_{31.4} = \mathrm{TB}_1 - \mathrm{TB}_2 - 2.0 \quad (2)$$

$$\Omega_{89.0} = \mathrm{TB}_1 - \mathrm{TB}_{16} - 3.0 \quad (3)$$

2.2 Parallelization models

The two main methods of approach for parallelizing algorithms are considered: data parallelism and functional parallelism.

2.2.1 Data Parallelism

This kind of parallel model is based on the need to process a lot of data in the same way. Well known examples are found in the area of image processing. The common principle is that one program code treats different kinds of data on several independently processing nodes. The corresponding software architecture usually consists of one 'master' process, distributing the data and several 'worker' processes that do the actual work.

2.2.2 Functional Parallelism

This kind of parallel model is based on different functional blocks in the application. The idea is to split the application into separate processing units, that communicate with a fixed number other units in such a way that the output of one part serves as the input of another part. Each of these parts may be placed on a different processor.

2.3 Implementation and technologies

Both types of parallel models are considered in the paper. They are implemented using C++ programming language. Implementation was made for the two basic architectures in parallel computing: message passing and multithreading, as well as hybrid models that incorporate both message passing and multithreading.

For realization of the message passing models on distributed memory systems MPI (message passing interface) is used and the implementation is based on MPICH2 library for C++ that allow transferring messages between nodes and dividing the data to be processed between them.

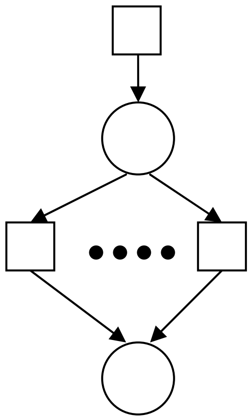


Figure 1. Communication flow using message passing

Realization of the multithreading model on a shared memory system was done using OpenMP API that allow defining different routines for each thread and automatic parallelization of loops.

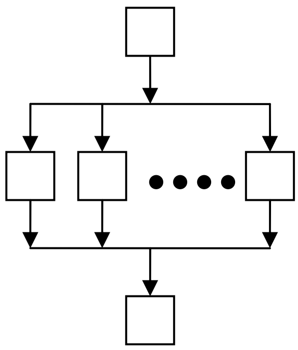


Figure 2. Processing with multithreading

Hybrid realization is built using both interfaces MPI and OpenMP: message passing to transfer input and output data to nodes and multithreading to distribute processing between threads in each node.

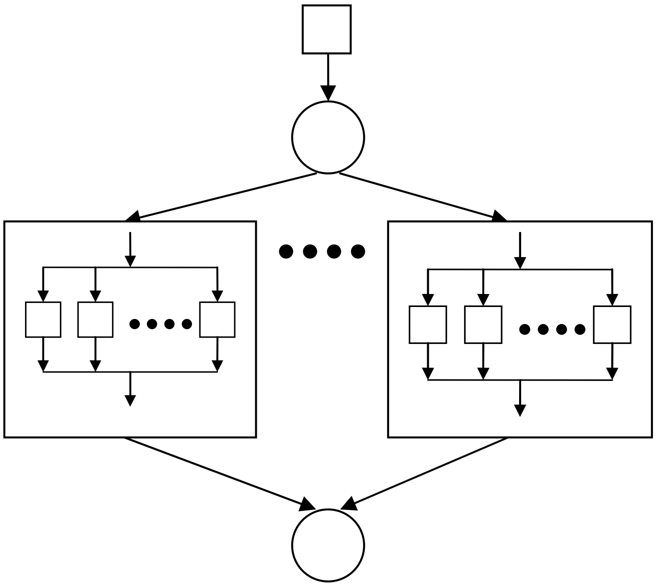


Figure 3. Hybrid architecture

Supplying with input data in HDF file format was made using HDF-EOS tools. The output data and graphic file manipulation was done with Magic++ API.

For each type of parallelism (data and functional) and each technology used (MPI message passing, OpenMP multithreading, hybrid) several different paradigms was implemented, such as „SIMD" (Single Instruction – Multiple Data), „Master-Slave", „Divide and conquer", 'Workpool".

3 Results and analysis

The experimental evaluation of the implementations of the different parallelization technologies and paradigms were made on compact heterogeneous cluster, but homogeneous subset of computing nodes was used in order to obtain clearer results (table 1). The execution time was monitored and mean execution time for each implementation was established. Each execution was run with the same size of input data.

High-performance computer platform	**Parameters**	
Heterogeneous multiprocessor cluster (situated at the Technical University of Sofia, Bulgaria)	10 compute nodes, 32 cores 8 AMD Opteron 64 Dual Core, 1.8 GHz, 2GB RAM 2 Dual Intel Xeon E5405 Quad Core, 2 GHz, 4GB RAM OS: Scientific Linux 5.3 64 bit, Middleware: MPICH2 1.1.1p2	
Parallel algorithmic platform	**Parallel model and programming interface**	
Visible/infrared snow cover detection	Data decomposition, MPI Data decomposition, OpenMP Data decomposition, MPI+OpenMP	Functional decomposition, MPI Functional decomposition, OpenMP Functional decomposition, MPI+OpenMP

High-performance computer platform	Parameters	
Microwave snow cover detection	Data decomposition, MPI Data decomposition, OpenMP Data decomposition, MPI+OpenMP	Functional decomposition, MPI Functional decomposition, OpenMP Functional decomposition, MPI+OpenMP

Table 1. Experimental platform

Best results among the tested implementations shows the parallelization based on data decomposition for both of the algorithms. The model with message passing using MPI shows some execution time improvement to consecutive implementation. The model of multithreading using OpenMP shows very strong execution time diminishing. The hybrid model shows no better results than pure MPI implementations.

Functional parallelism achieved worse results than data decomposition, because the algorithms cannot be divided to functions with equal computational weight and because of the division requires additional processing to combine the results of the subfunctions.

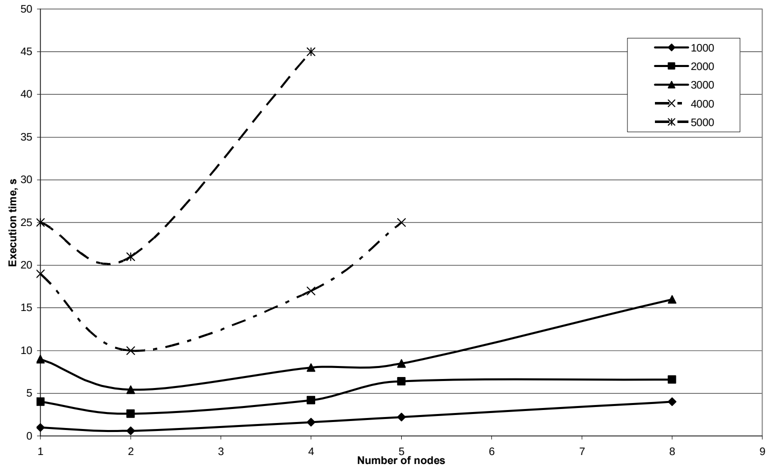


Figure 4. Execution times for data decomposition based SIMD parallelization using *message passing* (MPI) of *visible/infrared* snow cover detection

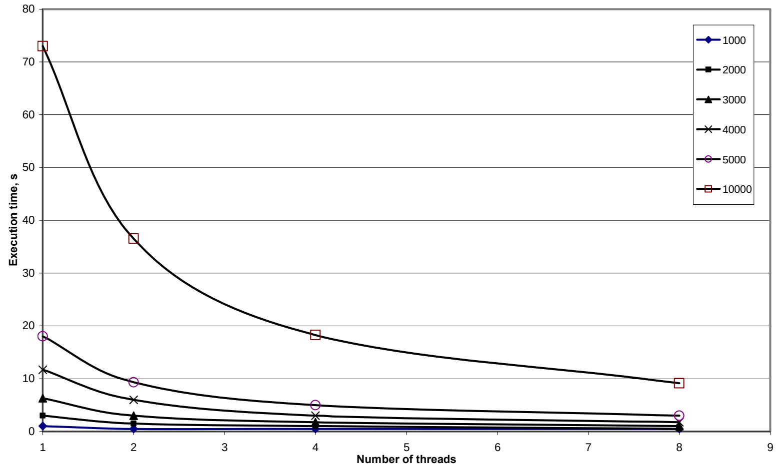


Figure 5. Execution times for data decomposition based SIMD parallelization using *multithreading* (OpenMP) of *visible/infrared* snow cover detection

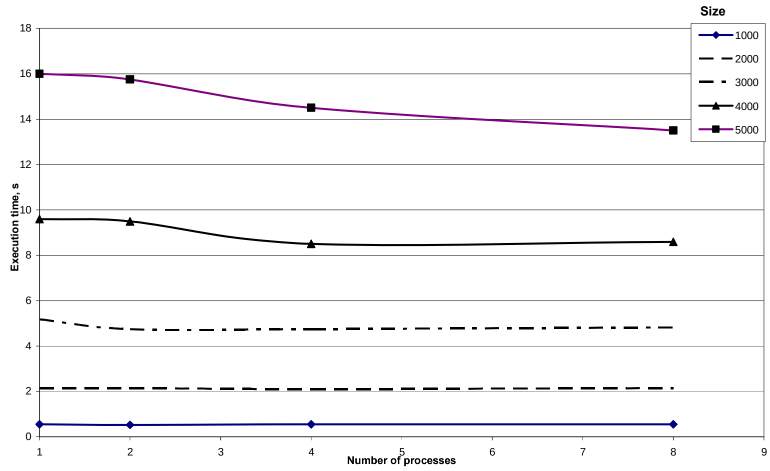


Figure 6. Execution times for data decomposition based SIMD parallelization using *message passing* (MPI) of *microwave* snow cover detection

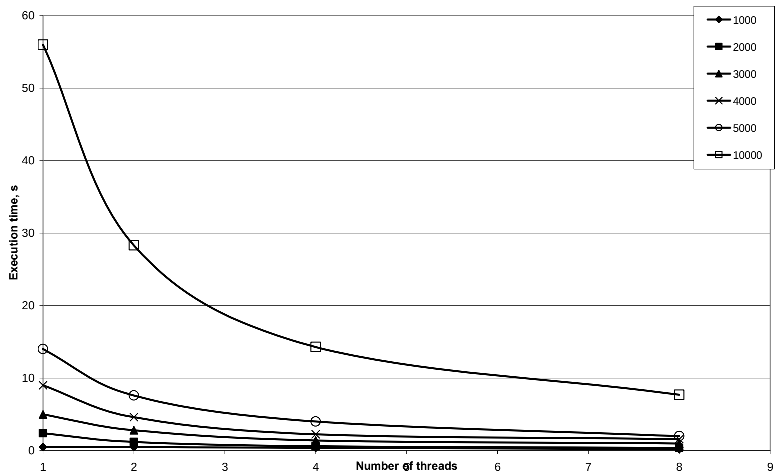


Figure 7. Execution times for data decomposition based SIMD parallelization using *multithreading* (OpenMP) of *microwave* snow cover detection

The speed-up achieved with message passing is low and is present only when processing on two nodes, due to high communication cost for the transfer of the divided input satellite data and transfer back of the output to the parent node. Furthermore the speed-up shows maximum for size 3000-4000, thus communication costs are too high for both low and high dimensional data.

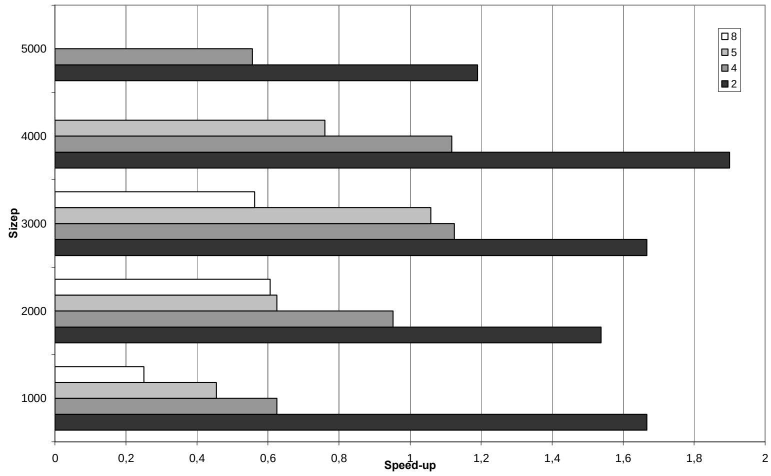


Figure 8. Speed-up for data decomposition based parallelization using *message passing* (MPI) of *visible/infrared* snow cover detection

The speed-up achieved with multithreading is remarkably high and is increasing when raising the input data size and increasing the number of processing threads.

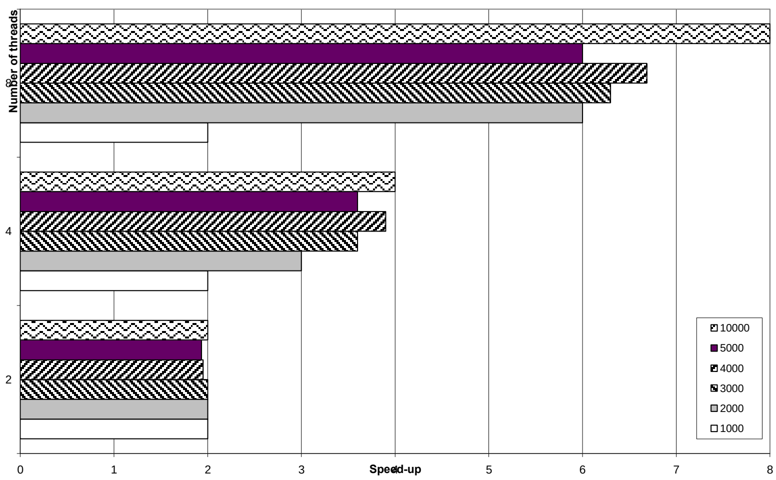


Figure 9. Speed-up for data decomposition based parallelization using *multithreading* (OpenMP) of *visible/infrared* snow cover detection

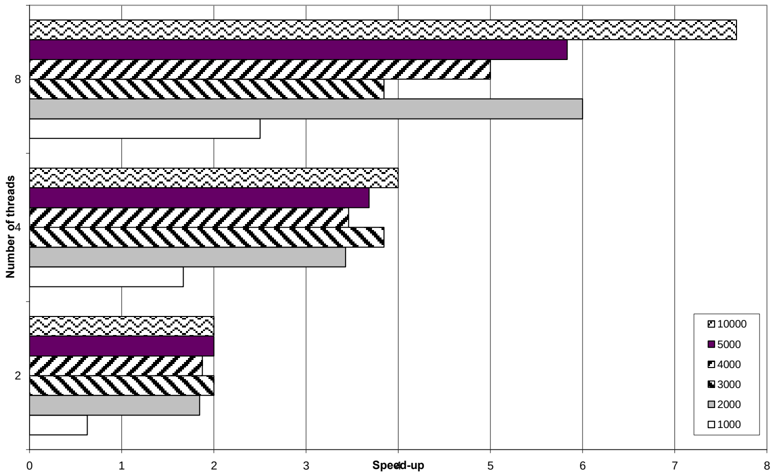


Figure 10. Speed-up for data decomposition based parallelization using *multithreading* (OpenMP) of *microwave* snow cover detection

The speed-up is close to linear and is high even for very large input data size; hence this method is promising for even higher-dimensional satellite sensors.

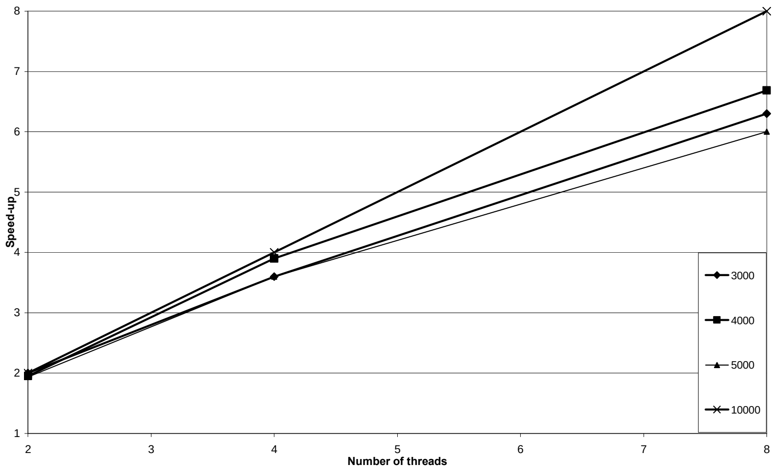


Figure 11. Speed-up for data decomposition based parallelization using *multithreading* (OpenMP) of *visible/infrared* snow cover detection

4 Conclusion

In this paper algorithms for remote sensing snow detection are reviewed and models for parallelization of data processing are presented. The used algorithms are based on visible/infrared snow detection and microwave snow detection.

The parallel models are implemented using the programming language C++ and the C++ frameworks MPI, for the message passing, and OpenMP, for the multithreading. The implementation was run on compact heterogeneous cluster and the execution times were monitored.

The achieved results show that best speed-up can be attained using multithreading and share memory computing model. The processing speedup in that case is almost maximal and close to linear for some of the parallel implementations and paradigms. The utilization of the processing units obtained is near 100%. The results are better when the input data is of greater size.

Other parallelization technologies implemented also achieve computational speedup but its lower and do not justify the additional processing power.

Faster processing of the satellite data can provide time for complementary computation, allowing better snow detection, like preprocessing, pixel clustering, segmentation, etc.

References

[1] J. Jensen, *Remote Sensing of the Environment – an Earth Resource Perspective*, Prentice Hall, 2007.

[2] T. Lillesand, R. Kieffer, J. Chipman, *Remote Sensing and Image Interpretation*, John Wiley & Sons, 2004.

[3] P. Clark, R. Michael, *Remote Sensing Tools for Exploration: Observing & Interpreting the Electromagenetic Spectrum*, Springer, 2010.

[4] A. Plaza and Chein-I Chang, *High Performance Computing in Remote Sensing*, Chapman&HallCRC, 2008.

[5] G. Li, Y. Ma, J. Wang, D. Liu, Preliminary Through-Out Research on Parallel-Based Remote Sensing Image Processing, *Proc. of Int. Conf. on Computational Science – ICCS 2006, Lecture Notes in Computer Science*, vol.3991, 2006, pp.880-883.

[6] Tongjun Qi, Jingyun Fang, Haojun Liao, Chengde Han, A parallel fusion algorithm of remote sensing images, *Proc. of 17th International Conference on Geoinformatics*, 12-14 Aug. 2009, pp.1-6.

[7] Liu, Zhaohua; Yang, Jingyu, Remote sensing image parallel processing system, *Proc. of the SPIE*, vol.7497, 2009, pp. 74970G-74970G-5.

[8] The National Operational Hydrologic Remote Sensing Center: NOAA's National Weather Service, http://www.nohrsc.nws.gov

[9] MODIS Land Surface Reflectance Science Computing Facility, http://modis-sr.ltdri.org/index.html

[10] F. C. Pivot, C. R. Duguay, R. D. Brown, B. Duchiron, C. Kergomard, *Remote Sensing of Snow Cover for Climate Monitoring in the Canadian Subarctic*, 59th EASTERN SNOW CONFERENCE, 2002.

[11] D. Hall, G. Riggs, V. Salomonson, *Algorithm Theoretical Basis Document (ATBD) for the MODIS Snow and Sea Ice-Mapping Algorithms*, http://modis.gsfc.nasa.gov/data/atbd/atbd_mod10.pdf, 2001.

[12] MODIS snow and sea ice global mapping project, http://modis-snow-ice.gsfc.nasa.gov/

[13] NESDIS Microwave Surface and Precipitation Products System, http://www.star.nesdis.noaa.gov/corp/scsb/mspps/main.html

PETAR MARINOV
Technical University of Sofia
Faculty of Computer Systems
Sofia, blvd. Kl. Ohdiski 8
BULGARIA
E-mail: petjofy@hotmail.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

3D Interior Design

Daniel-Sorin RADU, Vasile Nicuşor NECHIFOR
Coordinator: Prof. dr. Dana Simian,
Lect. Ralf Fabian

Abstract

The article presents an original application for interior design using Java 3D that helps to place furniture on a 3D house plan. This application with a user friendly interface was made for people who want to design their interior quickly, whether they are moving or they just want to redesign their existing home.

1. Introduction

Many home decorating projects can be done without much advance planning, like changing the paint color in living room or reupholstering a piece of furniture. But if you want to make a significant change, such as remodeling a kitchen or knocking down some walls, you really shouldn't start until you have a floor plan drawn up.

For decades, architects and engineers have been taking advantage of computer aided design (CAD) for drafting and planning beautiful homes. But for the non-professional, CAD programs often seem too technical and difficult to learn. So instead you might spend hours drawing a rudimentary sketch on graph paper, erasing what didn't look right, redrawing, and repeating, fortunately now it's easier to decorate a house using *3D Interior Design* application. This program makes it fun to plan your home's design and you can do it all without having to move a single piece of furniture. Remodeling a home requires a lot of planning. There tends to be a lot of money involved, so one wants to be sure to get things right. Also, people will probably have to live with their design decisions for a long time. Because of this, many people use interior design programs to get a better idea of how their room will look before making any lasting decisions. There are many interior design applications but most of them are difficult to use and very expensive(3D Spacer, Live interior 3D, Plan 3D). For this reason the fact that this application is free and open source is a great advantage.

A wonderful attribute of this design and style application is that furniture arrangements can be made easy on the screen. No much more moving around large pieces of furniture, only to change the mind and have to move it somewhere else. Using this software a furniture placement can be finalized on the 3D house plan before.

3D Interior Design is perfect for:

- People who are making decisions about the look of their future home.
- People who want to get a look at their interior before buying furniture and paints.
- Professional designers who prepare presentations for clients and advertising.
- Architects working on a home plan or layout of new building additions.

2. Application design

3D Interior Design is a three-dimensional modeling program for interior home design. Most programs of this kind are using a 2D window for manipulation of objects and the 3D window is only for viewing the result. The power and flexibility of our application are based on the direct manipulation of 3D objects in three-dimensional space. This application has a dual role main window that is used for viewing and manipulating used furniture models. Unique furniture pieces can be loaded into the scene and moved using the standard mouse.

In this section we describe some features of your program like 3D navigation options, importing 3D furniture objects, importing and manipulating 3D house objects.

2.1. Navigation

2.1.1 Aerial view

Two different way of viewing the home are available, the default selected mode set with *Navigation > Aerial view* menu and the other mode set with *Navigation > Virtual visit*. In both modes, you can use the mouse or keyboard arrows to change the current point of view or you can move or rotate the furniture if the *Navigation > Move Objects* option is enabled.

In the *Aerial view* mode the house can be explored using some controls:

-Rotate the scene: when the left mouse button is pressed, the scene shot and can be viewed from different viewing angles.

- Zooming the scene: when the middle mouse button is pressed, the scene is zoomed. The desired part of the scene can thus be viewed up close.

- Move the scene: if the right mouse button is pressed, the scene is moved. This is especially helpful when you have zoomed the scene.

2.1.2 Move Objects

If the *Move Objects* option is enabled, furniture models can be moved, rotated and deleted.

-Turning a piece of furniture: if the left mouse button is pressed, the furniture is turned on XY plan while moving the mouse in the horizontal direction.

-Delete a piece of furniture: if the middle mouse button is pressed, there will appear a small menu with "Delete" and the selected piece of furniture will be deleted.

-Moving a piece of furniture: while the right mouse button is pressed, the furniture is moved on XY plan until the furniture is in the desired location.

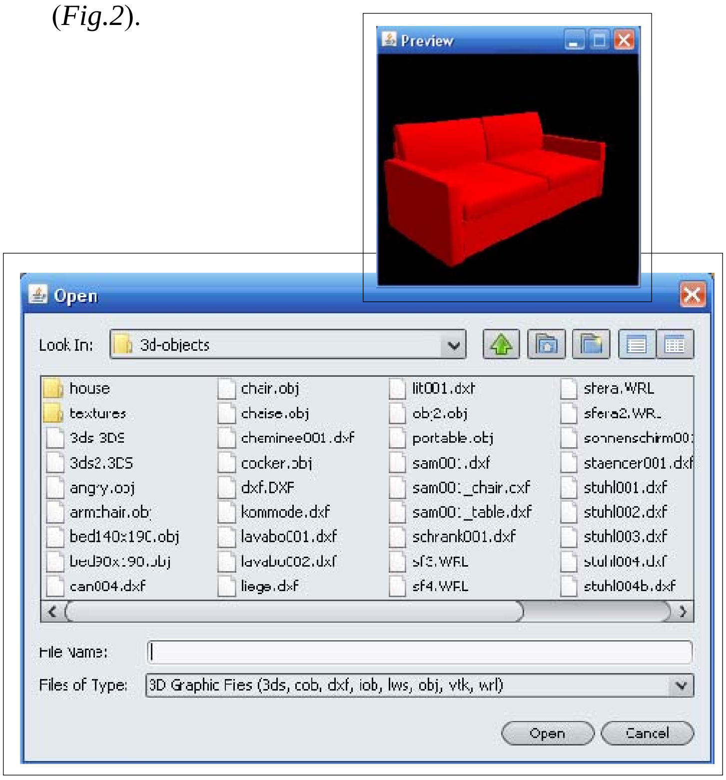
2.1.3 Virtual Visit

In the *Virtual Visit* mode the house can be explored and all the things are rendered at real size. The head angle can be tilt using the keyboard controls, to view various heights of the room.

- *Up arrow*: move forward
- *Down arrow*: move backward
- *Right arrow*: turn right
- *Left arrow*: turn left
- *Shift + right arrow* key: go sideways to the right
- *Shift + Left arrow* key: go sideways to the left
- *Page Up*: move to top
- *Page Down*: move to bottom
- *Home*: look up
- *End*: look down

2.2 Furniture

A piece of furniture can be loaded using the *Object >Load* menu. This option will open a file chooser dialog and a preview window (*Fig.1*) helping the user to load a desired object.
Another way to load furniture in the scene is the furniture tree catalog. By default the objects are grouped into categories called Bathroom, Bedroom, Kitchen and Living Room. This tree catalog contains a preview window of the furniture and specific icons for each item as can be seen in (*Fig.2*).

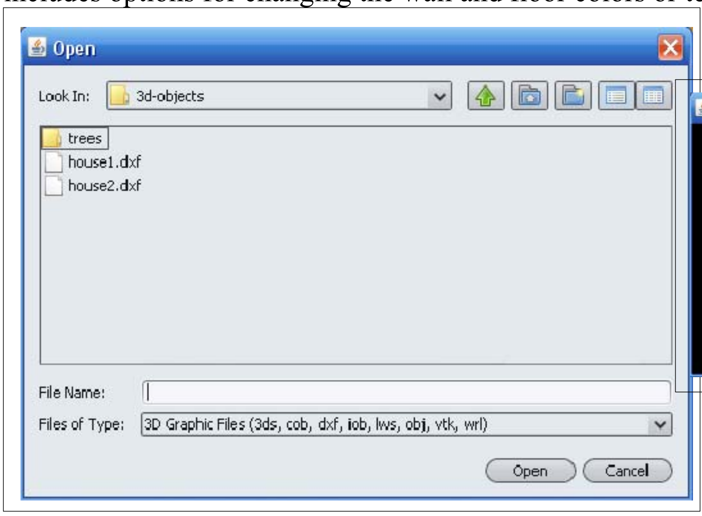


File chooser dialog Fig.1

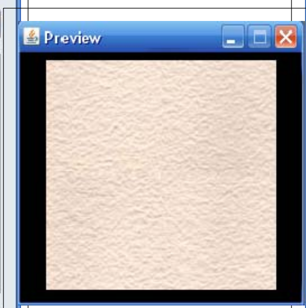
Furniture tree catalog Fig.2

2.3 The house menu

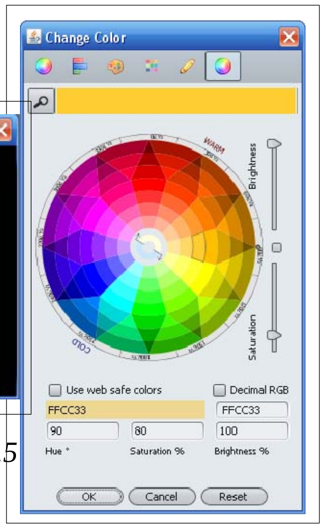
In *House* menu, there is an option *New House* which allows loading a new house 3D object (*Fig.3*). This option will delete all existing furniture and will reset walls and floors textures. This menu also includes options for changing the wall and floor colors or textures.



File chooser dialog Fig.3



Texture preview Fig.5



Color change Fig.6

2.4 Saving and Loading

Another important feature of this application is that a project can be saved or loaded at any time. This allows the user to continue a project or to save a desired house design configuration.

To save the current setup, select *File >Save* menu and there will appear a dialog in which the desired file name can be specified.

To load a saved setup select *File >Open* menu and there will appear a dialog in which the desired file can be selected.

2.5 Menus and their functions

The *File* menu:

- *Open*: open a saved device
- *Save*: saves the current facility
- *Reset View*: the view of the scene is reset
- *Exit*: exit the program

The *Navigation* menu:

- *Navigate Scene*: navigating the scene with the mouse
- *Move Objects: move the furniture with the mouse*
- *Walk Scene*: walk around the house

The *House* menu:

- *New House:* load a new house
- *Change Wall Color*: changing the wall color
- *Floor Color Change*: change the color of the ground
- *Apply Wall Texture*: put texture on the walls
- *Apply Floor Texture*: texture lay on the floor
- *Color Reset*: resets the colors of the walls and floor

The *Object* menu:

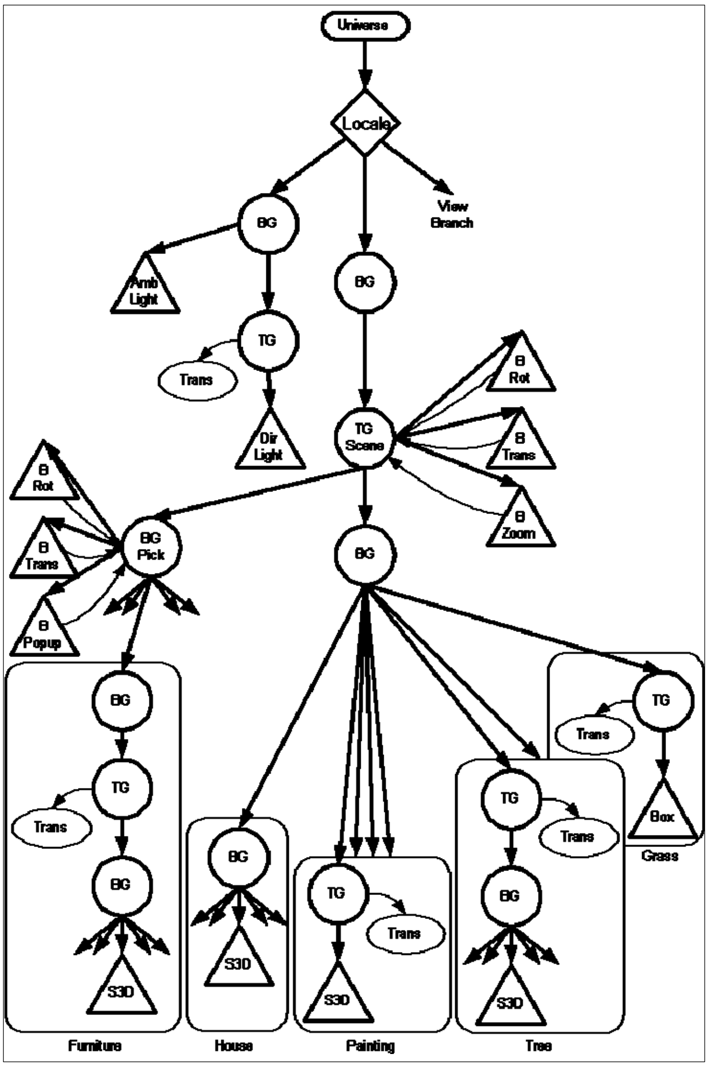
- *Load:* load a piece of furniture
- *Show Preview*: view the preview for furniture
- *Remove all objects*: delete all the furniture from the scene

3. Implementation

The implementation of this application was based on the Java 3D API. I chose this technology because the design of Java 3D is significantly different from popular 3D graphics APIs such as OpenGL and Direct3D, which are low-level procedural APIs that are closely tied to the design of 3D hardware. Main characteristics of Java 3D can be found in [1]. With Java 3D, all the graphics objects (also called geometry objects) must be set up in a scene graph, which is a hierarchical model containing all the information about the objects in the scene and how they will be rendered. Then Java 3D will render the scene graph. Java 3D can take advantage of any 3D acceleration that a graphics adapter provides.

3.1 Content Scene Graph

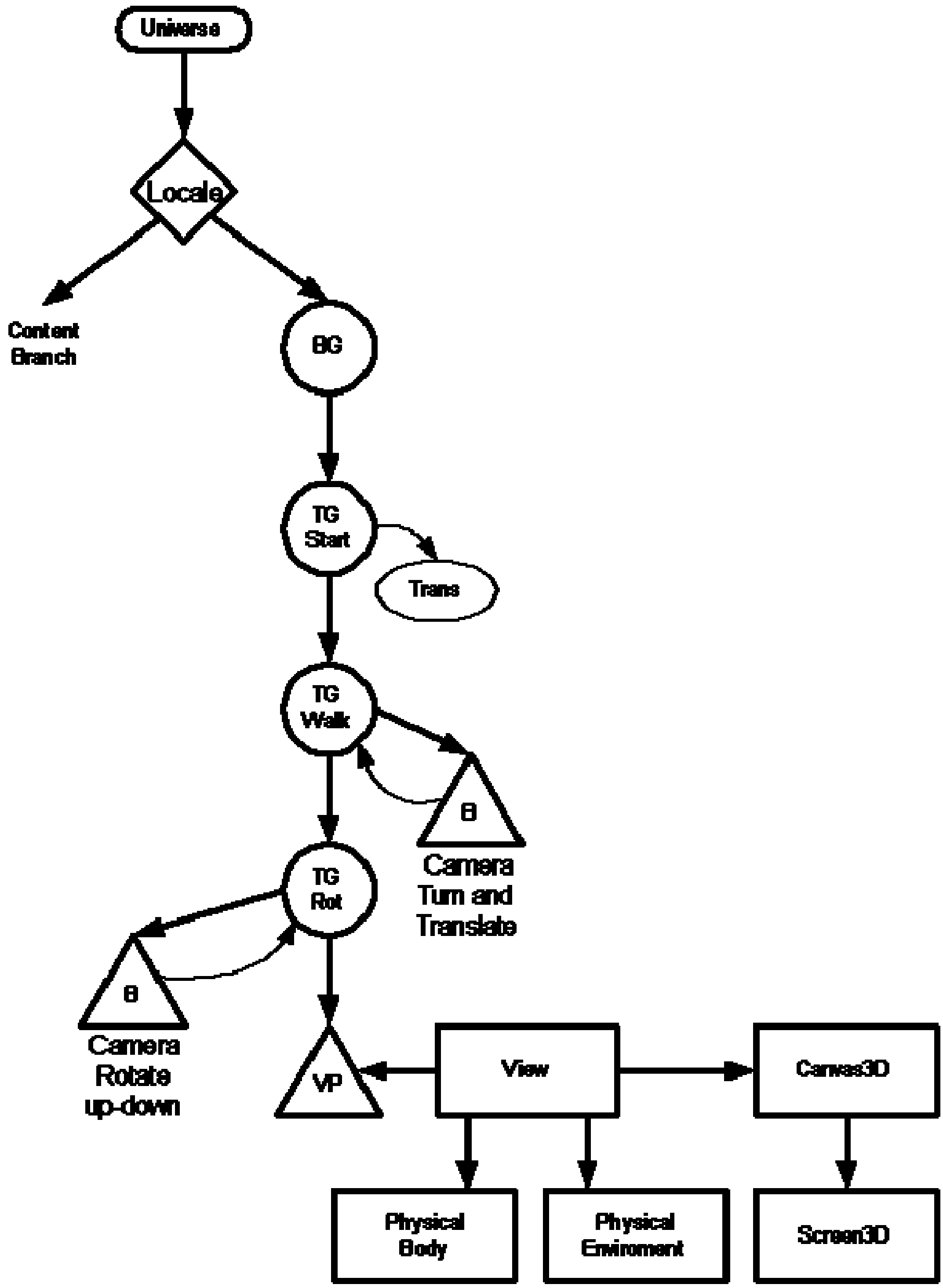
On the *Content Scene Graph*, the light and the 3D objects will be inserted. The scene *Transform group* is responsible for moving the scene. *Pick The Branch Group* has all the 3D objects among themselves, which can be selected by picking, move and rotate when each of the transform of the *Furniture Group* object is changed. To simplify all of the objects were not shown when more of the same species are present. Arrows indicate where more such objects are attached to. Similarly, the *Geometry and Appearance* objects were omitted. For better understanding, objects, which consist of several groups or shapes, are surrounded by a frame.



Content graph scene Fig.7

3.2 View Scene Graph

The View Branch is structured so that the camera can be moved through the house. The transform group, to which the view platform is attached, is only for the rotation up and down charge. Transforming the parent group makes the camera moves through space. In this way the camera moves up or down during the up or look down moves.



View scene graph Fig.8

3.3 Particular implementation

3.3.1 File format

To save the current setup, must have the information about position and rotation of the objects, textures and colors of the house, and the file of the house to be known. Then the values can be stored in a file. It uses one row per object. The data is stored as text into a file. The individual components of a row are separated with a rope point.

Position

The position of each object is found relatively easily. The vector of displacement can be read from the corresponding transform group. Its individual components are separated by a semicolon and stored in the file.

Rotation

The problem is to find the angle of rotation around the Z axis. This angle can not be easily read from the transform group. There is only the possibility of a matrix with the components of the rotation to get. In a rotation around the Z axis, the matrix contains the following values where the angle “a” is the desired one:

cos (a)	-sin (a)	0
sin (a)	cos (a)	0
0	0	1

With the corresponding inverse functions, the angle can be calculated. With these functions, we get only one of two solutions. Therefore, we need to know in which quadrant you find the desired solution. This can be seen in the sin and cos values.

Now the desired angle can be determined. The following code calculates the angle where the variable contains the value of a the result in radians.

```
Matrix3f Matrix3f m = new ();
t3d.get (m);
double a = 0.0;
if (((m.m00> = 0) & & (m.m10> = 0)) | | / / 1 or 2 quadrants
((M.m00 <0) & & (m.m10> = 0)))
 a = Math.acos (m.m00);
else if (((m.m00 <0) & & (m.m10 <0)) | | / / 3rd or 4th quadrant
       ((M.m00> = 0) & & (m.m10 <0)))
       a = 2 * Math.PI-Math.acos (m.m00);
```

3.3.2 Texture coordinates

The texture coordinates are calculated by using the Java class First, it gave unusable results. To obtain a useful result, the S and T levels are correct. Unfortunately, the documentation is not clear how accurate the calculated texture coordinates with these levels. Therefore, the levels were found by trying. For the walls, the values are set as follows:

```
TexCoordGeneration tcg TexCoordGeneration = new ();
tcg.setPlaneS(new Vector4f(1.0f, 1.0f, 0.0f, 0.0f
tcg.setPlaneT(new Vector4f(0.0f, 0.0f, 1.0f, 0.0f));
For the base, the values are set as follows:
TexCoordGeneration tcg = new TexCoordGeneration
tcg.setPlaneS(new Vector4f(1.0f, 0.0f, 0.0f, 0.0f));
tcg.setPlaneT(new Vector4f(0.0f, 1.0f, 0.0f, 0.0f));
```

4. Future developments

In this section we specified some future developments, features that would improve our application.

- implementation of a window for drawing a floor plan using a picture as background;
- export to .jpg, .gif and .jpeg formats;
- measuring tools: linear and angular rulers, ability to measure length, radius, areas;
- gridlines that allow accurate snapping;
- options to insert windows, doors, embrasures, niches, switches and decorations;
- advanced lighting effects;
-export the house to .obj and reuse it in 3D software like Blender to improve rendering quality.

5. Conclusion

The article presents an original application for interior design using Java 3D. Using *3D Interior Design* allows users to digitally add new spaces and elements of interior design. Our application allows owners of intelligent interior design skills to create their own plans, eliminating the need for a professional consultation.

Whether the user is a professional interior designer or just someone looking to remodel a home, *3D Interior Design* can help considerably with the design process. Our application is a free top home design software because of its flexibility.

Using *3D Interior Design* is an excellent way to layout a residence. It will assist the user to finalize all strategies prior to start. This will help for saving a lot of time, cash, and effort on altering elements soon after start. Property interior design computer software is actually the wave of the design long term.

References

[1] Andrew Davison . *Killer Game Programming in Java*. O'Reilly Media, 2005.

[2] Kevin Sowizral, Kevin Rushforth, Henry Sowizral. *The Java 3D API Specification*., Addison-Wesley Professional, 2000.

[3] Daniel Selman. *Java 3D Programming*, Manning Publications Co, 2004 .

[4] Java3D homepage: http://java.sun.com/products/java-media/3D/index.html

[5]Java3DAPISpecification: http://java.sun.com/products/javamedia/3D/forDevelopers/j3dguide/j3dTOC.doc.html

[6] Java3D forum: http://forum.java.sun.com/list/discuss.sun.java3d

[7] J3D.ORG: http://www.j3d.org/

Daniel-Sorin RADU
"Lucian Blaga" University of Sibiu
Department of Informatics
No. 5-7, Dr. Ion Ratiu st., 550012 Sibiu, Romania
E-mail: radu_s_o_r_i_n@yahoo.com

Vasile Nicuşor NECHIFOR
"Lucian Blaga" University of Sibiu
Department of Informatics
No. 5-7, Dr. Ion Ratiu st., 550012 Sibiu, Romania
E-mail: vasilenicusor@yahoo.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Methodology for creating automated unit of organisation scientific conferences

Sivko Lidiya
Teacher Coordinator: Oleksandr Dorokhov

Abstract

At the modern stage of transition from post-industrial community to information society in conditions of permanent radical shifts in scientific and technological sphere substantially increase demands on the level of information of society. This involves a complex of fundamental problems. One of the most important of such problems is a scientific issue. The part of solving this fundamental problem is an urgent need for automation unit for scientific conferences. This necessity is caused by the fact that the main way of providing quality and reliable information, which contains the results of scientific activity, is the science conferencing. One of the key elements to ensure the automation of this module is the process of creating websites of scientific conferences. Resources of this type has already become one of the ways (with great potential) to grab the attention of the scientific community to organized conferences. At the moment, the problem which concerns to formalization of methodology and determine the direction of the module support for the conference is still very actual.

1 Introduction

A scientific conference is one of the main forms of implementation the function of scientific knowledge in society, as well as presentation and storage results of researches. In general, a scientific conference is organized meeting of people to exchange information with the same theme. There are three types of scientific conferences: the scientific-theoretical conference, scientific-practical conference, scientific-technical conference. In the process of organizing and holding such events for two major subjects of scientific conferences − the organizer and the participant − it is necessary to perform some routine operations. If the conference has regular character (for example annual etc.) execution of the similar actions will be unacceptable. Thus, all preconditions for process of automation are formed. In the presence of a variety of hardware and software implementation means of this process is inevitable. Automation module for scientific conferences focused on the following tasks:

Formation of the fund of scientific works of certain subjects.

Presentation conditions of the conference.

Providing information of the preceding paragraphs in the form of Internet resource.

The emergence of many scripts and software and hardware tools for creating websites of scientific conferences resulted in the need for the main stages of development.

2 Stages of development of the science conference site

2.1 Definition of milestones of the science conference site

When you select a stage in the development of websites for scientific conferences you should be aware that the image of each site is unique. Proceeding from this point it is necessary to determine the stage of development so that a formal process will be universal. You should also determine the appropriate level of detail, i. e. to allocate the minimum possible number of interrelated stages, sufficient to cover the entire process. Determining the sequence execution stages also plays an important role. We can distinguish six main stages of the process of creating websites for scientific conferences (Fig. 1).

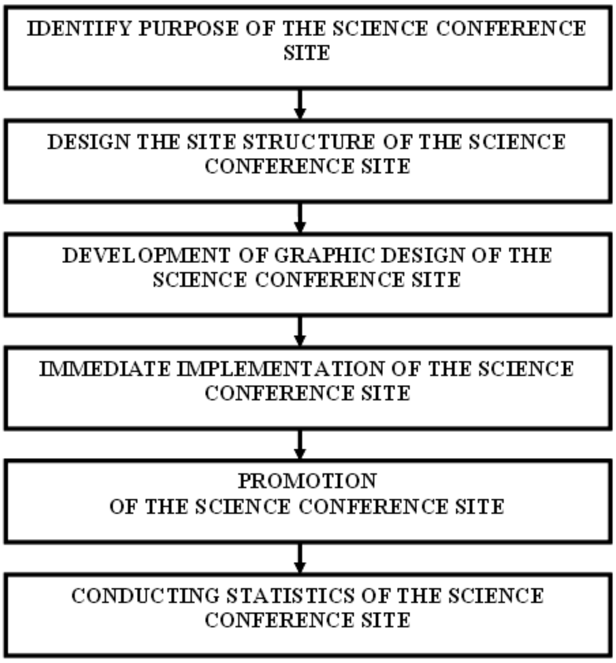


Fig. 1. The main stages of development science conference websites

2.2 Identify purpose of the science conference site

As the site is being developed for the upcoming conference, it should contain more information for users interested in upcoming events and also it should allow organizers promptly make this information available without performing routine tasks which require a lot of time and human resources. At this stage, we have to identify goals and outline the tasks which will be performed while creating a site. For example, such goal may be to attract participants to address the conference, or interested individuals who just want to hear new reports.

2.3 Design the site structure of science conference

The site structure should be determined at the early stages of the project. And in the future, web-masters should strictly stick to it. During the implementation of this phase the main sections of the future website [1] and one of types it organization are distinguished. In general, we can distinguish the following types of structures: linear, linear with the alternatives, linear with branches, tree and grid. Each of the sections of site has to contain pages filled with high quality content. The site structure

should be designed according to the further promotion of the site and according to it purpose. During the design of site for scientific conferences we have to identify main keywords according to which users will find needed pages and include those words on our site. Given the multiuser mode of created software product we can divide three main modules:
Module designed for conference organizers. It provides the ability to administrator of the site (Fig. 2). To work with this module it is necessary to undergo the procedure of authentication using a personal username and password.

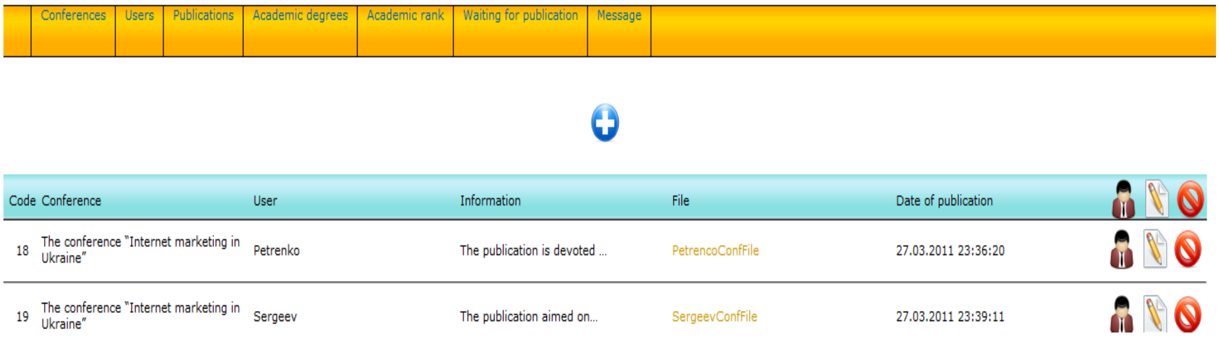


Fig. 2. Module designed for conference organizers

Module designed for participants of the conference. As a part of this module, for each of participants of the conference, for which the website is creating, there has to be a personal space − personal account − in the form of a single page (Fig. 3). To obtain such personal account it is necessary to register at the site. When the participant enters his own page, he will be needed to pass user authentication using a personal username and password.

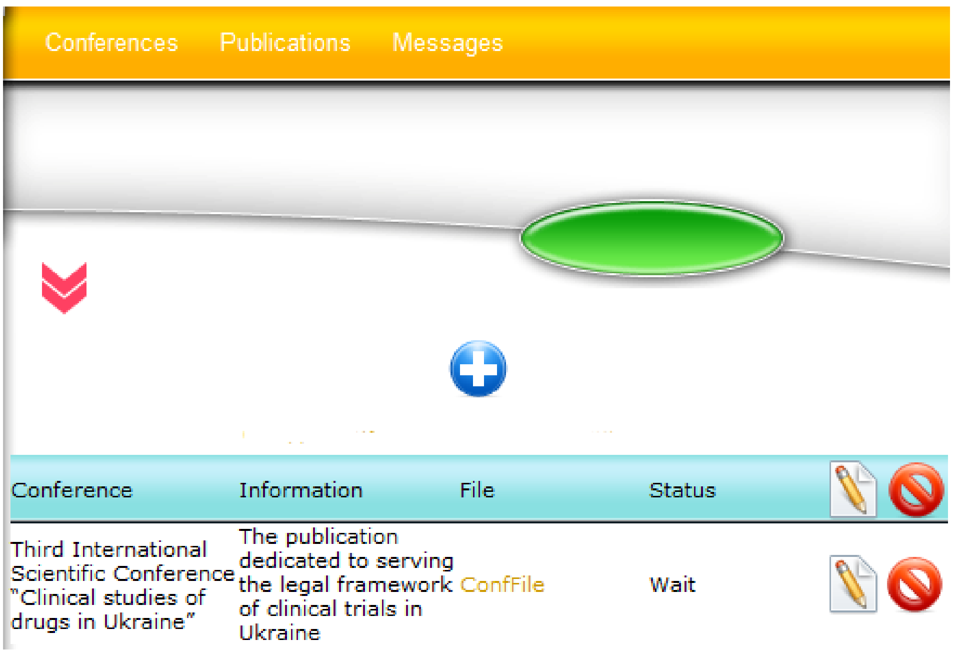


Fig. 3. Module designed for participants of the conference

Module designed for general users (Fig. 4). The work with this module does not require passing the procedure of authentication.

Fig. 4. Module designed for general users

This approach is actively used in practice for creating the website of annual scientific conference.

2.4 Development of graphic design of the science conference site

The design of the conference site must be made in the classic style and it have to focus on the structure of the site. It is unacceptable to overload the site images and advertisements [2].
To post photographs from the conference the designer has to create a separate page. It is recommended to place a map with a distinguished venue of the conference or to lay the optimal route from a certain point (if site is created for international scientific conference).
The use of animation is allowed only to attract visitors to the site content, providing more visual presentation of information. For example, technological process can be presented using the animation or you can animate the corporate logo. Currently, the animation is being developed using technologies Macromedia Flash, which significantly reduces the download speed.

2.5 Immediate implementation of the science conference site

At this stage, three main means of development can be identified. Selecting each of the funds has individual character.

HTML-editors. At the dawn of the Internet, all sites consisted of static html-pages [3]. To change the information on this site it was necessary to fix code of corresponding html-pages. The first attempt to automate this process was the creation of html-editors. With such editor, the user, which does not know html, can create html-pages and make changes to existing ones.

Dynamic sites. With developing of server languages CGI-programming (in particular, Perl), approaches to managing site content are improved too. Perl creates html-page, when the user is referring to it. Thus, it became possible to store the information on this website in the databases (which at first is a text file). When you access the page some information is extracted from the database and new html-page is generated. Also there are special administrative pages with restricted access to change the information in a database. Such sites, which contain dynamically changed information, are called dynamic.

CMS (Content Management System). Creating of dynamic sites − time-consuming process, so at this time in the Internet are very popular so-called CMS (CMS − Content Management System). They allow you to create sophisticated dynamic websites quickly and inexpensively.

2.6 Promotion of the science conference site

The promotion of web site of science conference − a process that involves placing a site in Internet with further attraction the attention of visitors. In implementing this stage it is necessary to develop proper indexing in search engines, website directories (using the special pay or free services). The active advertising at the forums of sites with similar subjects is used at the stage of promotion. Correct completion of META tags is also an important point. Because search engines take a subject from this tags to index pages. META tags are recommended to fill with maximum number of words associated with the subject of designed site.

2.7 Conducting statistics of the science conference site

The implementation of process estimating statistical data usually contains accounting the number of visitors for the current day and the number of viewed pages. It is important to estimate the ratio between the number of viewed pages and visitors, not only attendance. A large number of visitors and ratio greater than one point indicate a good promotion of the site, which is not interesting for the visitor. For highly specialized professional resource such like the site of a scientific conference valid value ratio of the number of viewed page to the number of visitors is three or four points. The necessity of full statistical estimating is caused by complexity of the statistical characteristics, the main ones are:

Depth of browsing the site. This statistic shows the characteristic distribution of visitors by the number of viewed pages. Typically, about 50% of visitors coming from search engines and directories, see the first page and realize that they made a mistake. This is the inevitable specificity of the search engines, and 50% a good result. But if the first page leave 80-90% of the visitors, it is said that the site or badly optimized (a non-target searches), or poorly executed.

The path through the site. This option displays the sequence of viewed pages. It allows you to track the interests of each individual user.

Keywords. The mapping of words that visitors type in search engines and using which they come to the site. Mentioned characteristics allow us to estimate the degree of optimization.

Directories from which visitors come to the site. Tracking the directory in which visitors go to a specific site (such as Yandex).

4 Conclusions

Over time, the benefits of using an automated module for the organization and holding of scientific conferences have become an obvious fact. The main stages of the process of creating websites of scientific conferences are clearly delineated. However, selected stages of development have been obtained by the synthesis of the general provisions of a commercial websites. But the direction of the sites of scientific conferences has more specific character.

The complexity of the object of study, the presence of multiple actors and the need for non-standard approach to solving problems make the process of research very complicated. Besides, the permanent addition of functionality such resources and continuous development of design technologies makes a barrier on the way of definition of universal stages of creating an automated module of scientific conferences. All these factors have led to the urgent need for further study of this topic.

References

[1] В. Н. Печников. *Создание web-страниц и web-сайтов,* Триумф, Киев, 2006.

[2] B. Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design,* Morgan Kaufmann, Burlington, 2007.

[3] И. М. Авдеева. *Инновационные коммуникационные технологии в работе конференций,* Профессионал, Санкт-Петербург, 2007.

SIVKO LIDIYA
Kharkiv National University of Ecomomics
Faculty of Economic Informatics
61001, K
harkiv, pr. Lenina, 9a
UKRAINE
E-mail: sneg_snegok@i.ua

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Cost effective funding method (CEFM software)

Alexandra Elena Stan, Claudia Elena Pamfiloiu
Associate professor Ph.D. Marian Cristescu
Student Ph.D. Laurenţiu Ciovică

Abstract

In a market economy and inevitably in a global financial crisis enterprise resources are often limited and insufficient to cover all needs or to implement investment projects necessary to continue their work. The funding process presents a particular interest, considering the necessity of obtaining them with lower costs. The options for collecting enterprise resources are characterized by diversity and complexity. This paper explores the entities' potential in raising capital for investment and presents a case study on the most likely financing options with B Test. The proposed new program comes to help managers in choosing the best solutions in terms of costs. The friendly interface suggests that with the variables included in the program the user can view the most profitable form of medium and long term investment: loan or lease.

1 Introduction

The enterprise financing decision is assumed, for the most part, by management. Shareholders pursue a payment of the initially advanced capital at a superior return rate than that of the investment opportunities offered by the financial market at a given moment. In other words they want an internal return rate superior to the capital fructification rate offered by the financial market. Thus, shareholders will get an increase of their final wealth, higher than that offered by another investment alternative at a given time. In essence, the company management acting in the interest of shareholders is forced to pursue the reduction of the cost of borrowed capital, aimed at increasing the company's financial return rate.
Thus we propose to analyze the medium-term financing decision of the company by using bank loans repaid through constant annuities and financial leasing, in other words develop a model on which to base the choice for an external financing source from the most likely option. Since the B test is only a theoretical concept at present, we wish through the proposed model to bring the concept closer to the business environment by creating a computer program that can determine the optimum financing method in the shortest time possible. We believe that the presentation of a certain example is essential in order to understand the computer application.
For exemplifying purposes we chose a company that wants to buy a bus whose purchase price and delivery is of 120.000 RON (27 907 Euro). For this, it addressed a leasing company who offered:
The financial offer contains the following contract terms:

- residual value: 24 000 Euro
- management fee of 4% of the good's value,

- 20% of the good's value paid in advance
- 12% interest for the leasing company,
- location duration of 24 months,
- full Insurance 4% of the purchase price of the property ,

This company may also contract a loan where the loan represents 80% of the property, value the remaining 20% being own contribution. The user must guarantee with a good whose value is at least twice higher than the loan. The user will get CASCO insurance (4% of property value per year). The normal operation time of the property is five years and the depreciation will be linear. In addition the user must insure the property offered as guarantee at (Property insurance) 0.14% annual percentage of the purchase price of the property. The management fee charged by the bank is 3.9% of the loan and feasibility study expenses are 500 euro. The interest rate is fixed for d' = 9% per year. For analysis the company resorts to the B test.

2 Sources of financing an investment - B Test applicable models

Loans are known and regulated by law since the beginning of economic and social life.
Nowadays, the need of funding is solved by the entity, in most part, by resorting to bank loans and financial leases.

2.1 Credit

Loan definition. *A loan is an arrangement in which a lender gives money or property to a borrower and the borrower agrees to return the property or repay the money, usually along with interest, at some future point(s) in time. Usually, there is a predetermined time for repaying a loan, and generally the lender has to bear the risk that the borrower may not repay a loan (though modern capital markets have developed many ways of managing this risk).*[1]
Medium-term credit. *It is a loan whose repayment is made in between 1 and 5 years. It is granted for import-export activities for investment activities. Reimbursement usually occurs in monthly installments. These installments are calculated along with interest.*

2.2 Leasing

For entities like definitions, theorems, propositions, examples, lemmas, exercises, remarks, corollaries, use Another financing alternative is financial leasing. **What does leasing mean?** *Leasing refers to hiring equipment, such as a car or a piece of machinery, to avoid the capital cost involved in owning it. In some companies it is advantageous to use capital for other purposes and to lease some equipment, paying for the hire out of the income. The equipment is then an asset of the leasing company rather than the lesser. Sometimes a case can be made for leasing rather than purchasing, on the grounds that some equipment quickly becomes obsolete*.[2]
Leasing requires safe guarantees. Financial leases are considered capital leases, capitalized on the balance sheet, meeting the following conditions: [3]

- Risks and benefits of ownership pass to the user when signing the lease
- Upon expiration of the lease ownership of the asset is transferred to the user
- Period of use of the property in the leasing system covers at least 75% of the normal use of the property

For lease financing to be evaluated and compared with another source of long term financing, lease accounting transparency must be ensured, that correctly reflects it in accounting.

3 B TEST [4]

B Test is the ratio between the cost of capital currently under lease solution (Sl) and the cost of currently capital under loan repaid through annuity solution consistent (Sc) with the following details:

- if B < 1 the lease financing will be preferred.
- if B > 1 bank loan financing will be preferred;
- if B = 1 the choice of financing source is indifferent;

$$\text{Sl} = \sum_{i=1}^{m} \frac{Al_i + \text{int}\,erest_i \cdot (1-\alpha) - Am_i \cdot \alpha}{(1+k)^i} + \frac{VR_m}{(1+k)^m} - \sum_{i=m+1}^{n} \frac{Am_i \cdot \alpha}{(1+k)^i}; \tag{1}$$

$$\text{Sc} = \sum_{i=1}^{c} \frac{Ac_i + I_0 \cdot k_a \cdot \left(1 - \frac{i-1}{c}\right) \cdot (1-\alpha) - Am_i \cdot \alpha}{(1+k)^i} - \sum_{i=c+1}^{n} \frac{Am_i \cdot \alpha}{(1+k)^i}; \tag{2}$$

I0 = the amount of the capital investment (asset aquisition cost);
Ali = allowance for depreciation at a "i" period level;
Aci = direct depreciations of loan credit during "i" period (I0/c) level;
m = period term period;
n = period of the equipment's life;
c = period of the bank loan;
ka = nominal interest rate on the given bank loan;
k = cost of borrowed capital (company) (k = r (1-α));
Ami = annual depreciation of the asset, according to the "i" period;
$I_0 \cdot k_a \cdot \left(1 - \frac{i-1}{c}\right)$ = actual interest paid by the company in the "i" period ;
Interest I = leasing interest at the "i" period level;
VRm = residual value of the asset at the end of the lease;
α = rate of income tax on coefficient form.
Note:
Ali + interest.i = debt service on "i" period level; (leasing)
Aci + $I_0 \cdot k_a \cdot \left(1 - \frac{i-1}{c}\right)$ = the debt's service at the "i" period level;

3.1 B Test applied to the company:

$$S_l = \frac{\text{paid.in.advance} + \text{residual.value}(1-\alpha)}{1} + \sum_{i=1}^{n} \frac{A_{li} + (\text{interest.}i + insurance.i)(1-\alpha) - A_{mi} * \alpha}{(1+k)^i} + \frac{VR_m}{(1+k)^m} - \sum_{i=m+1}^{n} \frac{A_{mi} * \alpha}{(1+k)^i} \tag{3}$$

$$S_c = \frac{paid.in.advance + (management.fee + \text{feasibility.study.expenses})(1-\alpha)}{1} + \tag{4}$$

$$+ \sum_{i=1}^{c} \frac{A_{ci} + \left[I_0 + K_a * (1 - \frac{i-1}{c}) + insurance.casco + insurance.property\right](1-\alpha) - A_{mi} * \alpha}{(1+k)^i} - \sum_{i=c+1}^{n} \frac{A_{mi} * \alpha}{(1+k)^i}$$

Financial leasing: Sl = 117.771,11
Loan with constant annuities: Sc = 114,775,99

$$B=\frac{S_l}{Sc_1}=\frac{117.771{,}11}{114.775{,}99}=1{,}0260>1 \Rightarrow B=>1 \quad (5)$$

In this case it's advantageous to contract a bank loan repaid through constant annuities. You must take into account the legislative facilities which can make (more) profitable leasing rates (for example, if the property is acquired from import). Where there is financial trustworthiness it's recommended the bank loan. The leasing company finances its property by taking out a bank loan. The leasing company has sufficient financial resources it can fund from its own property. In this case, the interest is the same as from banks or less sometimes to the most payable product.
To sum up, the choice of funding source must be such as to choose the financing structure that minimizes the total cost of capital.

4 Application Design

In order to materialize these results and to offer a final product that users can use, we decided to create a small application, a web-based application which can be accessed from anywhere through either the browser from our desktops, laptops or phones / Smartphone's.

The application design is very simple because we want to provide a clean, easy to use interface through which the users can easily make their calculations and find their proper funding method depending on their incomes, period and other direct on indirect factors.

The application advantages are:

- very simple to use and operate;
- fast calculation;
- friendly interface;
- accessible from everywhere, due the fact that is made on a web platform;
- performs precisely the same operations each time they are run, thereby eliminating human error;
- minimise the costs of any entity that will use it;
- help managers in choosing the best solutions in terms of costs;
- this application, program, develop / offer the best investment solutions on the final activity of the organisation;

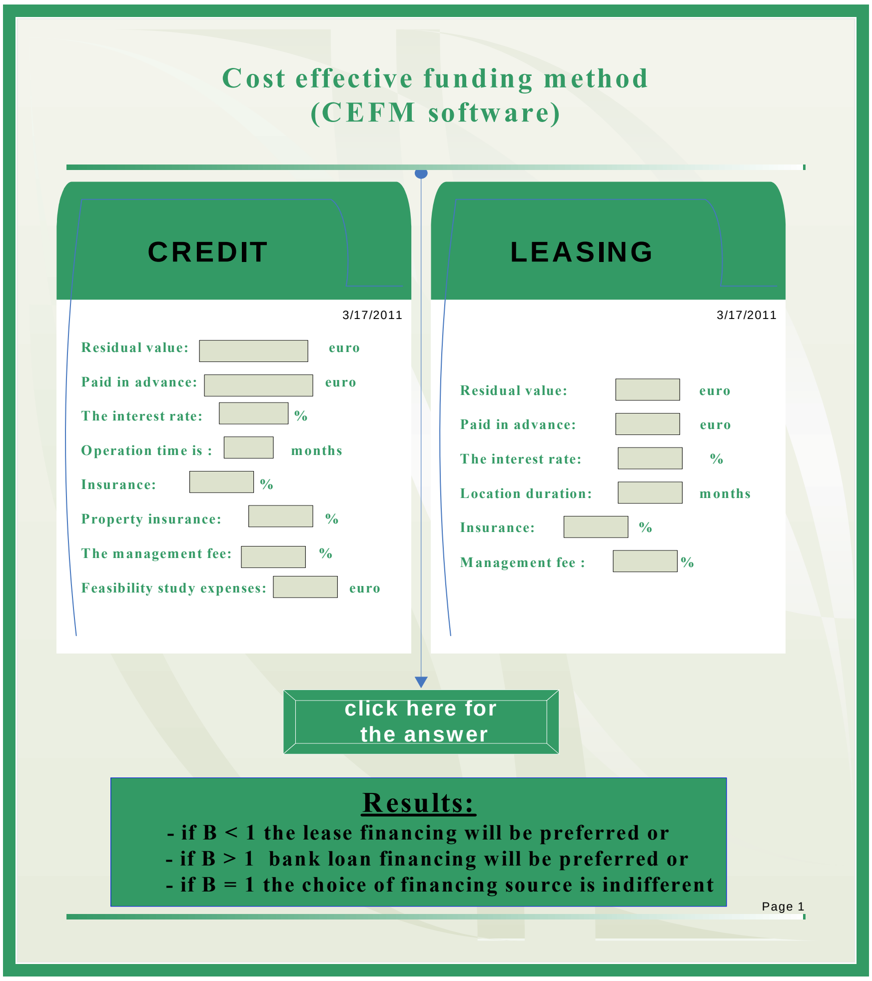


Figure 1 - Application UI Concept

5 Conclusion

B Test does not imply the selection of the right/perfect funding source; it actually implies the selection of the most cost effective funding option.
Any other variable introduced in the model must be carefully analysed and quantified to on the period on which it is available, then introduced in the program/software. In this way the accuracy of the chosen option as a funding source can be checked, because it is supposed to be a future expense.

Excepting the methods of choosing the funding source, B Test also offers an alternative of substantiating the funding decision. If from the current cash flow value (for both lease and loan), due to the existing relationship between those, the absolute values can be removed, the model will operate with coefficients only that will simplify the funding decision analysis.

We consider that nowadays, an efficient time management is crucial/very important. CEFM software offers the fastest possible solution for substantiating the investment decision for a company. In this way, the risk of choosing the wrong funding method does no longer exist, mainly due to the fact that the test is based on the opportunity cost. The test precisely shows the most suitable result and its purpose is to minimise the costs of any entity that will use it.

Moreover, the software offers a friendly user interface. The manager fills in the required information; he obtains the result and can make the right decision.

Our main objective regarding this application is to offer companies support in taking the most suitable and cost effective funding decision. The software is practical and useful for any entity when it confronts with making high risk decisions

References

[1] Able to www.investorwords.com
[2] *Oxford Dictionary of Finance and Banking*, 2005
[3] Art.2, litera e, din O.G. nr. 51/1997 privind operatiunile de leasing si societatile de leasing, republicata in M. OF. Nr.9 din ianuarie 2000.
[4] Vasile Bratian*, Modelarea deciziei de finanţare a firmei pe termen lung – Testul B*, Revista Finanţe Publice şi Contabilitate, editată de Ministerul Finanţelor Publice, Nr 11-12 /2003.
[5] Ilie Moga, *Moneda si credit*, Editura Universitatii"Lucian Blaga", Sibiu, 2008

ALEXANDRA ELENA STAN
The Faculty of Economic Sciences
Informatics in Economy
CALEA DUMBRAVII nr.17
ROMANIA
alexandra.stn@gmail.com

CLAUDIA ELENA PAMFILOIU
The Faculty of Economic Sciences
Accountancy and financial informatics
CALEA DUMBRAVII nr.17
ROMANIA
pamfiloiuclaudia@yahoo.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

System for voice control of car modules communicating through CAN

Dimitar Stoimenov
Teacher Coordinator: Philip Petrov

Abstract

A system for voice control of car electronic modules is presented in the paper. Speech recognition of the driver voice commands is used. The system is based on duplicating the CAN messages and moving the relevant actuators as certain voice command is issued. The system can be configured for car models with different CAN network specifications. The system comprises a software implementation with intuitive user interface and flexible navigation as well as hardware components additionally installed in the car.

1 Introduction

As technology evolves more interesting and futuristic inventions and engineering solutions are designed and developed in some cases, created just for fun, while others created to facilitate and support. The perceptual human computer interfaces based on speech recognition are the core of the innovative ideas in that filed [1]. Once properly understood and recognized, spoken language can be used by a computer to perform various functions, complex tasks or even manage other devices and machines.

The increasingly widespread use of onboard computers and sophisticated networks in the automobiles allows for the implementation of various multifunctional smart systems. There are many cars with evoice control of the main functions produced by Bentley, Mercedes, BMW. The main disadvantage of these models are their cost makes them not available in mass production. There are many commercial systems for integrating a portable computer system in a car but most of them include only multimedia functions. They provide an opportunity to play music or surf the Internet using voice commands and speech recognition. Even if speech databases are developed for automotive industries [2-5] there is no complete universal system for car voice control able to perform rich control functions as moving windows, mirrors and seats or controlling other car modules with voice commands. The main purpose of such system is a safe driving since finding the buttons for different functions on the car's control panel can be very dangerous while the car is in motion.

A recent car models are equipped with an appropriate interface that can be used for computer based diagnostics. Through various service programs and complex protocols the systems in a car can be examined – either checking the values of certain sensors in real time or moving different executive parts of the car. These complex systems are mostly intended for workshops, diagnostics and repair.

The system suggested in the paper is aimed at voice control of rich set of car's functions. The system is designed to be readily available for a wide range of users. It provides easy connection to the car network and possibility to set voice commands for certain functions according to the user's preferences. The suggested solution uses simplified interface and handling functions as well as using functions of modules already programmed and configured in a car.
The suggested voice control system uses the functionality of the control units programmed by the car's manufacturers. The CAN bus messages are used to activate the executive units or to read the values of certain sensors by the developed system in order to control the car's modules and to add new functionality to the existing CAN network.

2 Controller Area Network (CAN) system and Voice Control Engine

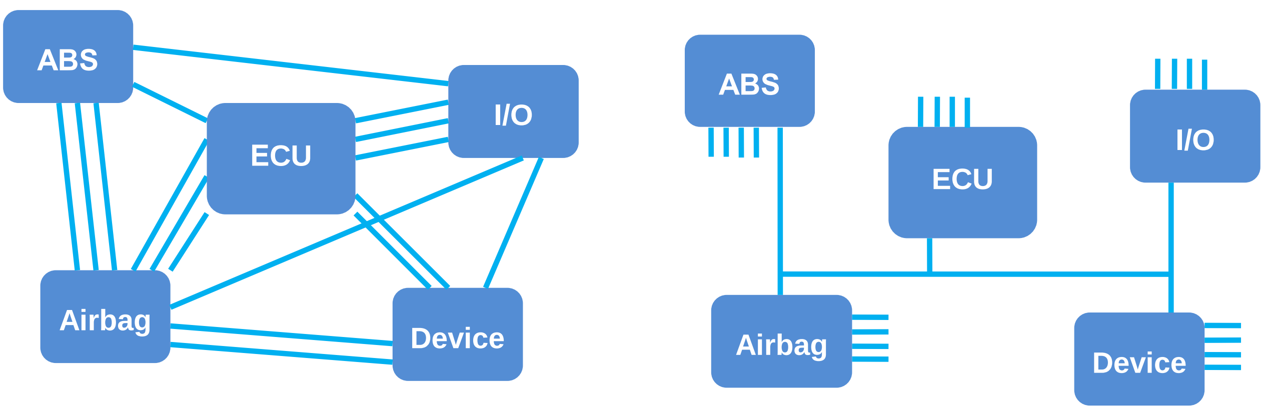
2.1 CAN

2.1.1 Connection

System uses direct connection to the CAN bus of the car. It can receive all messages and transmit different messages when detects a spoken command. The CAN bus is message oriented communication standard and therefore the control units in the car listens all the messages important for them and are not interested about the sender. Thus messages from the already installed units can be copied and functions of the control modules can be duplicated by a human voice commands.

2.1.2 CAN in automotive applications

Controller Area Network is the leading network in power-train and body electronic applications and it is very well accepted by car production companies [7]. One of the biggest car development companies – DaimlerChrysler, Ford and General Motors, are intensively involved in CAN network developments [6]. The difference of car systems with or without a CAN bus is shown on fig.1.



(a) (b)

Fig. 1 Differences between systems organized without (a) and with (b) CAN bus.

2.1.3 Basic organization of electronic units in a vehicle

The CAN bus connects the individual control modules and allows them to exchange information. Sensors are connected to the analog inputs of the control modules as well as various buttons or potentiometers can be connected to analog inputs. Signals from the sensors and controls are transferred to the closest control unit, and then converted into digital format and send in the form of coded messages to the bus. Each control unit can filter such messages, decode information from sensors and decide when and how to send analog signal to the actuator, controlled by it. (fig.2).

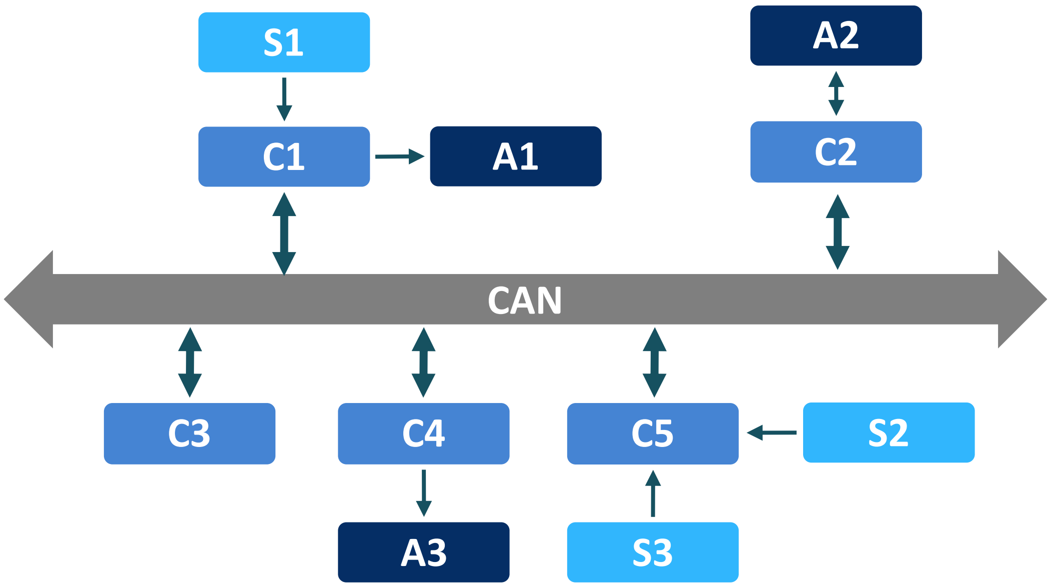


Fig. 2. Principle scheme of CAN related control units in a vehicle.
C – control unit, A – actuator, S – sensor, button.

Four groups of control units can be distinguished in CAN system:

-Control units with activation functions (eg. K3, K5): They are instructed to monitor messages from the bus and if needed to trigger a mechanism. The connection between them and executive mechanisms can be in both directions - then they follow the actual position of the actuator (eg. K3).

-Control modules with sensing functions (eg. K6): Their task is merely to transmit to the bus information about the value of the sensors or the state of buttons connected.

-Control modules with complex functions (eg. K1): Can perform the functions of the two control modules mentioned above. Most of the control units in automobiles are that kind. They do a lot of work at the same time and can handle a lot of data.

-Control modules with management functions (eg. K4): They don't have analog inputs and outputs, but are connected to the bus and depending on how they are programmed, communicate with other modules and transmit messages. Such modules can be used for central gateway between different CAN buses in the car or between cars and diagnostic equipment (like CGW - Central Gateway Module in new models of Mercedes). [8]

An example is given on fig.3 with a simple situation, where by pressing a button in the left front door - S20/1, an actuator move right back window up or down – M10/6. The task is performed in the following steps:

- pressing the button S20/1s4;

- control module in left front door – N69/1, sends a message by the CAN for pressed button S20/1s4;

- the message reaches all the control unites associated with CAN bus;

- control unit N69/4 in the right back door receives the message and activates the actuator for moving the right back window M10/6;

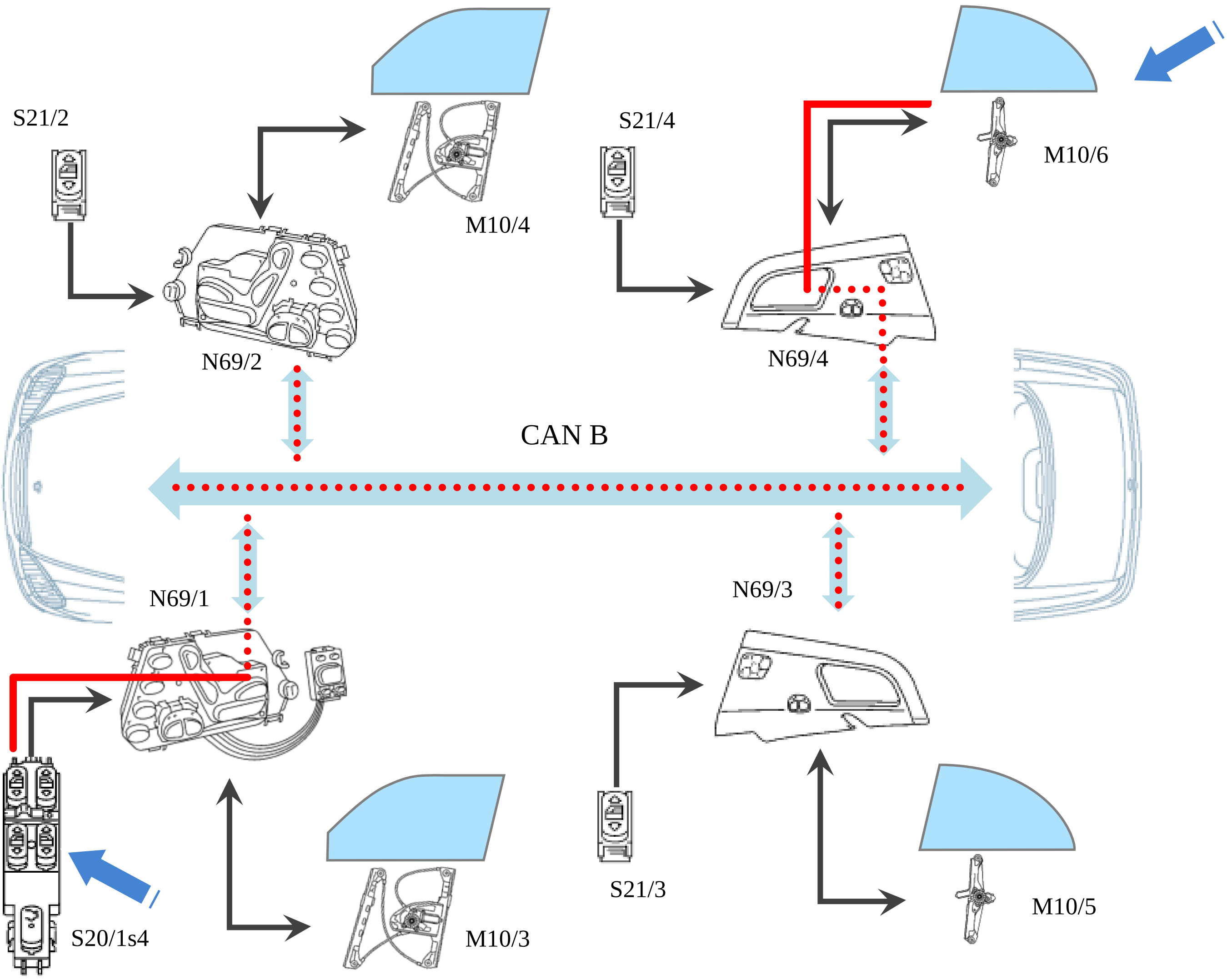


Fig. 3 Moving right back window in a vehicle [8]

2.2 Voice control and speech generator

2.2.1 Speech Recognition Engine

The voice control is a very flexible for control of devices; however not enough “trusted” way to control computer components. Different background noises or change in the human mood may disrupt the voice recognition making it impossible for the computer to parse the command. Therefore a very careful examination on what components of the car exactly can be controlled with voice must be made [3, 4].

For advanced speech capabilities, Microsoft provides native and managed interfaces for developing speech-enabled applications: a COM-based Microsoft Speech API (SAPI) and the .NET Framework 3.0 System.Speech* namespaces. SAPI is middleware that provides an API and a device driver interface (DDI) for speech engines to implement. The speech engines are either speech recognizers or synthesizers. Can be installed different engines for different languages. Although this is opaque to developers, the managed System.Speech* namespace communicates to these engines both directly and indirectly by calling through SAPI (Sapi.dll). Windows Vista supplies default recognition and synthesis speech engines, but this architecture enables plugging in additional ones without changes to applications. Each speech engine is language specific. The System.Speech namespaces are largely built upon and follow the general programming approaches of SAPI 5.3. These namespaces can be used to speech-enable console, Windows Forms, and Windows Presentation Foundation applications. To use this set of managed libraries, a reference to the System.Speech.dll assembly must be added to the project. The speech namespaces include some capabilities not found in SAPI 5.3, including a grammar builder (GrammarBuilder), prompt builder (PromptBuilder), an SRGS document object model (SrgsDocument), and strongly-typed grammars.

2.2.2 Text to speech synthesiser

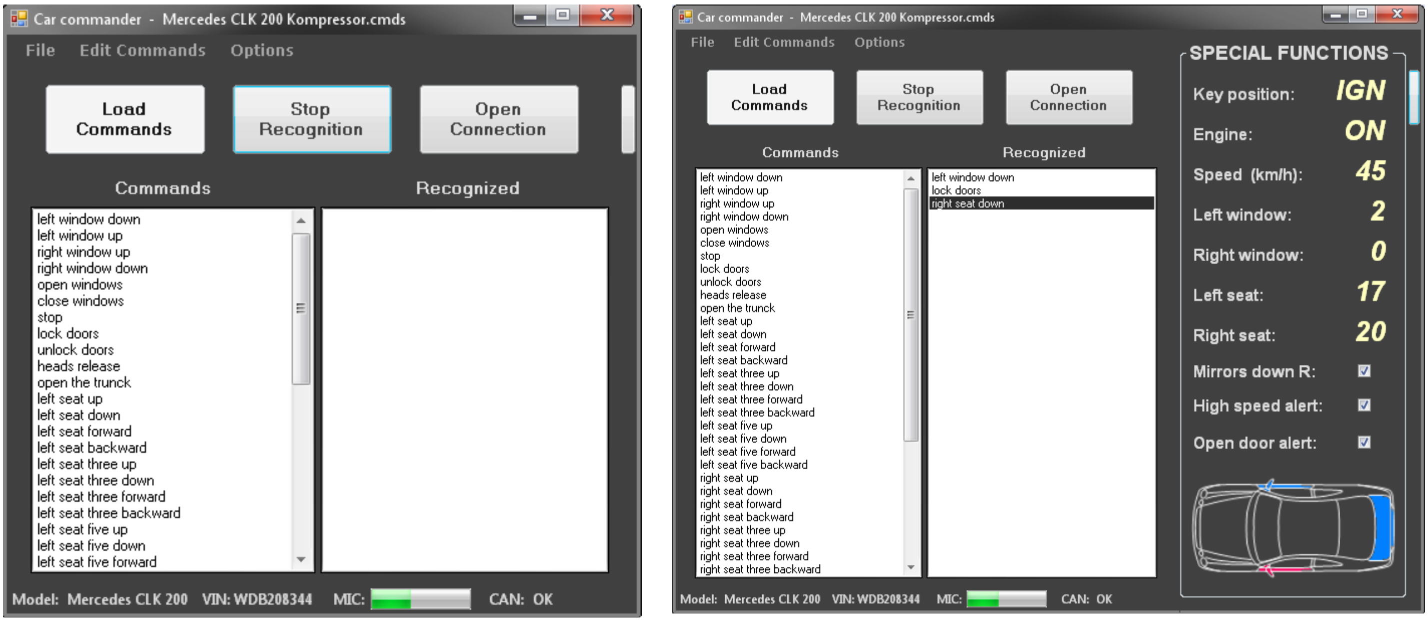
For generating answers of the computer Text to Speech technology (TTS) is used. The computer interacts with the user and responds to the pronounced command or to certain situations. TTS allows applications to stream text from virtually any source for conversion to an audio format. Thus pre-record information for storage as sound files is not required. Also, presenting dynamic data using TTS is one of the deployment approached of speech applications. Some vendors provide TTS with human-like quality and thus speech applications can substitute TTS for voice talents where prompts are used to direct callers. This level of quality presents an opportunity for a rich user interface for speech applications in general. With TTS users can configure the answers of the system as desired.

2.3 System architecture

2.3.1 User interface

The suggested system for voice control in car using CAN is developed as software system wuth separated user and administrative interfaces. The user interface is shown on fig. 4. The simple and easy to use interface provides possibility the system to be used even on small portable computers with touch screen. A functionality to load commands provides an option to load the preferences file and thus move the system from one car to another by only loading car-specific voice commands. The voice recognition mode is enabled either by pressing a button on the user's interface or to avoid distraction specific hardware button can be mounted in the car. Additional option is provided by the user's interface to establish the connection with the CAN network.

More details from the CAN bus can be monitored and updated in real time (fig.4-b) such as mirrors and seats positions, current speed and engine status.

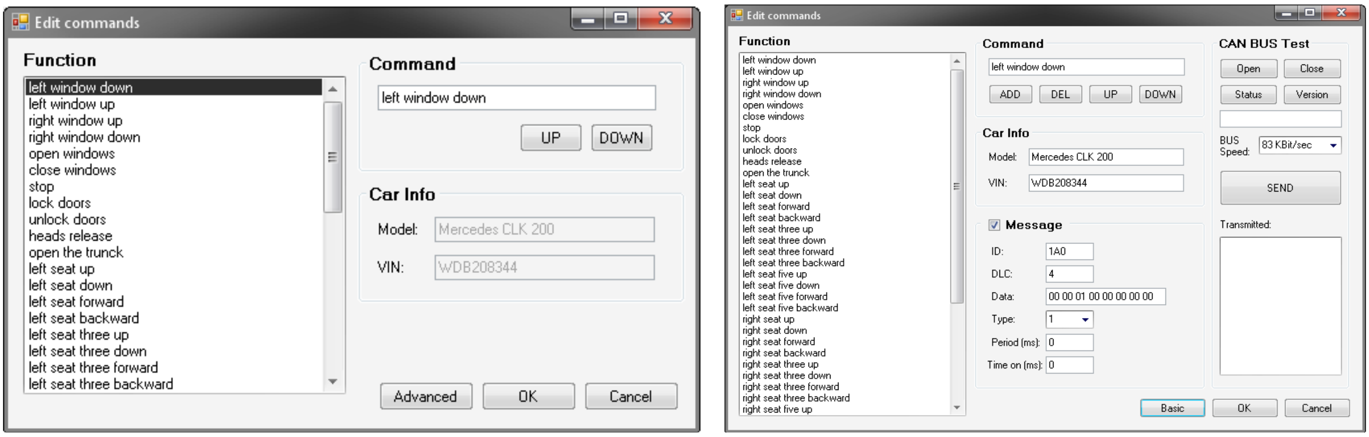


(a) (b)

Fig. 4 User interface. (a) – basic, (b) – extended.

2.3.2 Administrative interface

The administrative interface is shown on fig. 5 and provides options for adding, deleting and editing voice commands from the configuration files. An advanced mode is also provided in the administrative panel given the option to add completely new car features which are not yet recognized and are still not added to the software system.



(a) (b)

Fig.5 Administrative interface. (a) – basic, (b) – advanced.

Using the advanced mode the system is “sniffing” the current CAN network traffic in order to allow new features to be recorded and voice commands to be added accordingly. This makes the system well portable between different car models. Every voice command for a specific function is customizable and can be changed according to the user preferences and requirements.

2.4 Adding special features to the car functions

The developed system uses actual values from different units and depending on their values it controls actuators or carry out other functions. Additionally new features are also suggested and provided by the voice control system:

1. "Mirrors down R" – moving mirrors down when reversing. In reverse gear the system sends a signal to left and right door control modules for the movement of mirrors in a downward direction for 2 seconds. This gives the driver good view to the rear wheels and make parking easier. In exclusion of reverse gear the system automatically returns both mirrors in their previous state. (actual value – the position of gear selector; controlled units – left and right door);

2. "High speed alert" – a warning for speeding. Computer - synthesized voice alerts the driver that the speed is increasing. The maximum allowed speed before the alert message is not adjustable at the moment. Future system extension will use connection to a GPS receiver and real time update of the maximum speed. The warning is repeated every 15 seconds. (actual value – vehicle speed; function – voice warning);

3. "Open door alert" – warning for an opened door. Computer synthesized voice alerts the driver for an open door if speed is bigger than 10mph. (actual value - speed and doors knobs, function - voice warning).

3 Results and analysis

The suggested voice control system was implemented and installed on Mercedes CLK 200 year 2001 – a car without any stock voice recognition capabilities. All the experiments show that the voice commands of car devices provide flexible and comfortable features which may bring the hands free driving at further level. The fully functional system uses a microphone, a button to enable voice commands, a notebook and a CAN adapter (fig.6).

(a) (b) (c)

Fig.6 System components (a) – microphone, (b) – button, (c) – notebook.

4 Conclusion

The versatility of the system depends on the opportunities provided by the target vehicle. Automobile manufacturers organize different CAN communications in different cars. The

suggested voice control system allows for duplication of functions of the modules in the car such as moving windows, seats and mirrors in all directions, opening the trunk, locking and unlocking the doors.
Future extension of the system will connect a GPS receiver and based on data received maximum allowed speed will be dynamically determined or a destination will be selected through voice commands. Another possibility is adding a module for manipulating multimedia car features as for example start music player with a voice commands. Loading command dictionaries will allow desired track to be searched and automatically played using speech recognition of the voice command.
The system can be developed for smart phones or pads and the connection with the car can be made via bluetooth. This eases the user and make it possible to use the system immediately after entering the car. The app on the smart phone can communicate with the car by an OBD2 Blue tooth dongle, and send CAN messages. Using of only one adapter for the car and heands free for the smart phone will make the installation of the system very easy and every user can set it by himself.
Further system extension will be the development of electric circuit for USB based communication with the program. In this way even the car modules not related to the CAN bus will be controlled thus making the suggested solution available in automobile without CAN network.

References

[1] A. Dix, J. Finlay, G. Abowd, *Human-computer interaction*, Pearson Education, 2004.
[2] P. Geutner, L. Arevalo, J. Breuninger, *VODIS - Voice-Operated Driver Information Systems: A Usability Study on Advanced Speech Technologies for Car Environments*, Proc. of the Sixth Int. Conf. on Spoken Language Processing: ICSLP, 2000, vol.4, pp.378-382.
[3] A. Moreno, B. Lindberg, C. Draxler, G. Richard, K. Choukry, S. Euler, J. Allen, *SpeechDat Car: A Large Speech Database For Automotive Environments*, Proc. of 2nd Int. Conf. on Language Resources and Evaluation, Athens, 2000.
[4] M. Matassoni, M. Omologo, and P. Svaizer, *Use of real and contaminated speech for training of a hands-free incar speech recognizer*, Eurospeech, 2001.
[5] D. Langmann, H. R. Pfitzinger, T. Schneider, R. Grudszus, A. Fischer, M.Westphal, T. Crull, and U. Jekosch, *CSDC – the MoTiV car speech data collection*, Proc. of Int. Conf. Lang. Resources and Eval., 1998, pp. 1107–1110.
[6] St. Djiev, *Industrial networks for communication and control*, TU-Sofia Press, 2003 (in Bulgarian language).
[7] M. Farsi, K. Ratcliff, M. Barbosa, *An overview of controller area network*, Journal Computing & Control Engineering, vol.10, №3, 1999, pp.113-120.
[8] Mercedes *StarFinder* Catalog, 6/2008.

DIMITAR STOIMENOV
Technical University of Sofia
Computer systems and technologies
BULGARIA
E-mail: dimitar.ns@gmail.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Edge Detection in Spatial Domain

Jelena Tasic, Igor Valjevic
Teacher Coordinator: Milan Tuba

Abstract

This paper introduces two classical approaches to detecting edges in images, which, respectively, make use of the first and second derivative of the intensity values across the image. A particular attention is given to the assumptions implicitly made by these approaches as to what makes an edge in the image, drawing a particular distinction between the discontinuities in the gradient and the user-defined, content-sensitive notions of the edge. The methods that are discussed in more detail include Gradient, Roberts Cross, Sobel, Prewitt and Kirsch (examples of first order differential methods), as well as, Laplacian and Laplacian of Gaussian (examples of second order differential methods).

Keywords: Digital image, Image processing, Edge detection, Spatial domain

1 Introduction

Presumably, the purpose of detecting sharp changes in brightness in natural images is to capture important events and changes in properties of the world that are often reflected by such changes. In general, the working assumption is that any discontinuities in image brightness are likely to correspond to some discontinuities in the world, whether these indicate variations in depth, surface orientation, material properties, or scene illumination. Ideally, therefore, applying an edge detector to an image would return a set of connected curves that indicate the boundaries of objects, materials, markings and other features of human interest. In such an ideal case, edge detection algorithms may significantly simplify processing of visualizing data by filtering out the mass of information that is typically regarded as less relevant. For this reason, edge detection is one of the fundamental steps in image processing, image analysis, image pattern recognition, and computer vision techniques (for introduction for edge detection see [1], [2] and [3]).

In practice it is seldom possible to obtain ideal edges from real life images of even moderate complexity. For all the success stories that have marked the history of the vigorous research in the area of edge detection, it remains a hot topic and continues to attract the attention of new generations of researches. A telling example of its ongoing appeal as a research problem is its role in the recent development of content aware methods for image resizing. All the issues to do with the subjectivity of what makes an edge in an image play out in content aware resizing in a particularly direct an intuitive way. Here, too, the simple gradient base definition of the edge takes one a long way, yet, in addressing its remaining limitations one quickly runs into all the subtleties the content-aware notion of the edge that refer it, beyond the image, to objects and features of the world of human interest. To this end, as with other uses of edge detection, a more satisfying solution inevitably makes use of user interaction. (For the original research, see [4]. For an example of the many interesting spin-offs, see [5]).

At the same time, fully automatic edge detection is proving essential in a growing number of domains. For example, in a recent work [6], the edge information of an image has been used to generate its signature set. As this information is particularly robust, it allows detection of copies of digital images (e.g. transmitted over the Internet) even where these have undergone substantial modifications such as rotation, shifting, unconstrained resizing and cropping.

Examples of other recent and ongoing interesting research include work that has successfully combined edge detection with region growing for adaptive threshold techniques [7], generalization of edge detection to contour detection for image segmentation [8], an image quality based method involving hexagonal edge detection, thinning algorithm and twin determination [9], an application of the emerging topological gradient method to edge detection [10], and the development of the framework for detecting the image edge based on the sub-pixel multi-fractal measure [11].

2 Implementation

A demonstration program has been written in support of the comparative discussion of methods for edge detection below. The program features implementations of the main examples of edge detection solutions. It is written in C# without a recourse to any specialized libraries. It takes as input common image formats, such as JPEG, PNG, and BMP, and allows output to be saved likewise. The program implements four first derivative methods: Gradient, Sobel, Roberts, Prewitt, and Kirsch, allowing for easy comparison of their performance and the quality of the results. Similarly, the program features implementations of two second order differential methods: Laplacian and Laplacian of Gaussian. A comparative discussion of these methods is the main goal of this paper.

3 Digital Image Processing in Spatial Domain

Vision is the most advanced of our senses. Given that so much of the human brain is dedicated to it, it is not surprising that images play such an important role in human affairs. Even though the field of digital image processing is based on mathematical and probabilistic formulations, visual intuition and subjective visual evaluation play central roles in the choice of one technique over another.

Image can be defined as a two dimensional function $f(x,y)$, where x and y are spatial coordinates, and the value of f at any pair of coordinates represents channel intensity or gray level of the image at that point. When x, y and the value of f are discrete quantities we call such $f(x,y)$ a digital image. The term spatial domain refers to the image plane itself, and approaches in this category are based on direct manipulation of pixels in the image, unlike frequency domain processing techniques which are based on modifying the Fourier transform of the image.

Spatial domain processes are denoted by expression:

$$g(x,y) = T(f(x,y))$$

where $f(x,y)$ is the input image, $g(x,y)$ is the processed image, and T is an operator on f, defined over some neighborhood of (x,y). The simplest form of T is when g depends only on the value of f at (x,y), which is to say when the neighborhood is of size 1x1, i.e. a single pixel. The effect of this transformation can be to produce an image of higher or lower contrast, brightness, or threshold, or a negative. Larger neighborhoods allow more flexibility. Neighbors of pixel (x,y) can be represented by a matrix. For example, a 3x3 neighborhood can be represented as:

$f(x\text{-}1,y\text{-}1)$	$f(x,y\text{-}1)$	$f(x+1,y\text{-}1)$
$f(x\text{-}1,y))$	$f(x,y)$	$f(x+1,y)$
$f(x\text{-}1,y+1)$	$f(x,y+1)$	$f(x+1,y+1)$

4 Edge Detection

Like shadows, edges are not in themselves physical entities. Moreover, what appears to be an edge depends on where one is looking from. Coming closer to the scene of interest, one often finds that the apparently clear and simple edges one saw earlier now themselves feature more edges in turn. Typically, edges correspond to significant local changes of intensity in the image. Intuitively, an edge is a set of connected pixels that lie on the boundary between two regions. Intensity changes are caused by various physical phenomena involving color, texture, reflection, and shadows. The goal of edge detection is to produce a line drawing that succinctly reconstructs the scene from its image. The features of the scene thus indicated by the drawing as straight lines, curves, and corners can then be used by higher level computer vision algorithms.

Classical methods of edge detection involve an operator which is sensitive to large gradients in the image while returning the value of zero in uniform regions. Variables that affect such operators include edge orientation, edge structure, and noise levels. They can be optimized to look for horizontal, vertical or diagonal edges. Edge detection is difficult in noisy images because both noise and the edges of interest contain high-frequency content. Moreover, not all edges involve a step change in intensity. Effects such as refraction or poor focus can result in objects with boundaries defined by more or less gradual changes in intensity.

There are many ways to perform edge detection. Many of the existing methods may be grouped into two categories, distinguishing between those that make use, respectively, of the first and the second-order derivative of the intensity values across the image. First-order differential methods detect the edges by looking for the maximum and minimum in the first derivative. Second derivative methods search instead for zero crossings in the second derivative. An edge has a one-dimensional shape of a ramp and calculating the derivative of the image can highlight its location. First-order derivatives generally produce thicker edges and tend to have a stronger response to a gray-level step. Second-order derivatives have stronger response to fine detail, such as thin lines and isolated points, and produce a double response at step changes in gray level. The type of filter to be used depends on the application.

4.1 First-Order Differential Methods of Edge Detection

First-order derivatives of digital image are based on various approximations of the 2-D gradient. The gradient of an image $f(x,y)$ at location (x,y) is defined as a vector

$$\nabla = \begin{bmatrix} Gx \\ Gy \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

The magnitude of this vector is given by:

$$\nabla f = \mathrm{mag}(\nabla) = [Gx^2 + Gy^2]^{\frac{1}{2}} = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial x}\right)^2\right]^{\frac{1}{2}}$$

The gradient orientation of this vector is given by:

$$\alpha(x,y) = \tan^{-1}\left(\frac{Gy}{Gx}\right)$$

The gradient magnitude gives the amount of difference between pixels in the neighborhood (the strength of the edge). The gradient orientation gives the direction of the greatest change, which presumably is the direction across the edge. Many algorithms use only the gradient magnitude. The components of the gradient vector itself are linear operators, but the magnitude of this vector obviously is not because of the squaring and square root operations. Even though it is not strictly correct, the magnitude of the gradient vector is often referred to as the gradient. The equation of gradient magnitude is difficult for computational implementation, which is why we approximate magnitude of the gradient by using absolute values instead of squares and square roots:

$$\nabla f \approx |Gx| + |Gy|$$

This equation is simpler to compute and it still preserves relative changes in gray levels, but the isotropic feature property is lost in general.

4.1.1 Estimating the Gradient with Finite Differences

$$\frac{\partial f}{\partial x} = \lim_{h \to 0} \frac{f(x+h,y) - f(x,y)}{h}$$

$$\frac{\partial f}{\partial y} = \lim_{h \to 0} \frac{f(x,y+h) - f(x,y)}{h}$$

Gradient can be approximated by finite differences:

$$Gx = \frac{\partial f}{\partial x} = \frac{f(x+hx,y) - f(x,y)}{hx}) = f(x+1,y) - f(x,y), (hx = 1)$$

$$Gy = \frac{\partial f}{\partial y} = \frac{f(x,y+hy) - f(x,y)}{hy}) = f(x,y+1) - f(x,y), (hy = 1)$$

We can implement Gx and Gy using masks:

$$Gx = \begin{bmatrix} -1 & 1 \end{bmatrix} \text{ and } Gy = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

We can now find just the vertical, or just the horizontal edges using masks Gx and Gy respectively. Or we can use both together.

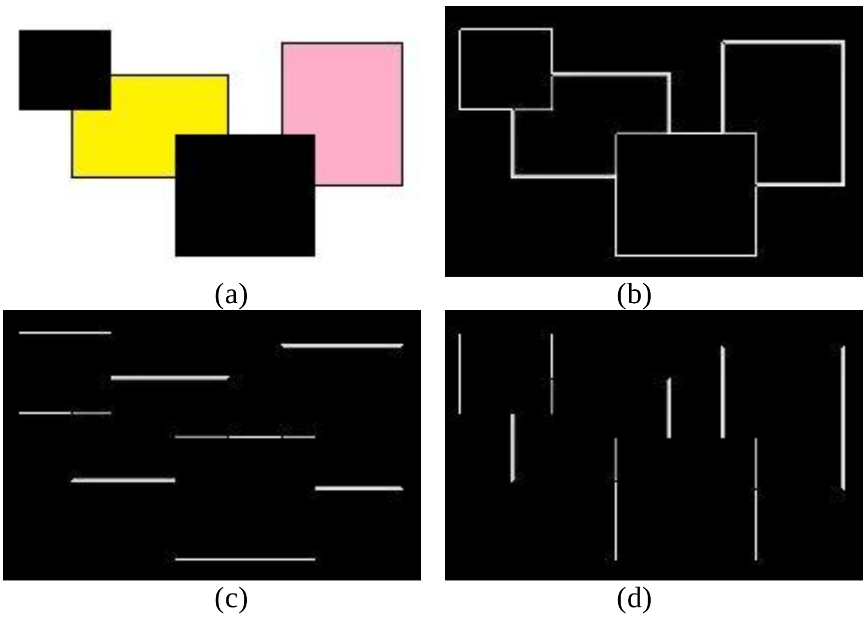


Figure 1. Original picture (a), edge detection using gradient filter (b), detecting only horizontal edges (c), and detecting only vertical edges (d).

4.1.2 Roberts Cross

The Roberts Cross operator performs a simple, quick to compute, 2-D spatial gradient measurement on the image. The operator consists of a pair of 2×2 convolution kernels:

$$Gx = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } Gy = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

One kernel is simply the other rotated by 90°. This is very similar to the Sobel operator. Roberts Cross uses the diagonal directions to calculate the gradient vector. The kernels can be applied separately to the input image to produce separate measurements of the gradient component in each orientation. These can be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient.

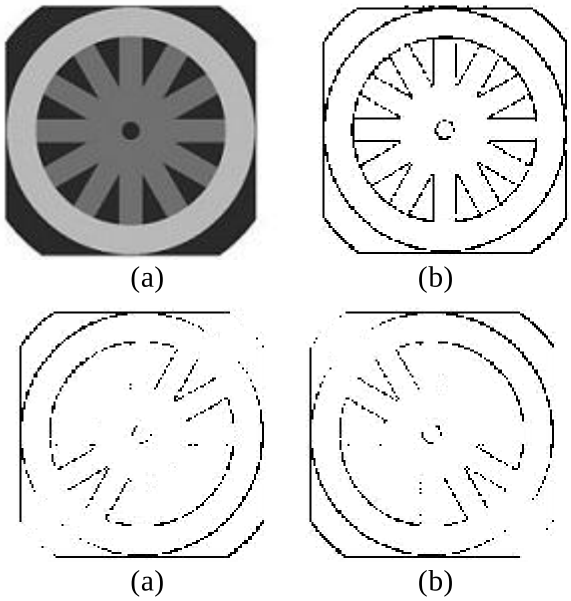


Figure 2. Original picture (a), edge detection using Roberts Cross operator (b), edge detection using only kernel Gx (c), and edge detection using only kernel Gy (d).

The main reason for using the Roberts Cross operator is that it is very quick to compute. Only four input pixels need to be examined to determine the value of each output pixel, and only subtractions and additions are used in the calculation. In addition, there are no parameters to set. Its main disadvantages are that, since it uses such a small mask, it is very sensitive to noise. It also produces very weak responses to genuine edges unless they are very sharp. The Sobel operator performs much better in this respect.

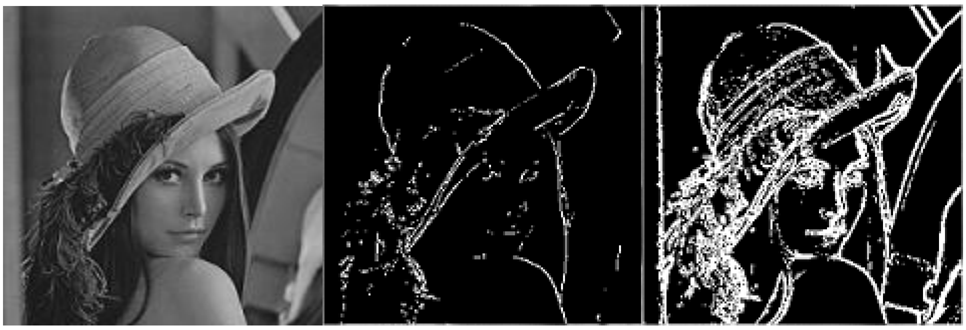
4.1.3 Sobel Edge Detector

The Sobel edge detector uses two convolution kernels, one to detect changes in vertical contrast (Gx) and another to detect horizontal contrast (Gy). One kernel is simply the other rotated by 90°. This is very similar to the Roberts Cross operator. The operator consists of a pair of 3×3 convolution kernels:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations.

The Sobel operator is slower to compute than the Roberts Cross operator, but its larger convolution kernel smooth the input image to a greater extent and so makes the operator less sensitive to noise. The operator also generally produces considerably higher output values for similar edges, compared with the Roberts Cross.



(a) (b) (c)

Figure 3. Original picture (a), edge detection using Roberts Cross operator (threshold 90) (b), and edge detection using Sobel operator (threshold 90) (c).

4.1.4 Prewitt Edge Detector

The Prewitt kernels, as in Sobel, are used for detecting vertical and horizontal edges in images.

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

The Prewitt kernels also rely on central differences, but Sobel kernels give greater weight to the central pixels when averaging. The operator calculates the gradient of the image intensity at each point, giving the direction of the largest possible increase from light to dark and the rate of change in that direction. At each image point, the gradient vector points in the direction of largest possible intensity increase, and the length of the gradient vector corresponds to the rate of change in that direction. This implies that the result of the Prewitt operator at an image point which is in a region of constant image intensity is a zero vector and at a point on an edge is a vector which points across the edge, from darker to brighter values.

Figure 4. Edge detection using Prewitt operator – threshold 90 of picture from Figure 3 (a).

4.1.5 Kirsch Edge Detector

Given k operators, g_k is the image obtained by convolving $f(x,y)$ with the k-th operator. The k defines edge detection. The gradient is defined as:

$$g(x,y) = \max_k g_k(x,y)$$

These equation do not explicitly calculate the gradient. Instead, it calculates first-derivatives in specific directions. Taking the result that produces the maximum first derivative, we can approximate the gradient magnitude. The orientation is limited to these specific directions. The operator consists of eight 3×3 patterns for each direction: south, south-east, east, north east, north, north west, west and south west respectively:

$$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$$

$$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

Figure 5. Edge detection using Kirsch operator - threshold 90 of picture from Figure 3 (a).

4.1.6 Algorithm for Finding Edges Using Kernels

1. Smooth the input image $\hat{f} = f(x,y) * G(x,y)$
2. $fx = \hat{f}(x,y) * G_x(x,y)$
3. $fy = \hat{f}(x,y) * G_y(x,y)$
4. $mag(x,y) = |\hat{f}_x| + |\hat{f}_y|$
5. if(mag(x,y) > Threshold), than possible edge point

It is important to note that these definitions do not guarantee success in finding edges in an image. They simply give us a formalism to look for them.

4.2 Second-Order Differential Methods of Edge Detection

Differentiating the first derivative of the gradient magnitude gives us the second derivative of the intensity values across the image. Accordingly, finding maxima of gradient magnitude corresponds to finding places where the second derivative is zero. However, when applying differential operators to images, the zeroes rarely fall exactly on a pixel. Typically, they fall between pixels. We can isolate these zeroes by finding zero crossings instead – places where one pixel is positive and a neighbor is negative, or vice versa. One nice property of zero crossings is that they provide closed paths, except, of course, where the path extends outside the image border. On the other hand, there are at least two problems common to zero-crossing methods: they produce two-pixel thick edges (the positive pixel on one side and the negative on the other), and they can be extremely sensitive to noise.

For two dimensions, there is a single measure, similarly to the gradient magnitude, of second derivatives. Consider the dot product of ∇ with itself:

$$\nabla \cdot \nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

∇^2has special name and is called the Laplacian operator. When we apply it to a function, we get:

$$\nabla^2 f = \left(\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \right) f = \frac{\partial^2}{\partial x^2} f + \frac{\partial^2}{\partial y^2} f$$

One interesting property of the Laplacian is that it is rotationally invariant – it does not depend on what directions you choose, so long as they are orthogonal. The sum of the second derivatives in any two orthogonal directions is the same.

4.2.1 Laplacian

Gradient operation is an effective detector for sharp edges where the pixel gray levels change over space very rapidly. But when the gray levels change slowly from dark to bright, the gradient operation will produce a very wide edge. It is helpful in this case to consider using the Laplace operation. This method locates an edge by finding the zeros in the second derivative, because when the first derivative is at a maximum, the second derivative is zero.

Laplacian filter is the simplest isotropic filter, whose response is independent of the direction of the discontinuities in the image to which the filter is applied. In other words, isotropic filters are rotation invariant, in the sense that rotating the image and then applying the filter gives the same result as applying the filter to the image first and then rotating the result.

Partial second-order derivative in the *x*-direction:

$$\frac{\partial^2 f}{\partial^2 x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

Similarly in the y-direction:

$$\frac{\partial^2 f}{\partial^2 y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

The digital implementation of the two-dimensional Laplacian is obtained by summing these two components:

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)]$$

This equation can be implemented using the mask:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

which gives an isotropic result for rotations in increments of 90°. The diagonal directions can be incorporated in the definition of the digital Laplacian by adding two more terms in equation, one for each of the two diagonal directions.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

This mask yields isotropic results for increments of 45°. Two other implementations of the Laplacian are:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Those masks are used frequently in practice. They yield equivalent results as previous two, but the difference in sign must be kept in mind when combining (by addition or subtraction) a Laplacian-filtered image with another image.

The Laplacian generally is not used in its original form for edge detection because of unacceptably noise sensitivity. Moreover, the magnitude of the Laplacian produces double edges – an undesirable effect because it complicates segmentation. The Laplacian is, likewise unable to detect edge direction. Because of that Laplacian is used for finding zero-crossing property for edge location or for the complementary purpose of establishing whether a pixel is on the dark or light side of an edge.

Figure 6. Laplacian.

4.2.2 Laplacian of Gaussian (LoG)

One way to deal with the sensitivity of the Laplacian operator on the presence of noise is to blur the image beforehand. We first smooth out the noise in the image with a Gaussian operator and then apply the Laplacian operator. Convolution is associative and commutative, that is why we can combine the Gaussian smoothing operator with the Laplacian operator (by convolving them one with another) to form a single edge detection operator. This has two advantages. Since both the Gaussian and the Laplacian kernels are usually much smaller than the image, this method usually requires far fewer arithmetic operations. The LoG kernel can be calculated in advance so only one convolution needs to be performed on the image at run-time.

The 2-D LoG function centered on zero and with Gaussian standard deviation has the form:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right)e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

$$\begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 2 & 1 & 1 & 0 \\ 1 & 2 & 4 & 5 & 5 & 5 & 4 & 2 & 1 \\ 1 & 4 & 5 & 3 & 0 & 3 & 5 & 4 & 1 \\ 2 & 5 & 3 & -12 & -24 & -12 & 3 & 5 & 2 \\ 2 & 5 & 0 & -24 & -40 & -24 & 0 & 5 & 2 \\ 2 & 5 & 3 & -12 & -24 & -12 & 3 & 5 & 2 \\ 1 & 4 & 5 & 3 & 0 & 3 & 5 & 4 & 1 \\ 1 & 2 & 4 & 5 & 5 & 5 & 4 & 2 & 1 \\ 0 & 1 & 1 & 2 & 2 & 2 & 1 & 1 & 0 \end{bmatrix}$$

(a)

(b)

Figure 7. (a) Discrete approximation to LoG function with Gaussian σ = 1.4, and (b) Laplacian of Gausian.

5 Conclusion

Since edge detection is the initial step in object recognition, it is important to understand the differences between the various edge detection techniques. In this paper we briefly discussed the most commonly used methods. Certainly, all of the methods described here can effectively detect edges in an image, yet each to some degree struggles with the noise and the various optical artifacts one invariably encounters in practical, real-life applications. And these are only some of the more common problems that motivate ongoing exploration of very different approaches to edge detection that were not reviewed in this short paper. Some systems, for example, involve learning algorithms, millions of training data sets, and custom-built hardware. Only, as yet, none of the existent systems come close to human perception in unconstrained real-world conditions. Perhaps, it is not then that surprising that for many critical practices it is still necessary to enlist human help. One particularly telling new trend is the growing reliance of computer vision and other scientists on crowdsourcing Internet marketplaces such as the Amazon Mechanical Turk as the source of human generated training sets. Interestingly, as reported e.g. in [12], not only that human labeling is necessary for machine classification, but also the recent studies of how humans make mistakes while labeling is becoming instrumental for learning better generalizing features and machine classifiers.

References

[1] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Prentice Hall, New Jersey, 2002.
[2] W.K. Pratt. *Digital Image Processing*. PixelSoft, Los Altos, CA, 2007.
[3] S. Jayaraman, S. Esakkirajan and T. Veerakumar. *Digital Image Processing*. TMH, New Delhi, 2010.
[4] S. Avidan and A. Shamir. Seam Carving for Content-Aware Image Resizing. *ACM Trans. Graph.* 26, no. 3: 10, 2007.
[5] Y.-S. Wang, C.-L. Tai, O. Sorkine and T.-Y. Lee. Optimized Scale-and-Stretch for Image Resizing. *ACM Trans. Graph.* 27, no. 5: 1–8, 2008.
[6] C.C. Lin and S.S. Wang. An Edge-Based Image Copy Detection Scheme. *Fundamenta Informaticae*, 83: 299–318, 2008.
[7] C.H. Lin and C.C. Chen. Image Segmentation Based on Edge Detection and Region Growing for Thinprep-Cervical Smear. *International Journal of Pattern Recognition and Artificial Intelligence*, 24: 1061–1089, 2010.
[8] H.Z. Wang and J. Oliensis. Generalizing Edge Detection to Contour Detection for Image Segmentation. *Computer Vision and Image Understanding*, 114: 731–744, 2010.
[9] D. Chen, G.-T. Lui, J.C. Kuo. Application of Edge Detection Method Based on Image Quality Gradient for Twin Detection. *Journal of Microscopy*, 236: 44–51, 2009.
[10] L.J. Belaid. Application of the Topological Gradient to Image Restoration and Edge Detection. *Engineering Analysis with Boundary Elements*, 32: 891–899, 2008.
[11] G. Wang, L. Xiao and A.Z. He. An Improved Computing Method for the Image Edge Detection. *Chinese Optics Letters*, 5: 79–81, 2007.
[12] C. Pavlopoulou and S.X. Yu. Indoor-Outdoor Classification with Human Accuracies: Image or Edge Gist? *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA: 41–7, 2010.

Jelena Tasic
University of Belgrade
Faculty of Mathematics
Studentski trg 16, 11000 Belgrade
SERBIA
E-mail: jelenatasicsd@yahoo.com

Igor Valjevic
University of Belgrade
Faculty of Mathematics
Studentski trg 16, 11000 Belgrade
SERBIA
E-mail: igor.valjevic@gmail.com

Prof. Milan Tuba
University Megatrend Belgrade
Faculty of Computer Science
Bulevar umetnosti 29, Belgrade
SERBIA
E-mail: tuba@ieee.org

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Quantization in the JPEG image compression algorithm

Ana Tuba, Marko Milosevic
Teacher Coordinator: Milan Tuba

Abstract

This paper describes the quantization in the JPEG image lossy compression algorithm. JPEG algorithm is not a fixed standard but a set of recommendations. It uses discrete cosine transform on blocks of 8*8 pixels transforming 64 light intensity values to 64 frequency coefficients. Main compression is done by neglecting less important frequency coefficients. Which coefficients are less important is determined by the process of quantization which can be adjusted for different types of images and different applications. Some basic and some more advanced quantization methods are described. Software system for experimenting with various quantization methods and parameters is developed with corresponding graphical user interface.

Keywords: Digital image, Image compression, JPEG algorithm, JPEG quantization

1 Introduction

In the last twenty years many benefits of digital technology have led to its massive use for processing, storing and displaying images in different areas like graphic arts, medicine, satellite images, digital cameras etc. The problem with digital technology is that the representation of images in digital format generates a lot of data. The size of a typical digital image can be tens of megabytes. One solution to this problem is the use of compression techniques. Requirement for such compression algorithm is a very high degree of compression (tens or even hundreds of times), but fortunately exact decompression is not required. Lossy compression algorithms are known to have very high degree of compression with minimal loss in quality during decompression.

JPEG (Joint photographic experts group) compression algorithm was developed for image compression. It is not a fixed standard but a set of recommendations. The quantization is a main part where compression is archived. There are various ways of doing quantization since there is not a unique best way. Different types of quantization are appropriate for different types of images and different applications. It is possible to have different quantization in different regions of the image [1], specialized quantization matrixes for photos with human face [2], or even more specific, very low resolution human face image for mobile phone [3]. Quantization can be specifically adjusted to for recognition of iris in human eye for biometric purpose [4]. Quantization tables can also be designed to help separate images that have been processed by software from those that have not [5]. Some quantization tables are recommended with the JPEG standard. They are determined empirically and adjusted to human vision visual system, but research for alternatives continues [6].

A software system for experimenting with JPEG quantization is developed in C#. As an object-oriented system it is easy for maintenance and modifications and also has appropriate graphical user interface.

This paper is organized as follows: in Section 2 basics of digital images, compression and signal processing are introduced. In Section 3 the JPEG algorithm is described. Section 4 deals with the main topic of this paper: quantization and its various forms: threshold, a fixed number of coefficients and custom empirical matrixes for different applications.

Proposed methods of quantization tested with our software system facilitate rapid development of specific quantization tables for different types of images and different applications.

2 Digital images

2.1 Digitized image

Digital images are nowadays used almost everywhere and important characteristics is that they can easily be processed by mathematical methods [7], [8], [9]. The starting point for digital image is projection of the 3-dimensional world to a 2-dimensional rectangle, screen. This rectangle is first discretized in space, horizontally and vertically into small fixed size rectangles (usually squares) called pixels. Pixels should be small enough so that human eye would not identify them as separate elements, but perceive as a continuous image. Such digitized image is also celled raster image and the number of pixels horizontally and vertically represents the resolution of the image. Raster image is archived by recording for each pixel its brightness and possibly color information.

Depending on the number of colors represented, each pixel brightness will be coded with appropriate number of bits. For black and white images one bit (0 - white, 1 - black) is needed, while for a number of shades of gray a larger number of bits is required (e.g. 256 shades of gray - 8 bits). Color images are more memory demanding. Color specific image element is defined in the RGB color model with three numbers that represent the shades of red, green and blue component. The number of bits per picture element which are required to describe the color is called color depth. When viewing the archived images, it is necessary to interpret the data for each image element in the same order in which they were written. In this way, it is not necessary to write the position of each image pixel and it simplifies the writing and reconstruction of images. Usually the images are stored row by row, from left to right and top to bottom. Rows of image elements are called raster scan lines. The data needed for the reconstruction of recorded lines are: length of raster-scan lines (the number of picture elements on horizontal axis), the number of raster-scan lines (the number of picture elements in vertical axis) and the number of bits that is used to describe the color of each image element (color depth).

Popular and simple file format that is used in our software system is BMP without compression and 24 bit color depth since it is the easiest format for manipulation and in this experimental system we are not concerned with performance issues.

2.2 Compression

Compression algorithms code information in a different way so that it requires less space for storage. Compression is possible only when original information contains redundancy i.e. when its elements are not independent. All data compression algorithms can be classified into two categories: lossless compression and compression with losses. With lossless compression after decompression exactly the original information is reconstructed. Lossy compression sacrifice some data so the decompressed information is not exactly equal to the original. The differences are usually not visible to the human eye.

Lossless image compression is used in those cases where it is difficult or impossible to recover an image or when uncompressed image contains some important information that could be damaged in the process of compression (for example, medical images). On the other hand, image compression algorithms with losses are used in cases where it is easy to repeat the recording process or when some degree of loss of information can be tolerated (video telephony, television, multimedia systems, etc.).

All compression methods are based on the use of unnecessary physical information contained in digital images. Redundant information in the image appears because of space correlation of pixels. This redundancy of information in the image can be used to predict the values of pixels on the basis of its local neighbors.

2.3 Signal processing - Fourier transform

French mathematician Joseph Fourier discovered that any periodic function can be represented as a combination of sine and cosine functions of different frequencies with corresponding amplitude coefficients. This can also be applied to non-periodic function in some interval by repeating that interval. Signals in the real world usually are not periodic but by using this extension any function of interest can be considered.

In the space (or time) domain information is given as values of the signal at different locations or at different times. In case of digital images that signal is brightness at particular pixel. By Fourier transform these brightness values are transformed into amplitude coefficients that multiply different frequency sines or cosines. In that frequency domain it is easy to do certain transformations like noise removal since noise is usually in high frequencies. By inverse Fourier transform the image is then turned back into spatial domain.

3 JPEG algorithm

JPEG is most widely used standardized procedure for image compression. The name JPEG comes from the Joint Photographic Experts Group, which is the name of the association that proposed the standard. JPEG is designed for compression of color images as well as black and white or gray-scale images. It is used for photographs and it is not very suitable for text and technical drawings.

The main problem with digital images is their size: typical photograph size today is of the order of tens of megabytes. In raster graphics, digital images are stored in a matrix of pixels and make the structure suitable for compression, since neighboring pixels are often identical or differ by a small value. Most standard format for bitmap graphics (e.g. GIF, PNG) support lossless compression methods, which are usually some variation of run-length encoding. The advantage of these methods is that compression and decompression are simple and fast, but they do not achieve high degree of compression. For example, run-length coding uses only similarity between pixels horizontally (horizontal coherence), and not vertically. Since every image is actually a 2-dimensional array of pixels, it can be viewed as a digital signal and methods for processing digital signals can be applied, including Fourier transform as an essential tool for signal processing.

3.1 Basic idea

JPEG compression is a very powerful lossy compression technique. It often can reduce the file size by factor of twenty or even fifty without apparent loss in image quality to the human eye. This is a huge saving since the lossless compression can achieve saving of memory space of barely 50%. JPEG compression is based on discrete cosine transform (DCT). DCT and inverse DCT used in decompression by itself do not induce any losses in the data, besides the fact that integer DCT does rounding of real value to the nearest integer.

JPEG uses the characteristic of the human eye that it sees less difference in the shades of colors than in the light intensity. Photos in full color achieve best compression ratios. Lower, but still good, is compression of images in shades of gray. Lowest compression is archived for simple drawings with sharp edges and the text, where even at low compression visible defects appear. User can choose the level of compression. For the higher level of compression files will be smaller but damaging of images becomes more pronounced. The main problem of JPEG compression algorithm is the fact that the damage resulting from the compression is permanent, and with successive editing of images, the losses accumulate.

3.2 Block preparation

The first step in JPEG compression of an image is block preparation. Different color models are used for image representation. Most often used is RGB model where each color is represented as a combination of three basic colors - red, green and blue. It corresponds naturally to the human eye characteristics since eye has receptors for these three colors. However, other color formats may be more appropriate for image processing. One of the most interesting is YCbCr system, originally developed for television. One intensity (Y) and two hue (chroma) components (Cb and Cr) represent each pixel, and each value is in the range 0 to 255. Eq. 1 describes RGB-to-YCbCr conversion:

$$\begin{aligned} Y &= 0.29900*R + 0.58700*G + 0.11400*B \\ Cb &= -0.16874*R - 0.33126*G + 0.50000*B + 128 \\ Cr &= 0.50000*R - 0.41869*G - 0.08131*B + 128 \end{aligned} \tag{1}$$

Since human eye is much more sensitive to light intensity then color information, square blocks of four pixels are averaged in Cb and Cr matrices and less bits are used for Cb and Cr components coding; ratio of Y:Cb:Cr bits is 2:1:1. This down-sampling of two chroma channels already reduces the size of image file by 50%. This transformation is lossy, but unnoticeable to the human eye.

In order to make transform faster and more efficient, the image is divided into square blocks of size 8x8 pixels. Each block has 64 image elements, i.e. light intensity values. The size of 8x8 for blocks was selected since at the time when the JPEG standard was established the size of 16x16 or more was not possible because of excessive hardware requirements. Computation time increases with the size of the block and 8x8 blocks give satisfactory results. If the image is divided into smaller blocks, for example, with a 2x2 or 4x4 picture elements, then the DCT would be less effective in compression.

3.3 Discrete cosine transform

After the division of the image into 8x8 blocks two-dimensional discrete cosine transform (DCT) (which is similar to Fourier transform but more appropriate for this application) is applied to each block. The definition of the DCT transformation (limited to matrix 8x8) is described by Eq. 2, and inverse transform (IDCT) by Eq. 3:

$$D(i,j) = \frac{C(\mathrm{i})*C(j)}{4}\left[\sum_{x=0}^{7}\sum_{y=0}^{7} d(x,y)*\cos\frac{(2x+1)*\mathrm{i}*\pi}{16}*\cos\frac{(2y+1)*j*\pi}{16}\right] \tag{2}$$

$$d(x,y) = \frac{C(x)*C(y)}{4}\left[\sum_{i=0}^{7}\sum_{j=0}^{7} D(i,j)*\cos\frac{(2x+1)*\mathrm{x}*\pi}{16}*\cos\frac{(2y+1)*y*\pi}{16}\right] \tag{3}$$

where C(u) are constants:

$$C(u) = \frac{1}{\sqrt{2}} \text{ for } u=0, \text{ and } C(u) = 1 \text{ for } u>0.$$

Direct discrete cosine transform on the input received signal intensity in 64 points, and it outputs a series of 64 coefficients, which correspond to two-dimensional spatial frequency. The first factor corresponds to the average intensity values in block 8x8 samples and is known as the DC component, while the remaining 63 coefficients are called AC components. DC coefficient contains most of the picture information and is most important in the image reconstruction. AC coefficients, which are located near the DC coefficient, correspond to lower spatial frequencies, and the AC coefficients which are located towards the lower-right corner correspond to the high spatial frequencies. As the intensity of spatial signal slowly change, most of the higher frequency AC components have value close to zero.

Matrix form of Eq. 2 is obtained from:

$$T_{i,j}=\begin{cases} \dfrac{1}{\sqrt{N}}, \text{if i = 0} \\ \sqrt{\dfrac{2}{N}}\cos\left[\dfrac{(2j+1)i\pi}{2N}\right] \quad , \text{if i > 0} \end{cases} \tag{4}$$

For a block 8x8 it results in this matrix T:

T =

.3536	.3536	.3536	.3536	.3536	.3536	.3536	.3536
.4904	.4157	.2778	.0975	-.0975	-.2778	-.4157	-.4904
.4619	.1913	-.1913	-.4619	-.4619	-.1913	.1913	.4619
.4157	-.0975	-.4904	-.2778	.2778	.4904	.0975	-.4157
.3536	-.3536	-.3536	.3536	.3536	-.3536	-.3536	.3536
.2778	-.4904	.0975	.4175	-.4175	-.0975	.4904	-.2778
.1913	-.4619	.4619	-.1913	-.1913	.4619	-.4619	.1913
.0975	-.2778	.4157	-.4904	.4904	-.4157	.2778	-.0975

As expected from Eq. 4, the first row of the matrix has all coefficients equal to $\frac{1}{\sqrt{8}}$. The inverse of the matrix T is easy to calculate since T is orthogonal and its inverse is equal to transposed matrix T'.

Table 1 represents one block from the image "Lena.bmp" (Fig. 2) and, as usually, the changes between neighboring pixels are not large.

143	143	136	138	136	132	136	136
140	136	140	137	137	138	134	137
135	134	141	139	138	140	133	134
133	140	138	140	140	125	134	126
133	140	137	138	136	134	134	127
134	134	138	131	131	137	134	134
135	135	136	138	140	141	138	135
133	133	133	135	136	136	135	133

Table 1: 8x8 block of the original image

63	7	-8	1	-3	0	0	0
6	7	2	-2	2	0	0	0
3	-3	8	1	5	0	0	0
0	2	7	0	-1	0	-2	0
-3	6	-1	0	0	0	-9	0
4	0	-2	1	0	0	-3	0
-4	0	3	-1	0	0	2	0
2	0	-1	0	0	0	-1	-1

Table 2: DCT coefficients

Before the DCT can be applied, original intensity values between 0 and 255 have to be translated by -128, because the DCT is designed to work on pixel values ranging from -128 to 127. We will call this resulting matrix M. Now we perform the DCT on M, which is accomplished by matrix multiplication:

$$D=TMT'$$

After translation, DCT was applied and Table 2 represents frequency coefficients. In Table 2 we can see that DC component, 63, is rather large and few low frequency components are also significant, while in the lower-right corner high frequency components are close to zero. DCT did not cause any loss of information, inverse DCT applied to Table 2 should regenerate Table 1, again after translation by 128. Small errors would appear only as a result of rounding since only integers are used.

3.4 Zig-zag sequence

Quantization is the main step in the compression process but it will describe later in a separated section. After quantization procedure follows a zigzag encoding. 64 quantized DCT coefficients of each block are by zigzag order as in Fig. 1 arranged in one-dimensional array. The first element is always quantized DC coefficient, and then follows the quantized low frequency AC coefficients and finally high frequency AC coefficients which are mostly equal to zero. Since the DC coefficients of two adjacent 8x8 blocks are usually strongly correlated, they are encoded by difference between the current and previous DC coefficient.

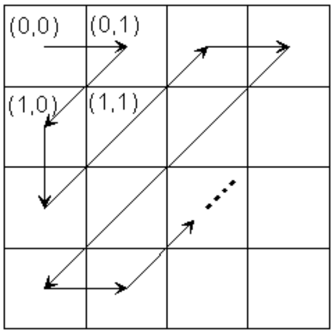


Figure 1. Zigzag order

3.5 Encoding

Entropy coding is the last step of the DCT-based JPEG compression. The move increases the compression ratio, without affecting the deterioration of the image. It is a more efficient encoding of quantized DCT coefficients based on their statistical characteristics. JPEG specifies two methods for entropy coding - Huffman's method and the arithmetic coding. The basic method of JPEG compression use Huffman's algorithm. The idea of Huffman's coding is that the numbers that appear often have shorter code, and those that appear less often have longer code. The same table that was used in the process of compression must be used in the process of decompression. Huffman's tables can be predefined and used for all images or can be calculated a priori from statistically significant picture.

In contrast to the Huffman's coding, arithmetic coding specified in the JPEG standard does not require additional tables, because it is able to adapt to a statistically significant picture in the encoding. Arithmetic coding gives a 5-10% better compression than Huffman's coding, but it is much more complex and not suitable for some implementations (e.g., fast hardware implementation).

3.6 Decoding

The first step of decompression is converting Huffman codes in the intermediate sequence. Then symbols from intermediate sequence are extended to one 64 element long array of DCT coefficients for each pixel 8x8 blocks. The next step is dequantization, where each DCT coefficient is multiplied with appropriate quantization step. Quantization step value is read from the quantization table. Dequantized DCT coefficient values are then compiled from the "zigzag" sequence to original sequence. The decoded and dequantized numbers and fed to IDCT, which will reconstruct image samples. These samples are similar to the original samples.

4 Quantization

Quantization is the main source of compression, but also a reason for information loss. Highly useful feature of the JPEG process is that varying levels of image compression and quality are obtainable through selection of quantization tables. As a result, the quality ratio can be tailored to suit different needs. Each of the coefficients in the 8x8 DCT matrix is divided by a weight taken from a quantization table and less important DCT coefficients are wiped out. If all the weights are 1, the transformation does not do any compression.

4.1 Threshold quantization

There are many ways to compress images by quantization of DCT coefficients. One way is to reject all coefficients that are smaller than a specified threshold. It is the simplest way of compression but appropriate for some applications. Our software allows selecting threshold and observing results. After DCT transformation 38% of frequency coefficients for tis image are already zero, with threshold 1 percent of zeros increases to 55%. Threshold 10 eliminates 91% of coefficients, but still with no difference visible to the human eye. As interesting examples we show images with threshold 15 (Fig. 3) and threshold 30 (Fig. 4) with 93 and 96 percent of eliminated coefficients respectively. In Fig. 3 we see very little difference from the original image, and only in some parts of the image, but the difference is more pronounced in Fig. 4. Since the increase in rejected elements is only 3% but with significant drop in quality between these two images it shows that threshold has to be carefully selected. After a certain point insignificant improvement in compression ratio can significantly degrade the quality of image.

Figure 2: Original image “Lena”

Figure 3: Threshold 15, 93% zeros

Figure 4: Threshold 30, 96% zeros

4.2 Fixed number of coefficients

Another type of compression can be called fixed number of coefficients. We can enter the desired number of low frequency components that form the upper left corner of different size: 1, 3, 6, 10 etc. Our software allows entering the number of layers that are to be preserved and observe the results. Extreme case is to keep only the first coefficient, DC component (Fig. 5), and to eliminate remaining 63 coefficients i.e. 98.5% of them. It will give extreme compression but at the cost of reducing linear resolution by factor of 8 since only average brightness for 8x8 blocks is used. For upper left triangle with 10 components (4 layers) the percent of eliminated coefficients is 84 and the resulting image is very close to the original image (Fig. 6).

Figure 5: Fixed number l, 98.5% zeros

Figure 6: Fixed number 10, 84% zeros

4.3 Quantization tables

Quantization table is a square 8x8 matrix whose elements are usually empirically determined. Values of elements in the table usually increase from upper left to lower right pert of the matrix. Each element in the matrix of frequency coefficients is divided by corresponding element of the quantization table.

4.3.1 JPEG recommended quantization table

Table 3 shows the JPEG recommended quantization table for brightness component, which is contained in the information annex of the JPEG standard. It is a quality level 50 quantization matrix which renders both high compression and excellent decompressed image quality. The user can to decide on quality levels different from 50, ranging from 1 to 100, where 1 gives the poorest image quality and highest compression, while 100 gives the best quality and lowest compression.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	130	99

Table 3: Quantization matrix Q_{50}

For compression levels higher than 50, the standard quantization matrix is multiplied by *(100-quality level)/50*. For a quality level less than 50, the standard quantization matrix is multiplied by *50/quality level*. For quality levels above 50 the elements of quantization table are proportionally decreased and by dividing by smaller numbers more AC coefficients will survive. Opposite is true for the case when quality level is less than 50.

Figure 7: Quality 15%, 95% of zeros

Figure 8: Quality 5%, 97% of zeros

4.3.1 Power of 2 quantization table

Table 4 shows another quantization table. Elements of the table are powers of 2. The first 4 frequency components stay the same, the next 5 will be divided by 2, and as we move towards the bottom right corner, coefficients will be divided by larger powers of 2. This quantization table facilitates fast computation since division by power of 2 can be implemented as shift-right machine instruction. Quantization table from Table 4 applied to image "Lena" produces the image shown in Fig. 9 Table 5 shows quantization table in which just first frequency components stay the same, the next 3 will be divided by 2, the next 5 will be divided by 4, etc. When we apply Table 5 to image "Lena", the result is image in Fig. 10 with reduced quality.

1	1	2	4	8	16	32	64
1	1	2	4	8	16	32	64
2	2	2	4	8	16	32	64
4	4	4	4	8	16	32	64
8	8	8	8	8	16	32	64
16	16	16	16	16	16	32	64
32	32	32	32	32	32	32	64
64	64	64	64	64	64	64	64

Table 4: Power of 2 quantization table 1

1	2	4	8	16	32	64	128
2	2	4	8	16	32	64	128
4	4	4	8	16	32	64	128
8	8	8	8	16	32	64	128
16	16	16	16	16	32	64	128
32	32	32	32	32	32	64	128
64	64	64	64	64	64	64	128
128	128	128	128	128	128	128	128

Table 5: Power of 2 quantization table 2

Figure 9: Image "Lena" after Table 4, 81% of zeros

Figure 10: Image "Lena" after Table 5, 87% of zeros

4.4 Special quantization tables

4.4.1 Small size images

The small size images may have different characteristics from the large images used in PC. Also, the photos with face have their own characteristics. The quantization tables for these images are derived from R-D optimization for the photos with face and it is shown that the proposed quantization tables have good performance for size and quality. For low bit rate coding of low resolution images in wireless handsets [3] the quality is done with two quantization tables Q_{75} (Table 6) and Q_{50}. The photos with face now serviced in wireless handset are searched. Some typical images are shown in Fig. 11. The quantization table design is based on R-D optimization and the quality control is also done with two quantization tables Q_{75} and Q_{50}.

Figure 11. Low resolution face images

R-D optimization is an algorithm to efficiently optimize rate-quality tradeoffs in an image specific way. However, to obtain a quantization table, R-D optimization is image-specific and it requires too much time. Even for the encoding on PC, it takes too much time to obtain quantization table. In R-D optimization, objective function is given as:

$$D(Q) + \lambda R(Q) \tag{5}$$

where λ is a Lagrange multiplier, D, R, Q denotes the distortion, rate, quantization table, respectively. The objective function (5) is minimized according to λ and quantization table Q. The R-D optimization can be summarized as follows:

1. Set $\lambda = 0$.
2. Increase the each component of quantization table and find the optimal Q which minimizes the R-D function (5).
3. Increase λ by δ and repeat Step 2.
4. Repeat Step 3 until λ reaches λ_{max}.

5. Plot R-D curve according to the variation of λ and find λ which is tangential to that curve with λ slope. At this time, the λ is the optimal λ and the quantization table is an optimal quantization table.

Each image is compressed with image quality 75, then quality 50. For each image, find Rmin75(Rmin50), Rmax75(Rmax50) and calculate (Rmin75 + Rmin50)/2 and find quantization tables Q75 (Q50) corresponding to (Rmin75 + Rmin50)/2. The final quantization table Q75 (Q50) for quality 75 (50) is an average of the quantization tables in last step.

8	6	5	8	12	20	26	31
6	6	7	10	13	29	30	28
7	7	8	12	20	29	35	28
7	9	11	15	26	44	40	31
9	11	19	28	34	55	52	39
12	18	28	32	41	52	57	46
25	32	39	44	52	61	60	51
36	46	48	49	56	50	52	50

Table 6: Quantization matrix Q_{75}

4.4.2 Biometric iris recognition

The JPEG still image compression standard allows use of custom quantization tables when image material with special properties is subject to compression. Example is iris imagery [4] which have different properties compared to common arbitrary images, and a pleasant viewing experience as being the aim in designing the default tables, might not deliver optimal matching results in the context of biometric recognition (e.g. sharp edges required for exact matching could appear appealing to human observers).

In Table 7 an increasing large number of high frequencies are suppressed by dividing the coefficients by 255, coefficients not affected are quantized as defined in the default Q-table. In the Table 8, only the 6 leading coefficients are quantized in the regular manner, the rest is severely quantized. The rationale behind the selection of these matrices is to investigate the importance of medium frequency information in iris recognition (high frequency information is assumed not to be useful in any case).

16	11	10	16	24	40	255	255
12	12	14	19	26	255	255	255
14	13	16	24	255	255	255	255
14	17	22	255	255	255	255	255
18	22	255	255	255	255	255	255
24	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

Table 7. Q_{12} for iris recognition

16	11	10	255	255	255	255	255
12	12	255	255	255	255	255	255
14	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

Table 8. Q_{16} for iris recognition

5 Conclusion

Software system is developed that facilitates, through user-friendly graphical interface, access to all internal parameters of the JPEG compression recommendations. Since quantization process is most important and critical step in JPEG image compression, besides changes in parameters, different ways of handling quantization are incorporated. There is a lot of research in the area of custom quantization tables that are appropriate for very specialized types of images and applications.

Further development of this software system will include more possibilities for new ways of generating custom quantization tables.

References

[1] Mitchell A. Golner, Wasfy B. Mikhael and Venkatesh Krishnan, *Modified JPEG Image Compression with Region-Dependent Quantization,* Circuits Systems Signal Processing, Vol. 21, No. 2, pp. 163-180, 2002.

[2] Gu-Min Jeong, Chughoon Kim, Hyun-Sik Ahn and Bong-Ju Ahn, *JPEG Quantization Table Design for Face Images and Its Application to Face Recognition*, IEICE Transaction Fundamentals, Vol. 89, No. 11, 2006.

[3] Gu-Min Jeong, Jun-Ho Kang, Yong-Su Mun and Doo-Hee Jungr, *JPEG Quantization Table Design for Photos with Face in Wireless Handset*, Springer-Verlag, Berlin Heidelberg Berlin, 2004.

[4] Mario Konrad, Herbert Stogner and Andreas Uhl, *Custom JPEG Quantization for Improved Iris Recognition Accuracy*, Springer-Verlag, Berlin Heidelberg, Vol. 5558, pp. 1091-1101, 2009.

[5] Jesse D. Kornblum, *Using JPEG quantization Tables to Identify Imagery Processed by Software*, Elsevier Digital investigation 5, pp. 21-25, 2008.

[6] Ching-Yang Wang, Shiuh-Ming Lee and Long-Wen Chang , *Designing JPEG quantization tables based on human visual system*, Signal Processing: Image Communication 16, pp. 501-506, 2001.

[7] Richard Szelinski, Computer Vision Algorithms and Applications, Springer-Verlag London, p. 812, 2011.

[8] Wilhelm Burger, Mark J. Burge, Principles of Digital Image Processing, Springer-Verlag London, p. 261, 2009.

[9] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Prentice Hall, p. 793, 2002.

Ana TUBA
University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: anatuba89@gmail.com

Marko Milosevic
University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: markoub@gmail.com

Prof. Milan Tuba
University Megatrend Belgrade
Faculty of Computer Science
Bulevar umetnosti 29, N. Belgrade
SERBIA
E-mail: tuba@ieee.org

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

An applied steganography study on the "Least Significant Bit" method

TUDORICĂ Gabriel, STÂNEA Paul
Coordinator Teachers: Lecturer Ph.D. Daniel Hunyadi

Abstract

The purpose of this project is to present the benefits of steganography and emphasize popular ways it can be applied to. We will focus on a digital steganography technique, using Bitmap files as carrier files for our hidden messages, thus hiding it in plain sight. Even though the picture can be seen by others, only the sender and the intended recipient will actually be able to get the messages.

Finally, we built an application using C# capable of applying this steganography technique, and allowing the user to embed hidden messages in 24 bit Bitmap files. Additionally, we implemented a chat-like environment using 24 bit Bitmap files to send the encrypted data.

1. Introduction

We built our application on the .NET Framework, using C# and it was designed to allow any user to embed hidden messages into 24 bit Bitmap files.

Using a substitution method called Least Significant Bit, this feature was made possible. This substitution method has a drawback though; namely the amount of text that can be hidden is limited to the size of the bitmap.

Basically, our application loads any 24 bit bitmap and using the Least Significant Bit method, it calculates the maximum number of bits (in characters) one can embed into the file. The substitution method replaces the last bit of every byte in the bitmap file with the bits from our binary transformed message. Clearly, our message must not exceed the maximum number of bits available for this process. Thus, we built a limiter which provides graphical feedback to the user.

After the message has been entered, for security reasons, the user must enter a password to protect the file for the decoding process.

After the password has been set, the application saves the new bitmap file with the embedded message and password and it can be sent to the intended recipient. The differences between the original file and the one with the embedded message are so small, that no one will notice unless they can compare it to the original.

In some cases, depending on the bitmap size and hidden message length, these differences are invisible.

For the decoding part, the recipient will load the file into the application, fill in the password field, and the hidden message will be revealed.

Since there are other applications capable of doing the same thing, we thought of a way to significantly improve ours. While most steganography applications use one technique to embed messages, we implemented another one, namely the generation technique.

Two out of three techniques have some serious drawbacks!

More precisely, the insertion method adds harmless bits to an executable file for example, increasing the carrier file size depending on the embedded message length, thus arousing suspicion.

The big drawback of the substitution method is that the maximum available size for our hidden message is determined by the size of our bitmap file, which can cause problems.

Since the two above mentioned methods are not 100% trustworthy, we used a third one, the generation method, which generates a large enough image to carry the entered message. This method requires the message to be entered first, then it calculates the size of the text in bits and finally generates a large enough 24 bit bitmap file capable of embedding the message. The resulting bitmap file size will be directly dependent to the message length and its width and height are chosen so that the resulting file is large enough to withhold the message and not too much larger then it is needed. Since this is directly linked to the message length, we won't have 2 images of the same width or height.

Furthermore, we added another great feature to our application using the generation method. We created a chat-like environment where two users can chat using secure hidden messages found in 24 bit bitmap files using a FTP server. Once the messages are created, a picture carrying the message will be generated and uploaded to a FTP server, where the other instance of the application scans the server for new images. When found, it downloads it locally, loads it, decodes it and displays the message. After the message is displayed, both the remote and local files are deleted. We did not protect the images with a password because we wanted to speed up the whole process and make it as close as a chat-like application as we could. And since the carrier files will stay for just about a few seconds on the server, the file does not present a security risk.

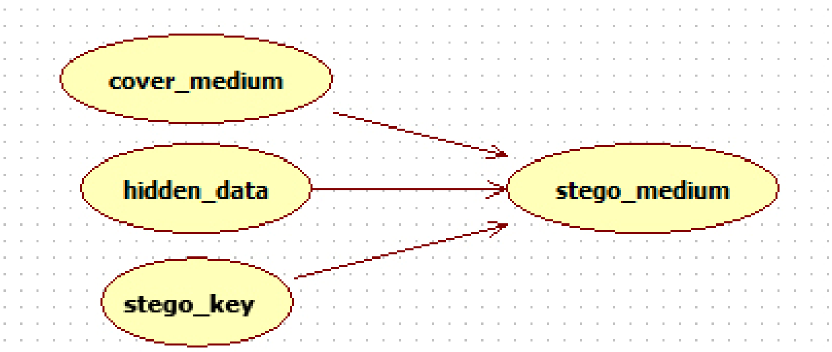
The main goal we managed to achieve here is that the whole conversation, viewed from outside, is camouflaged into a trivial file upload on a FTP Server, which is great!

2. Steganography

Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and the intended recipient, suspects the existence of the message. It's a form of security through obscurity.

The main advantage of using steganography over cryptography alone, is that the messages do not attract attention on themselves, furthermore the messages are double protected. [1][2][3]

Diagram (1) provides a very generic description of the pieces of the steganographic process:



(1)

Cover_medium is the file in which we will hide the hidden_data (the carrier file), which may also be encrypted using the stego_key. The resultant file is the stego_medium, which will be the same file type as the cover_medium and it will be perfectly usable. The cover_medium are typically image or sound files. Our application will focus on image files.[2][3]

History confirms what we already know: the best place to hide messages is in plain sight. An example of ancient steganography use is that of Histiaeus, who shaved the head of his most trusted slave and tattooed a message on it. After his hair had grown, the message was hidden. His purpose was to revolt against the Persians. [1][2]

Another useful purpose of steganography is the so-called *digital watermarking*. A watermark, historically, is the replication of an image, logo, or text on paper stock so that the source of the document can be at least partially authenticated. A digital watermark can accomplish the same function.

A graphic artist, for example, might post on a website sample images with an embedded signature (known only by him), so that he can later prove his ownership in case others attempt to portray his work as their own. [3]

2.1 Techniques

2.1.1 Physical steganography

Steganography has been and it is used today in many forms. The ways of hiding data nowadays are endless, but the most popular include:[4][5]

- Messages tattooed/written on messenger's body;
- Messages written in secret inks;
- Messages written below the postage stamps on envelopes.

2.1.2 Digital steganography

Steganography evolved in 1985 with the advent of the personal computer. Since then, there have been built over 800 steganography applications recognized by official institutions around the world. Some common digital steganography methods are: [4][5]

- Hiding messages in the lowest bits of noisy images or sound files;
- Hiding data in encrypted data or within random data;
- Adding harmless bits to executable files;
- Embedding photos in video material;

2.1.3 Network steganography

Network steganography uses communication protocols' control elements and their basic functionality, making the whole process even harder to detect or remove. An example of network steganography method is **Steganophony** - the concealment of messages in Voice-over-IP conversations, e.g. intentionally sending corrupted packets that the receiver would ignore by default; [4][5]

2.1.4 Printed steganography

Nowadays much of the steganography employed today is quite high-tech, but steganography's goal is to mask the existence of a message. A message can be concealed by traditional means and produce a ciphertext.

A popular and almost obvious method is called a null cipher. In this type of steganography, one would decode the hidden message by taking the first or other fixed letter from each word and create new words. Other forms include deliberately making mistakes, using different fonts or other hard to notice symbols. [4][5]

Consider this cablegram that might have been sent by a journalist/spy from the U.S. to Europe during World War I:

***P**RESIDENT'S **E**MBARGO **R**ULING **S**HOULD **H**AVE **I**MMEDIATE **N**OTICE. **G**RAVE **S**ITUATION **A**FFECTING **I**NTERNATIONAL **L**AW. **S**TATEMENT **F**ORESHADOWS **R**UIN **O**F **M**ANY **N**EUTRALS. **Y**ELLOW **J**OURNALS **U**NIFYING **N**ATIONAL **E**XCITEMENT **I**MMENSELY.*

The first letters of each word form the character string: *PERSHINGSAILSFROMNYJUNEI.* A little imagination and some spaces yields the real message: *PERSHING SAILS FROM NY JUNE I.*

2.2 Carriers

Steganography today, is allowing the user to hide large amounts of information within image and audio files. These forms of steganography often are used in conjunction with cryptography so that the information is doubly protected; first it is encrypted and then hidden so that an adversary has to first find the information (an often difficult) and then decrypt it.

We will focus on the image files, the BMP format to be more precise.

2.2.1 The BMP file format

The BMP File Format, also known as Bitmap Image File or Device Independent Bitmap (DIB) file format or simply a Bitmap, is a Raster graphics image file format used to store bitmap digital images, independently of the display device (such as a graphics adapter), especially on Microsoft Windows and OS/2 operating systems.

The BMP File Format is capable of storing 2D digital images of arbitrary width, height, and resolution, both monochrome and color, in various color depths, and optionally with data compression, alpha channels, and color profiles.

The Bitmap Image File consists of fixed-size structures (headers) as well as variable-size structures appearing in a predetermined sequence. Many different versions of some of these structures can appear in the file, due to the long evolution of this file format.[6]

2.2.2 BMP header

The BMP file begins with a 54 byte header with the following descriptions: [7]

Offset	Size	Description
0	2	signature, must be 4D42 hex
2	4	size of BMP file in bytes (unreliable)
6	2	reserved, must be zero
8	2	reserved, must be zero
10	4	offset to start of image data in bytes
14	4	size of BITMAPINFOHEADER structure, must be 40
18	4	image width in pixels
22	4	image height in pixels
26	2	number of planes in the image, must be 1
28	2	number of bits per pixel (1, 4, 8, or 24)
30	4	compression type (0=none, 1=RLE-8, 2=RLE-4)
34	4	size of image data in bytes (including padding)
38	4	horizontal resolution in pixels per meter (unreliable)
42	4	vertical resolution in pixels per meter (unreliable)
46	4	number of colors in image, or zero
50	4	number of important colors, or zero

3.3.3 BMP file size

The bits representing the bitmap pixels are packed in rows. The size of each row is rounded up to a multiple of 4 bytes (a 32-bit DWORD) by padding.
For images with Height > 1, multiple padded rows are stored consecutively, forming a Pixel Array.[6]

The total number of bytes necessary to store one row of pixels can be calculated using formula (2) below:

$$\text{Row Size} \approx \text{CEILING}\left(\frac{\text{ImageWidth} \bullet \text{BitsPerPixel}}{32}\right) \bullet 4 \qquad (2)$$

ImageWidth is expressed in pixels.

The total amount of bytes necessary to store an array of pixels in an n Bits per Pixel (bpp) image, can be calculated as follows:

$$PixelArraySize = RowSize * ImageHeight$$

ImageHeight is expressed in pixels.

$$FileSize \approx 54 + PixelArraySize$$

3. Algorithms and Techniques

There are three different techniques you can use to hide information in a cover file:

3.1 Injection (or insertion);
3.2 Substitution;
3.3 Generation.

3.1 Injection or insertion

Using this technique, the data is stored in section of a file that is ignored by the application that processes it. For example in unused header bits or adding harmless bytes to a file leaving it perfectly usable.

The more data you add, the larger the file gets, and this can raise suspicion. This is the main drawback of this method.[1][4]

3.2 Substitution

Using this approach, the least significant bits of information are replaced with desired data, which will be reproduced on decoding.

The main advantage of this method is that the file size does not change, but the file is affected by quality degradation, which in some cases may corrupt files and another flaw is that the available amount of data is limited by the number of insignificant bits in the cover file. [1][4]

3.3 Generation

Unlike injection and substitution, this technique doesn't require an existing file, it generates it just to embed the message, which is better than the other two mentioned methods because it's not being attached to another file to suspiciously increase the file size, it has no limitation and the result is an original file and therefore immune to comparison tests. [1][4]

4. The Least Significant Bit (LSB) method

The Least Significant Bit (LSB) method is the most common substitution technique, which basically replaces the least significant bytes in a file to conceal data, which is extracted at the decoding process. It's especially effective on lossless image files, such as 24 bit Bitmap files.

The method takes the binary representation of any form of data and overwrites the last bit of every byte in an image.

As an example, we will take a 3x1 pixel 24 bit bitmap, each pixel color is represented on one byte so our picture bitmap will have a total of 9 bytes. We have the following RGB encoding:

11010101 10001101 01001001
11010110 10001111 01001010
11011111 10010000 01001011

Now suppose we want hide the following 9 bits of data: 101101101. If we overlay these 9 bits over the original file using the LSB method, we end up with the following new RGB encoding, which visually is not too different. The bits in **bold** have been changed.

*11010101 1000110**0** 01001001*

*1101011**1** 1000111**0** 0100101**1***

11011111 10010000 01001011

We managed to hide 9 bits of information by only changing 4 bits or about 50% of the lest significant bits.[2]

Similar methods can be applied to lower color depth image files, but the changes would be too dramatic and therefore obvious. On the other hand, grayscale images provide an excellent cover file for the LSB method.

It is worth mentioning that *steganalysis* is the art of detecting and breaking steganography. A form o analysis is to carefully examine the color palette of an image, which very often has a unique binary encoding for each individual color. If steganography is used, the same color will end up having more binary encodings, which would suggest the use of steganography in the specific file.

But if we wanted to check for hidden information in images, which files should we analyze? Suppose I decide to send a message hidden in a picture that I will post on a website, where others can see it, but only certain people know that it contains hidden information. I can also post more images and only some of them would have hidden data so the potential stegananalyst would get confused. [2][5]

The quantity of potential cover files make steganalysis a Herculean task, and we will exploit this very spot of steganalysis in our application.

5. Steganography software

There are a number of software packages that perform steganography on just about any software platform. Some of the better known packages for the Windows platform include:

- Hide4PGP - Hide4PGP is a freeware program and its purpose is to hide any data in a way that the viewer or listener does not recognize any difference. Hide4PGP can use the following file formats: BMP, WAV, VOC.
  Available at: http://www.heinz-repp.onlinehome.de/Hide4PGP.htm;
- MP3Stego - MP3Stego will hide information in MP3 files during the compression process. The data is first compressed, encrypted and then hidden in the MP3 bit stream. Although MP3Stego has been written with steganographic applications in mind it might be used as a copyright marking system for MP3 files (weak but still much better than the MPEG copyright flag defined by the standard)
  Avilable at: http://www.cl.cam.ac.uk/~fapp2/steganography/mp3stego;

6. MessageHider Application

MessageHider is the application we developed using .NET Framework and C#. Compared with other software products that use only one steganography technique, our application uses two: substitution and generation. Our application has two modes: normal and chat mode. In the following, we will present the application in detail.

6.1 Normal mode

The normal mode, provides the basic functionality, using a substitution algorithm, more precisely the Least Significant Bit method. We will use 24 bit bitmaps as stego files, because of their lossless compression. Normal mode provides features for encoding and decoding hidden messages.

6.1.1 Encoding

To encode a hidden message, the user needs to first select the source file (24 bit bitmap). Once the bitmap is loaded, our application will estimate the maximum number of characters the user can hide in the selected file. As we have seen, the Least Significant Bit method, replaces the last bit of every byte in the file, with the bits for our hidden message. To get the maximum number of characters a user can hide we will use the following formula:

$$\text{AvailableCharacters} \approx \text{CEILING}\left(\frac{1}{16}\left(\frac{\text{ImageWidth}\cdot\text{BitsPerPixel}}{8}\,\text{ImageHeight} + \text{BmpHeader}\right)\right) \quad (3)$$

ImageWith, ImageHeight are expressed in pixels

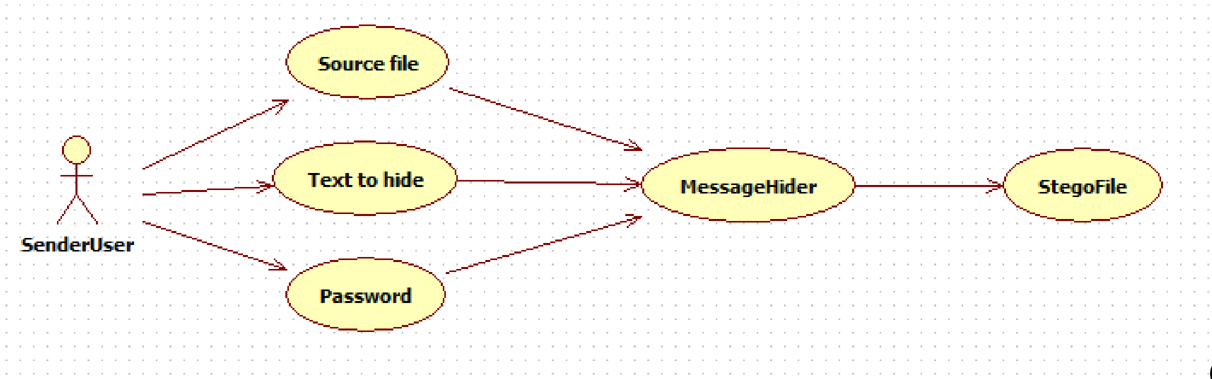
In other words, for a 50x50 bmp image we will have

$$\text{AvailableCharacters} \approx \text{CEILING}\left(\frac{1}{16}\left(\frac{50\cdot 24}{8}\,50 + 54\right)\right) \approx 473 \text{ characters} \quad (4)$$

The user will have a visual feedback as the message is being typed and will also limit the maximum number of characters he enters to the maximum available characters calculated before minus 32. The 32 characters we subtracted are reserved for the password, which can be by up to 30 characters, the two characters left being used as a delimiter from the original message on which the password will be concatenated.

After the message was entered and the password set, the Least Significant method will be called and the stego file will be generated.

The whole process can be visualized using diagram (5):



(5)

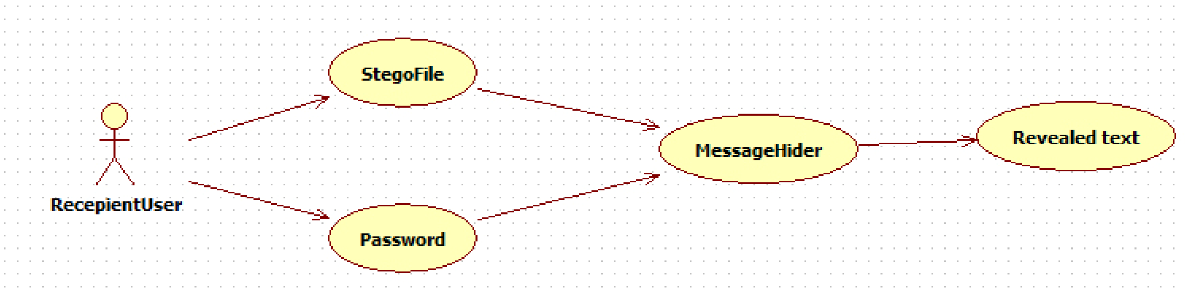
We can conclude that diagram (5), is the extended version of the diagram (1) we presented earlier.

6.1.2 Decoding

Firstly, the user will load the stego file into the application. Once this is done, the Least Significant Bit method will extract the last bit from every byte of the stego file and thus recreate the original binary message. After a conversion from byte to string, we get the original message. The message also has the password concatenated and delimited by the || characters.

The password is extracted from the string and stored for it to be checked in the next step, where the user is required to fill in the password field. If the password is correct, the original message sent will be revealed to the user.

The whole process can be visualized using diagram (6):



(6)

As diagram (6) shows, the decoding process is the inverse of the encoding process.

6.2 Chat mode

Chat mode uses the generation steganography technique. Basically, the user first types the message he wishes to embed, and a large enough picture capable to embed the message length will be generated as a stegofile.

Once in chat mode, the user will have to first assign a nickname which will be used to differentiate the message sender. Once the user fills in the nickname field, he will fill in the credentials in order to connect to the FTP Server and lastly press the Connect button. A visual feedback will notify the user about the connection status.

Once connected to the FTP Server, the user can begin to exchange messages with the other user. Using the message textbox, the user can type in the message he wishes to send. When he presses the send button, a couple of processes are initiating.

Firstly, we need to generate a picture large enough for our message to fit in. The width of the image will be generated by a random number between 100-200 pixels, all we have to calculate now is the height. Based on (3), we conclude that the ImageHeight is:

$$\text{ImageHeight} \approx \text{CEILING}\left(128 \, \frac{\text{MessageLength}}{\text{ImageWidth} \bullet \text{BitsPerPixel}} \right) \tag{7}$$

Now that we have the ImageWidth and ImageHeight, we can generate the carrier image. We just added noise to the image in order not to leave it blank, but random graphics can be generated or even fractals. Actually we have this in mind for future developments.

Once the image is generated, we apply the Least Significant Bit method, previously described in the normal mode – encoding part. After the message is embedded, we upload the bitmap file to the FTP Server and rename it as [nickname—unixTimestamp.bmp].

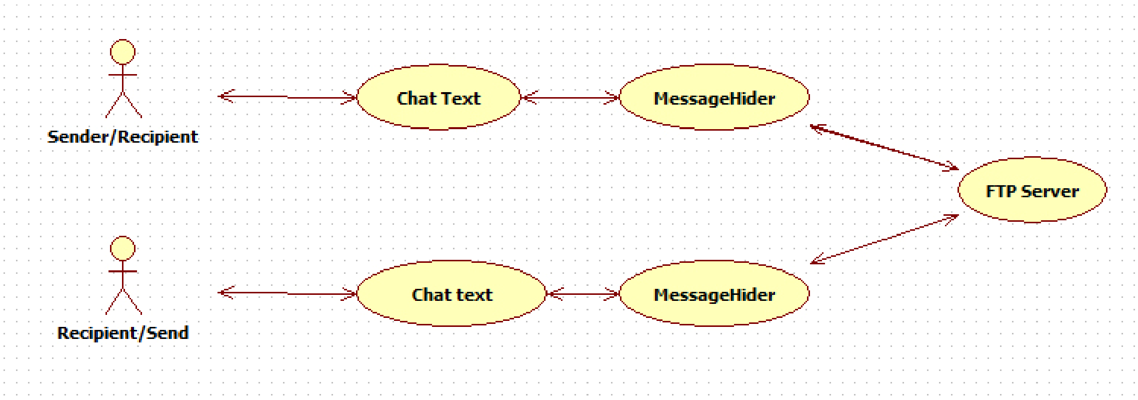
By using this naming convention we can easily know the message sender, and sort the files/messages in chronological order because of the unixTimestamp. Once the image has been uploaded, it is deleted from the local path.

The message we embedded in the file will also be displayed in the chat window, having the user’s nickname as a prefix.

The FTP connection is kept alive as long as the application runs in chat mode. And it will refresh the server directory every 10 seconds to search for new files/messages that do not belong to the current user. When a new file is found, the application downloads it locally, applies the Least Significant Bit method and extracts the message, which will then display in the chat window, having as a prefix the nickname of the remote user.

Once the file is decoded, both the remote and local files are deleted. When the user closes the chat mode, the FTP connection will close as well.

Visually the whole process can be visualized using diagram (8):



(8)

As diagram (8) shows, our application in chat mode is similar to a native chat application, only instead of using sockets and a chat server, our application uploads stego files to a FTP server.

7. Conclusion

The application provides a secure way to communicate with one or more users in real time through chat mode or in normal mode where the stego files can be sent via mail or uploaded on a website, thus hiding the message in plain sight which goes right to the heart of steganography's purpose.

Of course, there is always room for improvements. For future implementations we thought of encrypting the message using strong cryptography algorithms and embed the ciphertext in the bmp files, thus creating a double protection. Another feature we thought of, was to implement methods capable of embedding the messages in other file types (other image types or even sound or video files).

8. References

[1] http://en.wikipedia.org/wiki/Steganography
[2] http://www.garykessler.net/library/steganography.html
[3] http://www.garykessler.net/library/fsc_stego.html
[4] http://www.infosyssec.com/infosyssec/Steganography/techniques.htm
[5] http://en.wikipedia.org/wiki/Least_significant_bit
[6] http://en.wikipedia.org/wiki/BMP_file_format
[7] http://www.fastgraph.com/help/bmp_header_format.html

TUDORICĂ Gabriel
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Sibiu, Dr. Ioan Rațiu St. No. 5 - 7
ROMÂNIA
E-mail: office@eyenetworks.ro

STÂNEA Paul
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Sibiu, Dr. Ioan Rațiu St. No. 5 - 7
ROMÂNIA
E-mail: psb77black@yahoo.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Histogram manipulations as a pre-step to image recognition

Igor Valjevic, Ana Tuba
Teacher Coordinator: Prof. dr Milan Tuba

Abstract

Histogram graphs allow examination of image pixel color and intensity distribution at a glance. Speed at which histogram can be checked by human eye translates well into computer programs that allow very fast (usually linear) algorithms to enhance image. Various types of histogram matching adjust different types of images for different types of processing. In this paper we discuss several different types of histograms and histogram matching techniques in effort to improve overall efficiency of image recognition algorithms. Software system with appropriate graphical user interface has been developed to test and compare effects of different histogram matching techniques.

Keywords: Digital image, Image processing, Image histogram, Histogram matching

1 Introduction

Image histograms come in several variations depending on what type of data is observed. Some of more popular ones are intensity (brightness, luminosity) histograms, RGB composite histograms and histograms for separated RGB color components [1]. Most histogram manipulations are based on intensity histograms and as such work best on grey-scale images, but can be generalized to color images in one of the HSI formats.

When images are taken, requirements such as high contrast, even color saturation etc. that are essential for image recognition are often not taken into consideration. This is due to limitations of technologies used to obtain said images (X-Ray, Radio, Infra-red images etc.) or the fact that they were not originally intended to be used for automatic software recognition. For these reasons there is usually a need to adjust images to make process of recognition easier and more precise [2]. Unfortunately, there is no universal technique or algorithm that will work well with all types of images and it is usually based on trial and error to find the optimal algorithm that will work on given set of images for particular purpose.

Histograms are used to extract optimal threshold for segmentation in some algorithms such as image binarization that segments image in the foreground and background segments [3]. Based on this information it is possible to discern points of interest in the image. Histogram based image segmentation found its application in traffic control applied to license plate recognition that is both faster and more precise compared to some other methods [4]. Some systems go even step further developing image segmentation and recognition techniques based on histogram comparison. Those systems take full benefit from speed that histogram comparison offers and are used in system that demand high level of responsiveness, even in real time [5].

This paper is organized in three distinct segments. In Section 2 we discuss different color models with special emphasis on how to calculate intensity in each. Following that, in Section 3, software platform developed for testing purposes is described. Lastly, in Section 4 we focus on explanation of different histogram equalization and matching techniques and how they can help prepare image for better image recognition process.

Mentioned software platform has been solely developed by the author and offers wide variety of different image processing algorithms to test and compare to each other. As comprehensive platform it offers very easy way to add new algorithms, and test them against already implemented ones.

2 Intensity in different color models

For representation of color in computer systems abstract mathematical models called color models are used. Color models describe how colors can be represented as tuples of numbers. In different color models intensity is calculated differently and intensity value in one model does not exactly correspond to the same intensity value in other models [6].

2.1 RGB color model

RGB is an additive color model in which three base colors (Red, Green and Blue) are added together in different proportions to generate all other colors. It is based on human perception of colors with three different types of receptors in the eye and as such it is the most logical and easy to understand color model. So it comes as no surprise that it is the most widely used one. RGB model is often represented as a cube (Fig. 1).

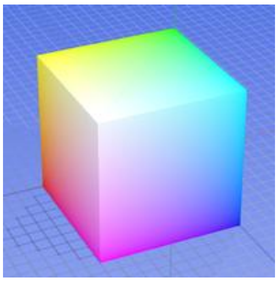


Figure 1. RGB color model as a cube

There are different models developed for calculating brightness in the RGB model. Most basic one is a simple average of all three components:

$$Y = (R + B + G) / 3 \qquad (1)$$

Although this formula works very well in a sense that it can transform color image into grey-scale, it has been observed that some shades of gray do not necessarily correspond to our expectations. Other formulas for calculating brightness have been developed:

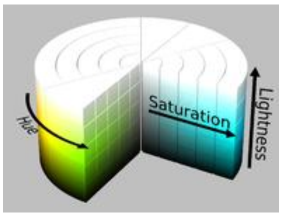
$$\begin{aligned} Y &= 0.2126 * R + 0.7152 * G + 0.0722 * B \\ Y &= 0.299 * R + 0.587 * G + 0.114 * B \qquad (2) \\ Y &= (R + R + B + G + G + G) / 6 \end{aligned}$$

These formulas take into account human perception of colors and are empirically determined.

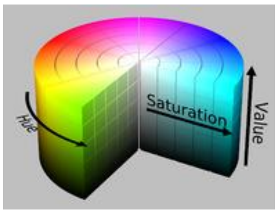
2.2 HSI (HSV, HSL) color model

While RGB model is usually represented by orthogonal coordinates, these models are represented in cylindrical coordinates. HIS and similar models transform original RGB model to a model closer to human visual perception. *Hue, Saturation, Intensity (Value, Luminosity*) are represented as tuples and are isomorphic with RGB model, meaning that one value in RGB model corresponds to exactly one value in HSI model and vice versa [7].

All these models are cylindrical shapes with their orthogonal angle describing *Hue*, red starting at 0^{o}, green at 120^{o} and blue at 240^{o}. Central axis is reserved for shades of gray with white and black on opposite poles of brightness. Distance from axis to the edge of figure determines *Saturation* with most saturated colors at the edges (Fig. 2). Difference between the two is that pure colors in HSL model are at lightness value of 1/2 and at HSV model they are at value 1. HSL model is closer to something we might have expected - that only white color is at lightness 0.



(a)



(b)

Figure 2. Cylindrical representation of HSL color model (a) and HSV color model (b)

Intensity values can be easily obtained in these models just by reading *Lightness* or *Value* but it should be noted that they do not correspond to each other or to RGB model precisely - for the same color in different models the intensity value will not be exactly the same.

3 Software platform

For experimenting with different histogram manipulation algorithms a tool is necessary that will allow easy change of the used algorithm and its parameters as well as examination of the results of applied image enhancement techniques. As a result of this requirement a program was developed in C# with simple and user-friendly graphical interface. From the start it was decided that it does not need to support many different formats as we will focus on 24 bits RGB format images with optional transformation to HSV, implemented as part of this software.

For testing purposes regular algorithms for *Brightness*, *Contrast* and *Color* component adjustment have been implemented. Different transformations from color image to grey-scale model have been implemented which allow to test impact of different color models on resulting image, before and after applying histogram matching techniques.

Histogram matching algorithm has been implemented with user interface that allows freehand drawing of target histogram, manual coefficient insertion or alteration and import from external textual file. Several standard target histograms come with program by default, but option to save any custom made target histogram is offered as well.

Aside from grey-scale histogram matching algorithm, algorithm that works on color images is offered as well. It is implemented by transforming image from RGB model to HSV and then applying histogram matching technique on *Value* component. All images presented in this paper have been processed by this application.

4 Image histogram

Image histogram is defined as a graph that on horizontal axis has different tonal values and on vertical axis the number of pixels that have that tonal value. With popularization of digital photography it became widely used because it can tell, at a glance, if image taken is overexposed or underexposed and its intensity distribution.

(a) (b)

Figure 3. Image (a) and its corresponding intensity histogram (b)

As we can see in Fig. 3.(b) there are not too many pixels close to 0 or 255 values implying that exposure when photography was taken was correct and no pixel was blown out by overexposure or blacked out by underexposure.

In the field of image processing, histograms are often used as guides when choosing values for tresholding, edge detection, image segmentation etc. Some techniques that offer adjustment of images rely heavily on histogram manipulations and offer much greater degree of control over resulting image. One of those techniques is histogram matching and its special case: histogram equalization.

4.1 Histogram equalization

Histogram equalization is a technique that is mainly used to adjust image contrast. It is used when picture is underexposed or overexposed and when better distribution of pixel intensities over whole available range is desired. Histogram equalization accomplishes this by effectively spreading the most frequent intensity values.

Histogram equalization for a given picture is calculate in three steps:

1) Calculate image histogram. As explained in Section 4, image brightness histogram is calculated by counting number of pixels that have the same intensity value, and aligning them in an ascending array.

2) Calculate new intensity value for each intensity level. New intensity value is calculated by applying formula:

$$O_i = \left[\sum_{j=0}^{i} N_j \right] * \frac{No.\,of\ \ Intensity\ Values - 1}{No.\,of\ \ Pixels} \qquad (3)$$

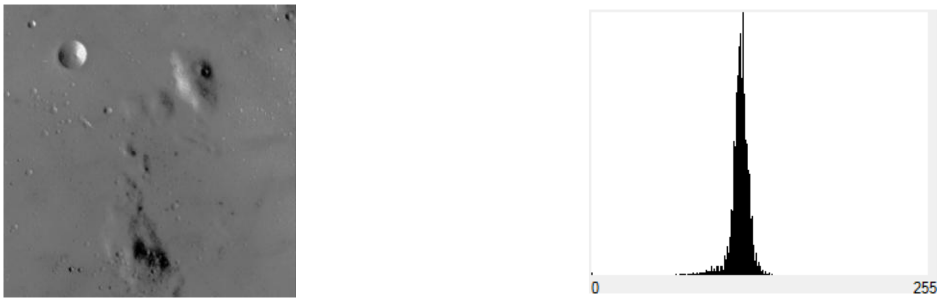
The expression is the bracket is the number of pixels having the intensity below the output intensity level or equal to it.

3) Replace old intensity value with the new one for every pixel where O_i is the new intensity value for pixels that had old intensity value i.

This algorithm can be expressed in pseudo code:

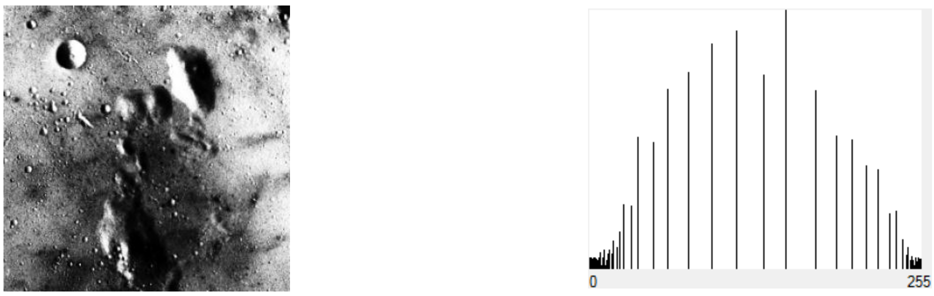
```
create histogram for given image
for each intensity level
        calculate new intensity
        add it to lookup table
for each pixel
        set its new value from lookup table based on old value
```

A key advantage of this technique is that it is both, simple to implement and invertible. Downside is that it can emphasize background noise. Histogram equalization quite often produces unlifelike images, but that is secondary to making it easier for computer to recognize, segment or in other ways process the image. It found large use in microbiology, medicine, astronomy, etc. Example of picture and its equalized counterpart can be seen on Fig. 4 and Fig. 5.



(a) (b)

Figure 4. Original image (a) with its corresponding histogram (b)



(a) (b)

Figure 5. Same image equalized (a) with its corresponding histogram (b)

Histogram equalization techniques can change average brightness value of image, often exhibited by oversaturating the resulting image. In some applications, such as television it is imperative to preserve original brightness and to avoid generation of nonexistent artifacts, methods that deal with such problems had to be created [8].

4.2 Histogram matching

Histogram equalization has limitation that it can produce only one result for specific image: an image with evenly distributed image histogram. Sometimes that is not enough, sometimes we want to control the shape of resulting image's histogram. We might want to do that because we want to emphasize certain intensity range or mask the other. Technique that let us set arbitrary histogram as target histogram is called histogram matching.

In its general form histogram matching algorithm can use arbitrary histogram as a target. We will present several images with its corresponding histograms and results of matching to different target histograms.

Sample image with its histogram can be observed on Fig. 6.

(a) (b)

Figure 6. Original image (a) and its corresponding histogram (b)

Regular algorithm to increase brightness just adds some constant to pixels intensity values, usually generating large patches of white color (overblown areas). With histogram matching we have much greater degree of freedom to evenly distribute intensity to produce image that looks more naturally brighter. On Fig. 7 we can see target histogram and resulting image with its histogram.

(a) (b) (c)

Figure 7. Brightness increasing target histogram (a), resulting image (b) and its histogram (c)

Similarly we can increase darkness without creating large patches of black areas, as seen on Fig. 8. This technique is usually applied to CAT scans to accent parts that are of particular interest to the researchers.

(a) (b) (c)

Figure 8. Brightness decreasing target histogram (a), resulting image (b) and its histogram (c)

Sometimes it is needed to reduce contrast. For example, with edge detection techniques we can get too many false edges, usually due to white noise. In addition to using filters to blur the image, we can try to reduce this noise by reducing contrast with histogram matching as in Fig. 9.

(a) (b) (c)

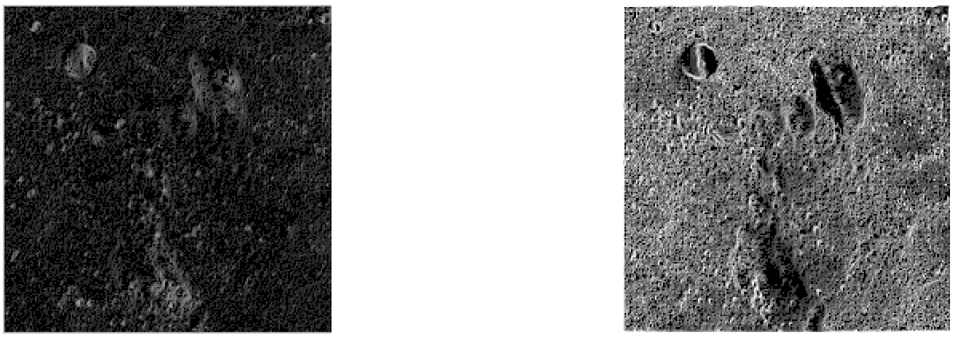
Figure 9. Gaussian target histogram (a), resulting image (b) and its histogram (c)

Finally, to reinforce that histogram equalization is a special case of histogram matching we present target histogram that produces same result as histogram equalization algorithm. That histogram matching is shown in Fig. 10.

(a) (b) (c)

Figure 10. Equalizing target histogram (a), resulting image (b) and its histogram (c)

To illustrate what difference histogram matching can pose with image recognition on Fig. 11 two images are presented. We applied *Edge detection* algorithm on image from Fig. 4 and resulting image is presented on Fig. 11(a). Same image after *Histogram equalization* has been applied on it, as seen on Fig. 5, produced much clearer result with the same *Edge detect* algorithm Fig. 11(b).



(a) (b)

Figure 11. Example of Edge detect algorithm applied to image (a) and to image that was Equalized first (b)

4.3 Histogram matching with color images

So far we used histogram matching algorithm on grey-scale images only. As mentioned before, it is possible to apply the same method to color images. Easiest way to achieve this is to transform image to one of the HSI models and apply histogram matching techniques to intensity component while leaving hue and saturation components unchanged. Example of color image and its equalized counterpart can be seen in Fig. 12.

(a) (b)

Figure 12. Color image (a) and same image after Histogram equalization (b)

When working with color images special care is needed to assure that resulting image is not unexpectedly oversaturated. As mentioned before [4] one of the ways to reduce this unwanted result is by calculating separate histograms for smaller portions of the image and combining them.

5 Conclusion

Algorithms to enhance image based on histogram manipulations offer much greater, nonlinear degree of control over resulting image compared to some other techniques. While histogram equalization is most often used, other histogram matching techniques have their significant place as preparation for image recognition algorithms. Unfortunately, with such great degree of freedom comes a setback in a form of nonexistence of single target histogram that would improve all images and make them easily recognizable by computer software. Even more, there is no algorithm that will automatically produce such histogram and that is why in most cases developing of appropriate histogram is still very much by trial and error.

Future goal is to implement fully functional image recognition software based on this platform that would allow different image recognition techniques to be compared to each other. In the work are edge detection and segmentation modules, as well as database that will be used for machine learning algorithms.

References

[1] L. Lu, Y.C. Zhou, K. Panetta, S. Agaian, Comparative Study of Histogram Equalization Algorithms for Image Enhancement, *Mobile multimedia/image processing, security, and applications 2010*, *Proceedings of SPIE-The International Society for Optical Engineering*, Vol. 7708, 2010

[2] T. Acharya, A. Ray, *Image Processing: Principles and Applications, Wiley-Interscience,* 2005, p. 425

[3] M.F. Hossain, M.R. Alsharif, K. Yamashita, An approach to color image enhancement using minimum mean brightness error dynamic histogram equalization*, International journal of innovative computing information and control*, Vol. 7, Issue: 2, pp. 827-840, 2011

[4] L.J. Guo, X.A. Tang, Y.B. Liu, Character Recognition Based on 2-D Histogram Image Threshold Segmentation*, Advanced Materials Research, Progress in measurement and testing, pts. 1 and 2*, Vol. 108-111, pp. 1366-1369, 2010

[5] H. Bang, D. Yu, S. Lee, I.H. Suh, An Object Recognition System Based on Color Co-occurrence Histogram and Geometric Relations of Pyramidal Image Patches*, International conference on robots and intelligent systems*, Vol. 1-3, pp. 3591-3596, 2008

[6] R. Gonzalez, R. Woods, *Digital Image Processing, Prentice Hall; 2nd edition*, p. 793

[7] J. Russ, *The Image Processing Handbook: Fourth Edition,* 2002, p.732

[8] T. Kim, J. Paik, Soft decision histogram-based image binarization for enhanced ID recognition*, 6th international conference on information, communications & signal processing*, Vol. 1-4, pp. 717-720, 2007.

Igor Valjevic
University of Belgrade
Faculty of Mathematics
Studentski trg 16, 11000 Belgrade
SERBIA
E-mail: igor.valjevic@gmail.com

Ana Tuba
University of Belgrade
Faculty of Mathematics
Studentski trg 16, 11000 Belgrade
SERBIA
E-mail: anatuba89@gmail.com

Prof. Milan Tuba
University Megatrend Belgrade
Faculty of Computer Science
Bulevar umetnosti 29, N. Belgrade
SERBIA
E-mail: tuba@ieee.org

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

DESIGNING AND CONSTRUCTING A ROBOTIC ARM

Ivan Vasilov
Teacher coordinator: Milena Lazarova

Abstract

The increased use of systems based on programming and robotics in the industry is motivated by the new possibilities that the relationship between these two technologies open for controlled robot systems. The paper presents the design and construction of a robotic arm with five rotational movements. The robot uses servomotors and a servo controller. For solving the inverse kinematics problem an algorithms based on cyclic coordinate descent is applied and a program written in C# is used to control the robot arm. A prototype of the robotic arm is built and the experiments made showed the suggested solution provides satisfactory results of the movement and control of the robotic arm.

1 Introduction

One respectable manufacturer of ABS sensors recalls problems due to the high level of failure. The main reason for the failure of the sensors was cables that were damaged during the installation process. The main cause turned out to be the fatigue of workers who were placing the wires in the body of the sensor. The company decided to install a Cognex robot. He was able to perform the task by using three-dimensional system and machine vision. The percentage of failure was reduced to zero and the system currently works three shifts per day without rest and interruption. [1]

The above is an example of how robotics helped to perform a task better than a human. The solution removes a boring and repetitive task, which often leads to unpleasant working environment.

Since the beginning of civilization people were interested in finding a way to ease their work. Starting with the various tools, moving into more advanced machines, the ability to make work easier and shorter has always been one of the greatest aspirations of humanity. The result of this desire to minimize work is the robot.

By definition, the robot is a machine (self-guided or led by a human) that can perform various tasks [2]. It's not uncommon for the robot to complete these tasks better, faster and more accurately than the human himself.

In the beginning, the robots were been used in specialized tasks, but then robotics began developing rapidly. Robots have changed today are used in all types of industry. This is due to the development of computers that made it possible to manage robots with several degrees of freedom [3]. Robots often imitate human movements.

The paper presents the design of a robotic arm that is controlled by a computer. It has five rotational movements and uses servomotors and a servo controller. For solving the inverse kinematics problem an algorithms based on cyclic coordinate descent is suggested and implemented as a program written in C# used to control the robot arm. A prototype of the robotic arm is built and some experiments are made that proven the suggested solution provides satisfactory results of the movement and control of the robotic arm.

2 DIY Robotic arm

2.1 Design and construction of the robotic arm prototype

The robotic arm has five degrees of freedom and a pincer for an end effector. All five movements are rotational. The simplified mechanical model is shown in Figure 1.

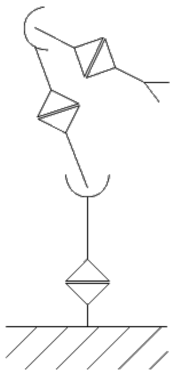


Figure 1.A Simplified mechanical model

Coordinate systems of various joints and the pincer are shown in Figure 2. The first, third and fifth joint are rotating around the axis Z, while the second and fourth join are rotating around the axis Y, as shown in Figure 5. The global coordinate system corresponds to the local coordinate system of the first joint.

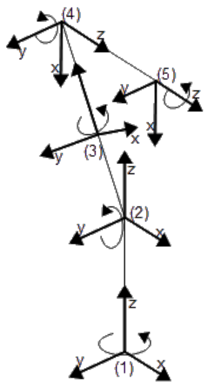


Figure 2. The coordinate systems for all joints

The whole robot is composed of 6 modules that are assembled from two or more parts. The modules correspond to the five degrees of freedom and pincer-like end effector.
Each piece is cut out of Plexiglas, using CO_2 laser CNC machine.
Polymethyl methacrylate (PMMA) or Plexiglas is known as transparent thermoplastics, which is often used as an alternative to glass. Chemically, it is a synthetic polymer of methyl methacrylate. The material was developed in 1928 in various laboratories and was first marketed in 1933 by Rohm and Haas Company, under the name Plexiglas. Since then it had been sold under many different names.
The modules are inter-connected with servomotors. Every servomotor of the n module holds and rotates the n+1 module.

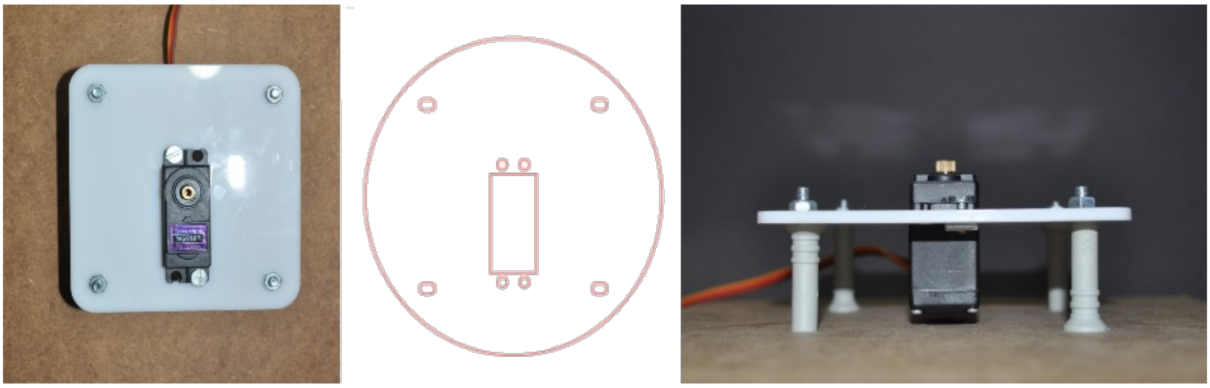


Figure 3. Design of the first module and its realization

Figure 3 shows the first module. It is mounted on a servomotor that rotates around Z axis. The module is attached directly to the base. The drawing differs from the finished module, because after the picture was taken the module broke in an experiment. At the moment, the module appears as drawn. The change of form is because of aesthetics. Also, the second version of the module used Plexiglas with a thickness of 5 mm, while the original unit is cut from Plexiglas with a thickness of 3 mm.

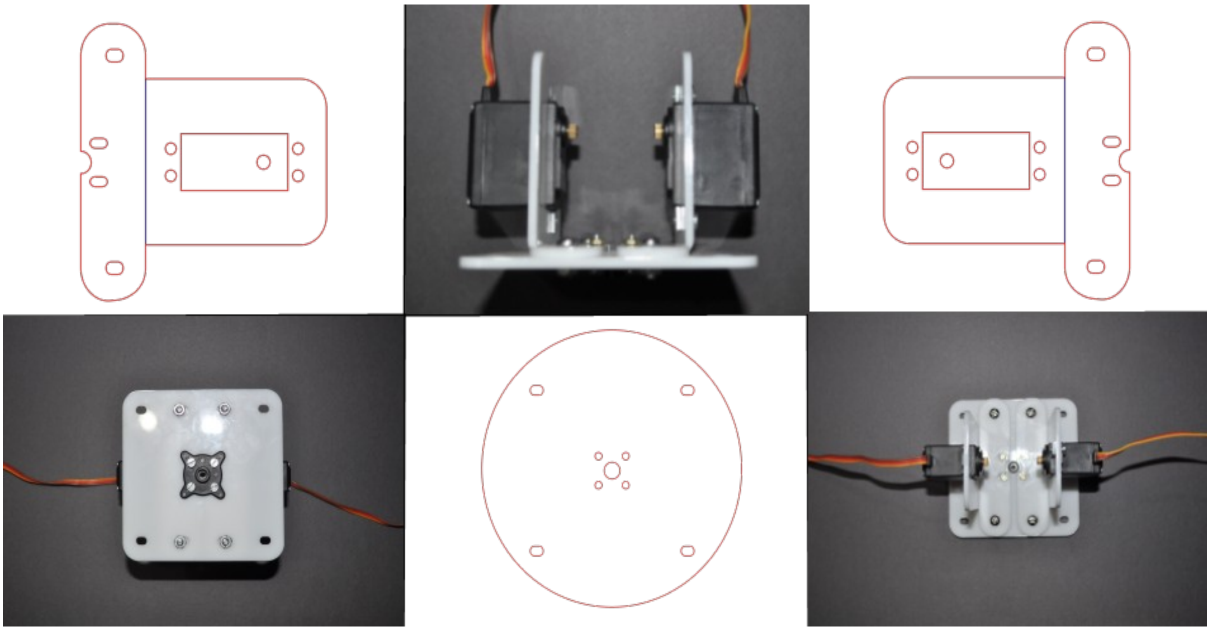


Figure 4. Design of the second module and its realization

The next module is the only one which uses two servomotors to rotate around a single axis, in this case Y axis. Also, it is the only one who uses parts that are obtained by bending Plexiglas. Bending sites are marked with blue lines. The fixing of the parts is with bolts.

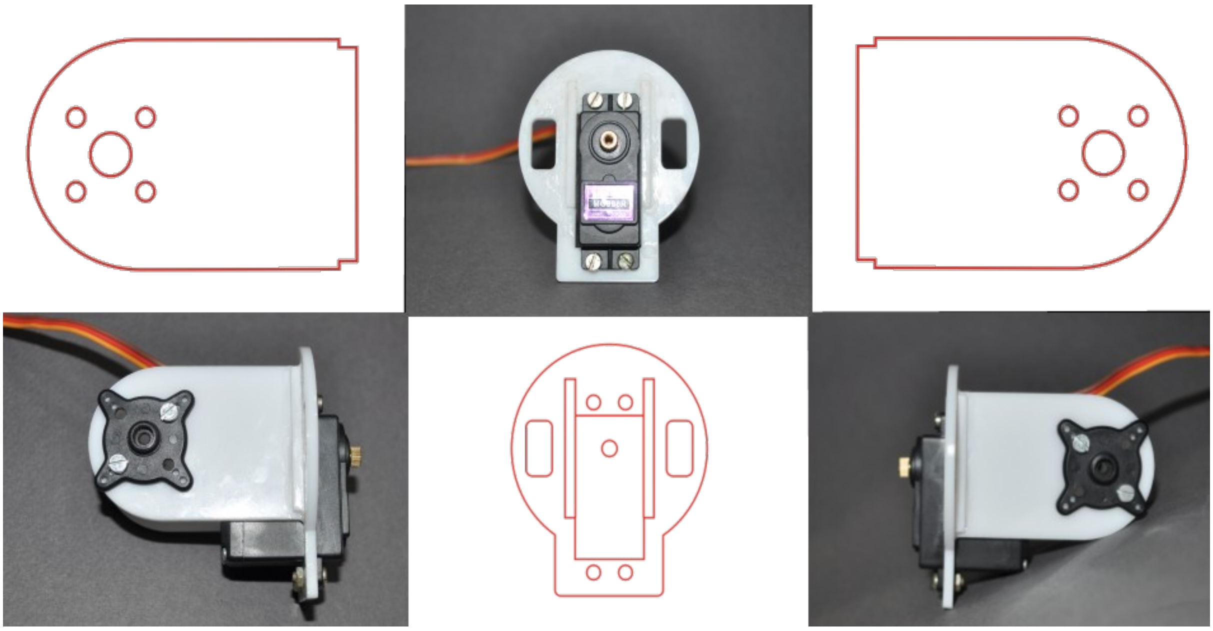


Figure 5. Design of the third module and its realization

In this module small holes are cut through which, cables will pass for servomotors used in subsequent modules. Individual parts are attached with glue.

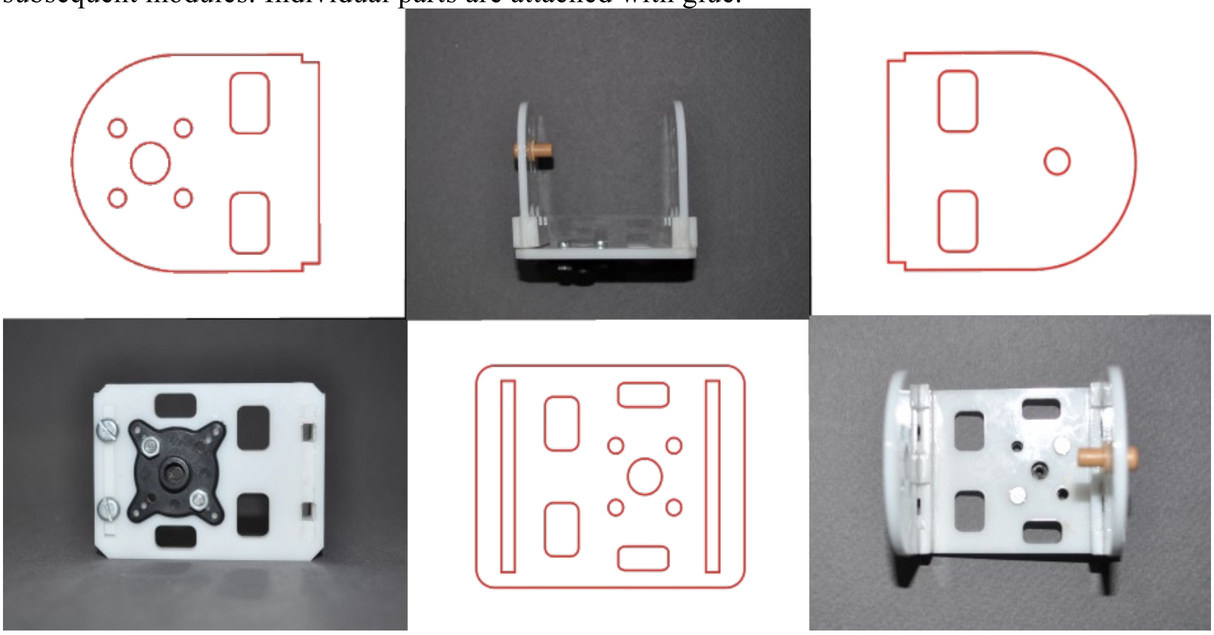


Figure 6. Design of the fourth module and its realization

This module is assembled with screws and glue. On the module itself, there won't be any servomotor mounted, but it will create an imaginary axis around which the next servomotor will be revolving. It's interesting that this module and the next one form a pair which can theoretically create a robotic arm with ***n*** degrees of freedom.

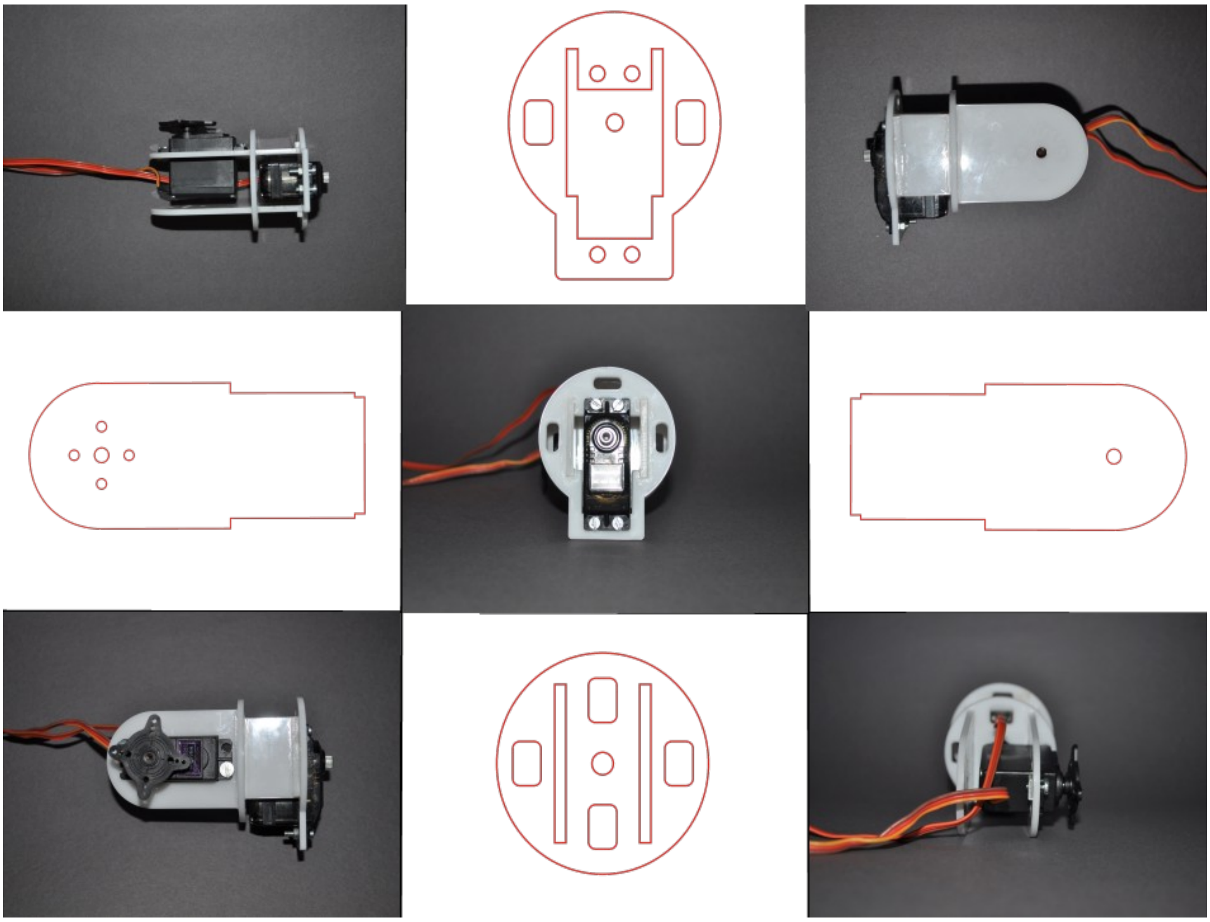


Figure 7. Design of the fifth module and its realization

This module consists of 4 parts which are connected by glue.

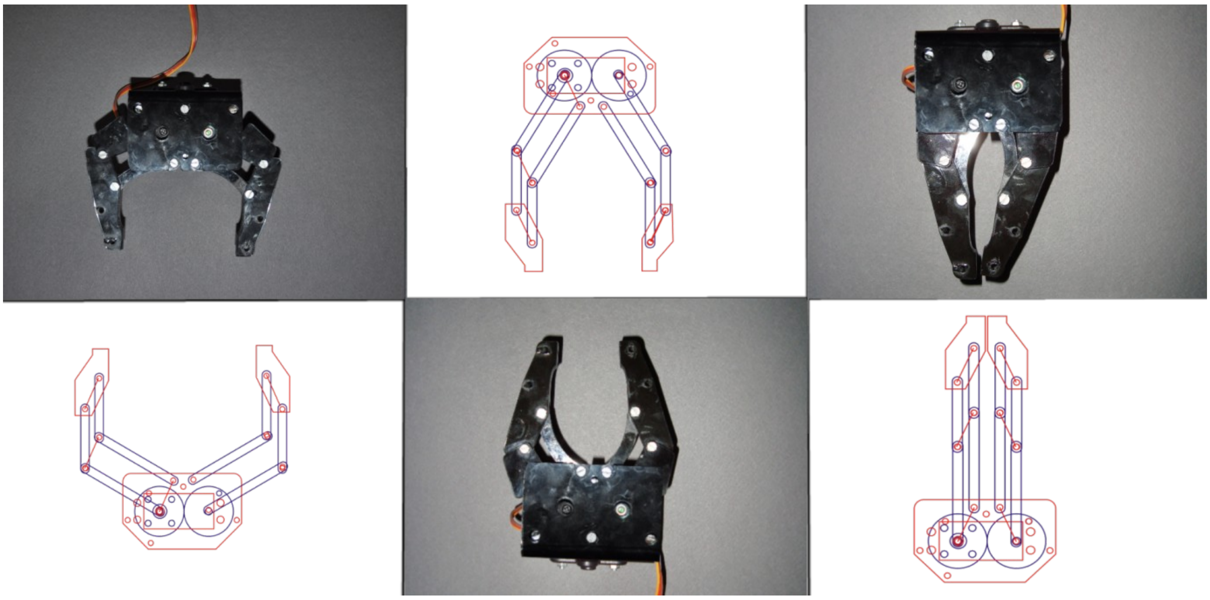


Figure 8. Design of the pincer and its realization

The end effector is designed with two fingers and forms the most complex module.

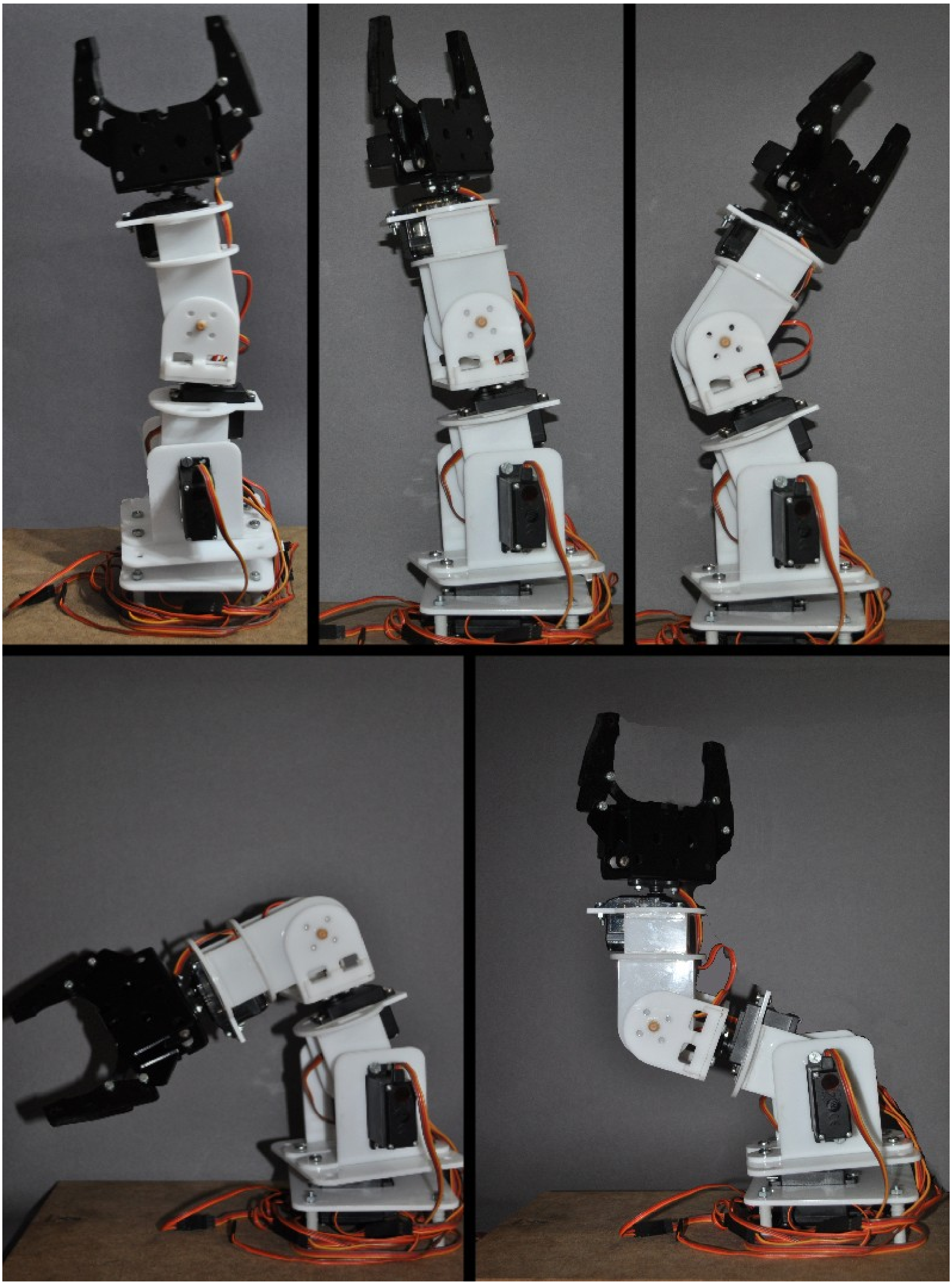


Figure 9. Photos of the assembled robotic arm

2.2 Servomotors and controller for them

Servomotor is an electric motor mechanically connected to a potentiometer (variable resistor). Pulse width modulated (PWM) signals are sent to the servomotors, which are interpreted into commands. When the servomotor is commanded to rotate, it rotates until the potentiometer reaches a certain value which corresponds to the commanded position. A standard servomotor is shown in Figure 10.

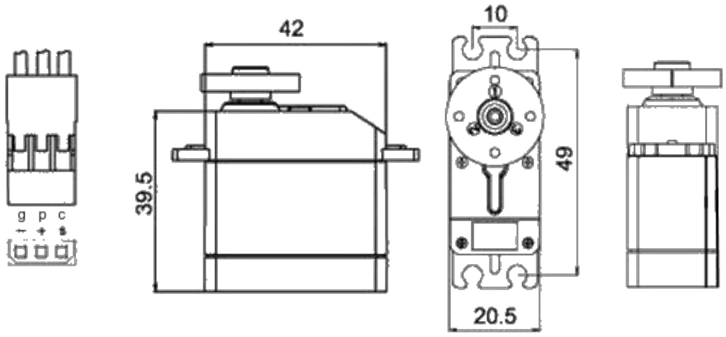


Figure 10. The standard dimensions of a servo motor

The servomotor is commanded by a pulse which occur every 20 milliseconds. The width of that pulse dictates the range of angular displacement of the servomotor. Pulse width of 1.5 milliseconds will put the servomotor in "neutral" position or 90 °.

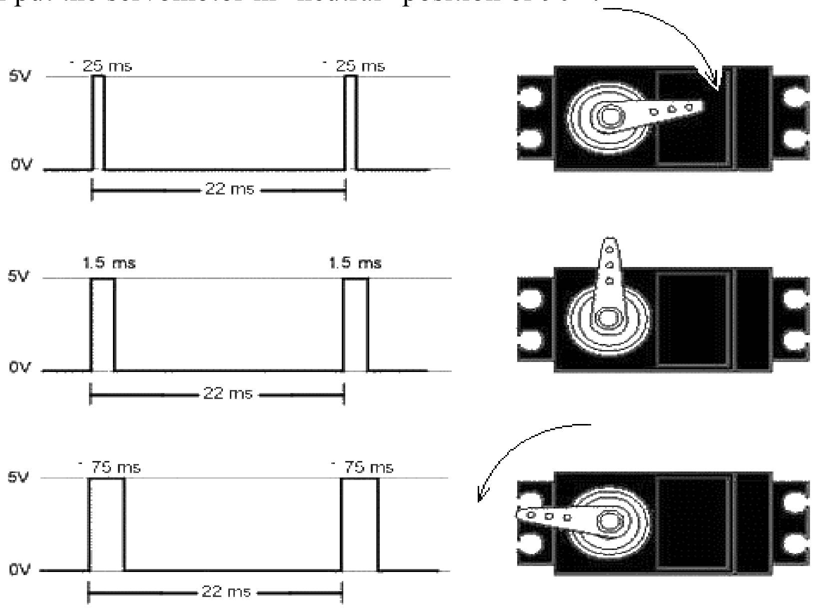


Figure 11. Basic control of a servo motor with a PWM signal

To correctly communicate with the commanding computer a servo controller is required which will create these pulse signals. One of the most often used is SSC32 [9] and it is also utilized in the described robotic arm. Containing connectors for 32 servomotors, it has high resolution (1uS) for accurate positioning, which enables servomotors to have very smooth movement. One of the more unique features is the “group command” which allows for a combination of servomotors to start and end their motion at the same time, regardless of their position. This is a very powerful tool for creating complex walking machines, which are using two or more servomotors. It connects to the computer through the RS-232 port.

2.3 The problem of inverse kinematics

Given the desired position of the robot's hand, the controller of the robotic arm must know all the angles of the robot's joints. This problem is called inverse kinematics [4]. People solve this problem without any conscious effort.
Because of its complexity, this problem is often solved by iterative algorithms. Iterative algorithms are good enough to cope with the problem of inverse kinematics, but they are usually computationally intensive and required great processing power. The algorithm suggested in the paper to be used for solving the inverse kinematics problem is usually applied in cinematography and animation.
Cyclic Coordinate Descent first presented in 1993 by Chris Uelman works with an analysis of the joints one at a time. CCD solves inverse kinematics problem through optimization. If a 2D device with 2 parts bones and 2 joints is regarded, starting from the last bone, each joint is optimized to place the endpoint of the last bone closest to the target point. This cycle is repeated until a solution is found or iterative limit is achieved. One example solution is presented in Figure 12.The arm is composed of ***n*** number of bones and joints.

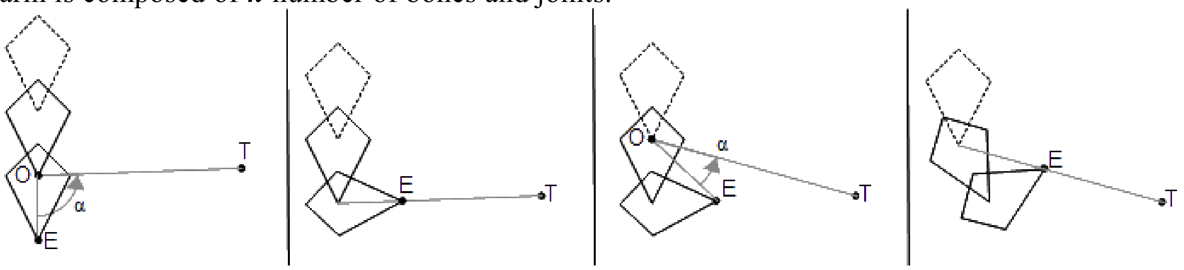


Figure 12. An example of solving inverse kinematics using the CCD algorithm

The full algorithm used for the inverse kinematics problem of the designed robotic arm is shown in Figure 13. The following notations of variables and functions are used:

- ***n*** is the number of bones in the arm.
- ***threshold*** is the maximum distance at which it is considered that the arm has reached the target.
- ***Max*** is the maximum number of iterations.
- ***i*** is the number of current bone, for which the rotation is calculated. The initial value is ***n.***
- ***j*** is used as a counter for iterations.
- ***d*** is the distance between **E** and **T.**
- ***α*** is the angle that's between ***OT*** and ***OE,*** and is calculated as a scalar product.

To move the end point to its optimum position the bone should be rotated by the unknown angle ***α.*** The known variables are:

- current position of the bone (point ***A)*** which forms a vector ***j*** with the coordinate start
- position of the end point (point ***E)*** forming vector ***e*** with the coordinate start
- target point (point ***T),***which forms a vector ***t*** with the coordinate start.

α is the angle between vectors ***(e-j)*** and ***(t-j),*** which is needed to turn the ***(e-j)***.One way of calculating it is by using the scalar product between these two vectors.

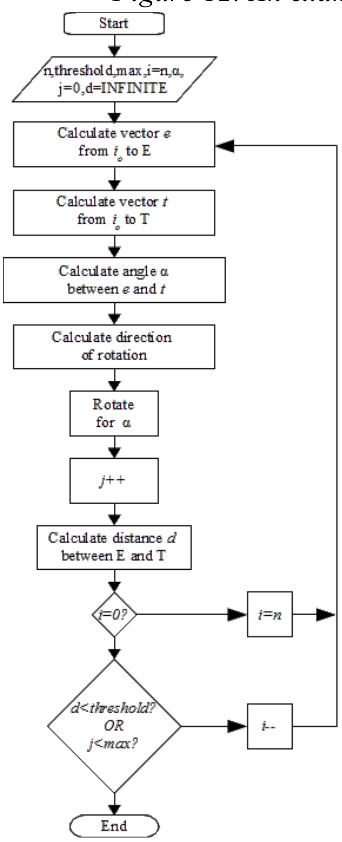


Figure 13. The CCD algorithm

Scalar product is the multiplication between two vectors:

$a \cdot b = |a||b| \cos \theta$

where

- $|a|$ is the length of vector ***a***
- $|b|$ is the length of the vector ***b***
- θ is smaller angle between ***a*** and ***b***

Using this formula the angle ***α*** can be determined**:**

$$\alpha = \cos^{-1}\left(\frac{(e-j)\cdot(t-j)}{|e-j||t-j|}\right)$$

In this equation the value of ***α*** is calculated and not its direction of rotation. For this purpose ***sin(α)*** have to be calculated using 3D vector product,

$$a \cdot b = |a||b| \sin(\theta) n$$

where ***n*** is the unit vector perpendicular to ***a*** and ***b*** in the direction defined by the right hand rule. If the vector product of two 3D vectors is used:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \times \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix}$$

and 2D vectors ***a*** and ***b*** are replaced the result is as follows:

$$\begin{bmatrix} a_x \\ a_y \\ 0 \end{bmatrix} \times \begin{bmatrix} b_x \\ b_y \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ a_x b_y - a_y b_x \end{bmatrix}$$

Further more aligning the equations:

$$\begin{bmatrix} 0 \\ 0 \\ a_x b_y - a_y b_x \end{bmatrix} = a \cdot b = |a||b| \sin(\theta) \begin{bmatrix} 0 \\ 0 \\ n_z \end{bmatrix}$$

the value of ***n*** is obtained**,** which changes between –1 and 1 and allows to determine the direction of rotation. For ***sin(x)*** applies

$-\sin(x) = \sin(-x)$

and hence

$n_z \sin(\theta) = \sin(n_z \theta) = \varphi$

where ***φ*** is the angle ***θ*** with the direction of $\boldsymbol{n_z}$.

$a_x b_y - a_y b_x = |a||b| \sin(\varphi)$

If ***a*** is substituted with ***(e-j), b*** with ***(t-j)*** and ***Φ*** with ***α*** the formula will be as follows:

$(e_x - j_x)(t_y - j_y) - (e_y - j_y)(t_x - j_x) = |e-j||t-j| \sin \alpha$

$$\sin \alpha = \frac{(e_x - j_x)(t_y - j_y) - (e_y - j_y)(t_x - j_x)}{|e-j||t-j|}$$

$$\alpha = \sin^{-1}\left(\frac{(e_x - j_x)(t_y - j_y) - (e_y - j_y)(t_x - j_x)}{|e-j||t-j|}\right)$$

The last actually solves the problem of inverse kinematics.

The above described algorithm is programmed using Microsoft C# 4.0. The program uses ***IO.Ports*** class for the communication with ***SSC32*** through the serial port.

3 Results and analysis

The program to implement the inverse kinematics problem and to control the constructed robotic arm is written as a console application. In order to use the robot the following conditions should be fulfilled:

- SSC32 is connected to a power source
- servomotors are connected to a power source
- computer on which the program will be run, is connected to SSC32 in serial port

If the program cannot find a solution because the target point is too far or the angles are impossible for the servomotors, the program displays and sends the result, which was closest to the target point. In this way the program response mimics a human response if the arm can not reach the goal. Screenshot of the program with the indicative target coordinates (15.5,15.5,15.5) is shown in Figure 14.

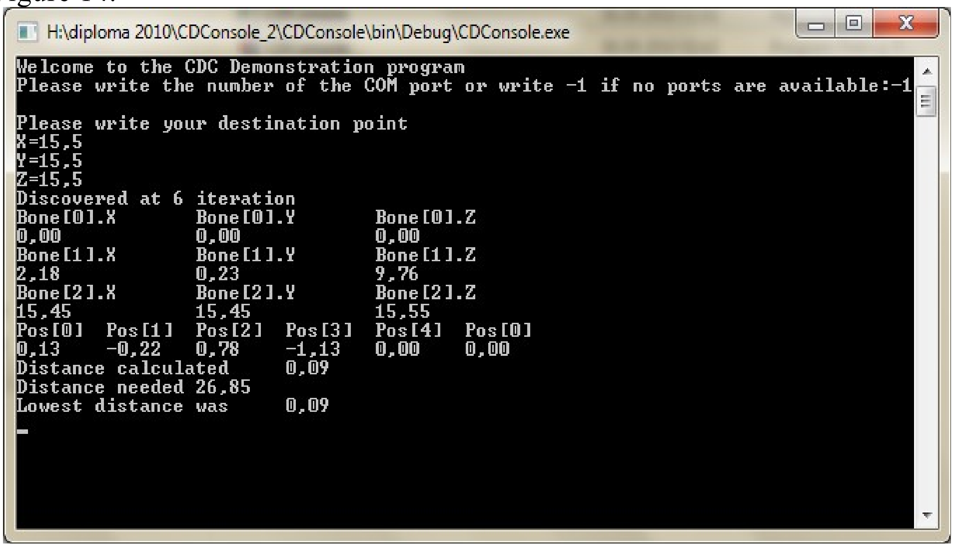


Figure 14. A screenshot of the controlling program

The program requires two input values:

- The number of the port through which the servo controller is connected.
- Coordinates of the target point that the robot should reach.

After calculation the results and auxiliary variables are printed out:

- ***Discovered at X iteration*** indicates how many iterations the program needed to find a solution.
- ***solution.Bone x, Bone y, Bone z*** shows the coordinates (in centimeters) of all the bones of the robotic arm.
- ***Pos [i]*** - shows the angles in radians of the joints.
- ***Distance calculated X*** shows the distance between the end point and the the target point.
- ***Distance needed X*** shows the distance from the starting position of the arm to the target point.
- ***Lowest distance was X*** shows the distance between the end point and the target point at the best case.

4 Conclusion

A prototype of a robotic arm was designed and constructed. Cyclic Coordinate Descent algorithm that is intensively used in computer graphics, game development and animation is suggested to be utilized for solving the inverse kinematics problem of the robotic arm. The experiments with the robotic arm prototype and the C# implementation of the algorithm for calculation of the angles of joints of the robot proved that the suggested solution is able to provide a correct solution for the robotic arm control and faultless operation of the robot.

Further work on the prototype and some improvement in order to provide a solution with precision required by the industrial implementations will include:

- use of more precise and more powerful servomotors;
- use of servo controller with better resolution of the PWM signal;
- use other electronics in the servomotors;

One of the alternatives is the project OpenServo [10] which is open-source replacement for the built-in electronics in the servomotors. OpenServo is one of the easiest ways to increase the precision of servomotors. Moreover, OpenServo is using serial bus I2C to communicate with servomotors which significantly reduces the number of wires leading to the servomotors.

Additionally a graphical interface will be added to the robotic arm control program. In order to control the movements of the robot by determining the maximum acceleration the algorithm will be improved by implementing acceleration and speed of the robot joints.

References

[1] *Cognex – Vision Technology*, http://www.cognex.com

[2] Y. Bar-Cohen, D. Hanson, The Coming Robot Revolution: Expectations and Fears About Emerging Intelligent, Humanlike Machines, Springer, 2009.

[3] P. Menzel, F. D'Aluisio, Robo sapiens: Evolution of a New Species, MIT Press, 2001.

[4] M. Spong, S. Hutchinson, M. Vidyasagar, Robot Modeling and Control, Wiley, 2005.

[5] Veljko Potkonjak, *Robotika,* Beograd, 1989. ISBN :86-23-420022-5

[6] Veljko Potkonjak, *Savremeni roboti: uvod u teoriju i primenu,* Beograd, 1986. ISBN :86-325-0057-0

[7] Miomir Vukobratovich, *Primenjena dinamika manipulacionih robota, modeliranje, analiza, primeri*, Beograd, 1990. ISBN 86-325-0213-1

[8] Daniel Sanchez-Crespo Dalmau, *Core Techniques and Algorithms in Game Programming,* Berkeley, California, 2003. ISBN :0-13-102009 -9

[9] *SSC-32 manual*, http://www.lynxmotion.com/images/html/build136.htm

[10] OpenServo: Open Source in Motion, http://www.openservo.com

IVAN VASILOV
Technical University of Sofia
Faculty of Computer Systems and Control
8 St. Kliment Ohridski blvd, Sofia 1000
BULGARIA
E-mail: vasilov.ivan@gmail.com

First International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2011, April 7-9
Sibiu, Romania

Parallel - A Zero-Client Computing Application

Mihai Stancu
Coordinator: Prof.PhD Dana Simian

Abstract

The purpose of this paper is to introduce the concepts of zero-client computing and the design and implementation of Parallel, a software solution that allows rapid deployment of such architectures. This leading-edge technology enables peripheral devices to be deployed at any user interaction point without a dedicated PC or thin client at that same location and without requiring any modifications to existing software applications. Topics include a zero-client definition, benefits, deployment scenarios, cabling options, technical details, software impact, management, and technology comparisons.

1 Introduction

In today's client/server world, the norm for most office, commercial or industrial applications is a dedicated PC or workstation running an application and accessing a variety of peripheral devices locally. The PC is then connected to a network server for backup and communication. This is the application architecture found in universities, libraries, retail, financial, airline, food service, manufacturing and most other commercial markets [1].

While very flexible, this architecture has a variety of high costs: capital costs: initial new equipment, initial configuration and installation, new infrastructure, cabling, training and design of administration, security and backup routines; and ongoing costs: repairs as equipment breaks down or wears down, replacing units to keep current with industry standards/needs, hardware upgrades, software and operating system upgrades, user created challenges, technical resources to perform regular maintenance and support [8].

Standard PC hardware has unreliable moving parts such as hard drives and fans, with expected life cycles of only three to five years [1].

Finally, computers have an unseen environmental impact in all phases of their lifecycle: from production, through use, to their disposal at end-of-life. An average desktop computer with monitor requires approximately 10 times its weight in chemicals and fossil fuels to produce. During use, a computer's electricity consumption and heat output have the most critical impact on the environment. When a computer reaches the end of its useful life, hazardous substances and materials contained in it such as heavy metals and brominated compounds place a heavy burden on the environment. Between 1997 and 2007 nearly 500 million personal computers became obsolete [3].

2 Zero-client computing

Zero-client technology has its earliest roots in mainframe computing, where computing tasks and program execution were centralized and information was sent and displayed to multiple users through terminal devices that lacked programmable intelligence, ergo, "dumb terminals" [2].

The zero-client is a set of components (monitor, keyboard and mouse), with no independent programmable intelligence, that rely on a centralized CPU for all program execution and information processing. The connection between the zero-client and the Host PC is a direct, point-to-point connection, requiring no network protocol. Zero-clients are typically implemented in clusters, using a "star-like" configuration [2].

Each cluster can function either as a network component of a distributed computing system or as a self-contained system. When combined with the high performance of point to point connections, zero-client technology offers an unequalled platform for small-group systems accessing shared resources.

Zero-clients offer smaller cost per seat than PCs and a single point location for upgrade, maintenance and support, drastically reducing licensing and lifetime system costs.

3 Parallel

Parallel is a Windows-based software solution that uses a single system to host multiple, simultaneous and independent user stations. Compared with other virtualization or thin client solutions, Parallel helps reduce the total cost of ownership with lower hardware acquisition costs, reduced IT management complexity, and reduced energy consumption. It provides a simplified IT solution for school classrooms, libraries and businesses.

In a Parallel solution, a user station consists of a station hub, monitor, keyboard, and mouse. The station hub can be a standard USB hub which adds more USB ports and extends the distance between the station and the host system. The station hub can also be a multifunction hub, which delivers connectivity to the end stations for their display, keyboard, mouse, and other peripherals such as speakers, headphones, or USB flash drives.

3.1 Hardware Requirements

A Parallel host system can support many standard stations (the number depends on hardware limitations) and a primary station (the first station that becomes active when Windows is powered on).

The following considerations must be taken into account when planning the system:

- The maximum distance from the host system to a station hub is 10 meters. An intermediate USB hub, with external power supply, or a USB repeater cable is needed for each 5-meter extension.

- The maximum levels of USB hubs is 5 (i.e., no more than 5 hubs can be daisy-chained together). Microsoft does not recommend more than 3 levels to ensure reliability.

The following table lists the recommended minimum requirements for processor and memory, based on the number of stations and type of application scenarios.

Station Scenarios	Productivity	Mixed	Graphics Intensive
	Word, Excel, E-mail, Web	Productivity software, video	Productivity software, graphics
1-5 Stations			
CPU	dual-core	dual-core	x64 dual-core
RAM	2 GB	4 GB	4 GB
5-10 Stations			
CPU	dual-core	quad-core	x64 quad core
RAM	4 GB	6 GB	6 GB

Table 1: Processor and Memory Requirements

3.2 Architecture

Depending on the types of station hubs used, there are two major architecture options when deploying Parallel: connection via video cable and USB hub, and connection via multifunction hub.

3.2.1 Connect via Video Cable and USB Hub

The first option is to set up the end station by connecting its keyboard and mouse via a standard USB hub and connecting its display directly to a graphic port on the host system. This architecture requires the host system to have a multi-port graphic card(s) to provide video directly to the end stations.

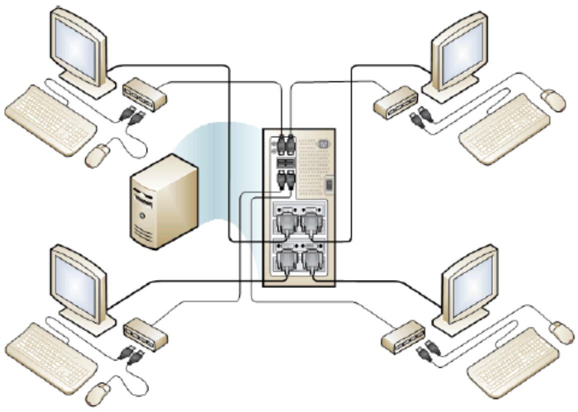


Figure 1: Parallel solution with four stations [11]

3.2.2 Connect via Multifunction Station Hubs

In the second architecture, each end station is connected with a multifunction hub which provides video, keyboard and mouse connectivity. Some multifunction hubs also provide additional peripheral support such as audio and USB flash drive. When using a multifunction hub, the host system can have a single port integrated graphic controller instead of a multi-port graphics card.

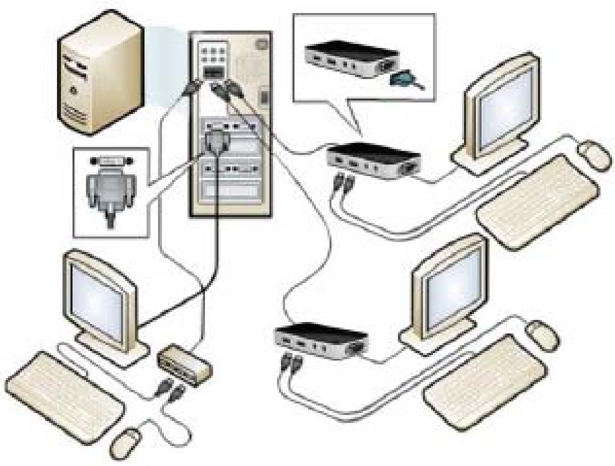


Figure 2: Parallel with Multifunction Hubs [11]

3.3 Deployment

The following guidelines must be followed when setting up Parallel. Unlike similar applications, Parallel distinguishes itself through the simplicity of its setup process, and ease of use.

3.3.1 Setting up the Host System

Before connecting the standard stations to the host system, it is important to complete the Parallel system installation on the host system:

1. Set up the primary station by connecting a USB hub and display directly to the host system, and then connecting the keyboard and mouse to the USB hub.

2. Install Parallel on the host system. A simple wizard guides the user through the installation process. System administrators can create a customized bootstrapped installation package to automate the installation on different hosts.

There is no unique driver requirement for Parallel. The latest Windows drivers for devices such as the video adapter, network adapter, and sound card, are necessary for a successful installation.

Once the host system installation is complete, the user is ready to physically set up standard stations.

3.3.2 Setting up Stations

As illustrated in Figure 1, the displays of standard stations can be connected directly to one of the graphic ports on the host system. Users must:

1. Ensure that the proper display driver is installed.

2. Turn off the host system and unplug it.

3. Connect a keyboard and mouse to a USB hub, and connect the USB hub to a USB port of the host.

4. Connect a monitor to one of the host video ports.

5. Repeat Steps 3 and 4 for the remaining stations.

6. Connect the power for the host and turn it on.

7. Follow the instructions on each display to manually associate the devices with the station.

In order to install multifunction hubs, the user must follow the manufacturer’s instructions. To set up some manufacturers’ multifunction hubs, manual association is required by pressing the required key prompted on the screen. The correct key entry triggers the manual association for the keyboard, mouse and display for that station.

3.4 Management

Like Windows MultiPoint Server, Parallel provides administrative users with the Parallel Manager to monitor and manage the stations, hardware status, user sessions, and user accounts. There are three panels in Parallel Manager: Desktop, Hardware, and Users.

At the Desktop panel, an administrator can view the current users and can manage them by either ending their session or disconnecting them from the Parallel station.

At the Hardware panel, administrators can view all the connected hardware devices for each station.

At the Users panel, administrators can manage the user accounts and add a new user. Details on how to conduct other management tasks such as installing updates, and troubleshooting are available in the help file and video tutorials accompanying the installation media.

3.5 Technical Challenges

Creating Parallel has proven to be a challenge in every step of the way, from the design process to the actual implementation.

Even though Windows does not technically prohibit the use of multiple input devices simultaneously, implementing a program that allows parallel use of multiple keyboards and mouse devices is far from trivial. It is true that many keyboards and mouse devices can be connected to the same computer and function properly, but the operating system serializes their input. For example, if the 'A' key is hold down on one keyboard, and the 'B' key is hold down simultaneously on another keyboard, the result would be 'ABAB…'. This problem was solved using global hooks.

Something similar happens to mouse devices: only one can control the pointer on the screen, at any given time. This problem is solved by manually reading input from mouse devices and interpreting it accordingly in order to place differently colored pointers at their correct locations on the screen, and manage their movements.

Yet another problem is the following: If two users are working in parallel with different windows, what window is the active window? If the windows overlap, which window is on top? If we just allow the primary user (i.e. the user with the primary mouse and primary keyboard) to control the active window and a secondary user to send his input to an inactive window, how does the secondary user know what control has focus to receive the input at a given moment?

Virtualization would have been a very simple solution, and is in fact, the one used in similar applications like Userful, for Linux, and Windows Multipoint Server. But virtualization comes with a high cost in performance. Instead, Parallel solves these issues and many more, in a straightforward and elegant manner, without the need for simulating an entire operating system for every one of its users.

3.6 Implementation Details

The language of implementation is C++, which has clear advantages over all other options available. A special interest was given to compiler optimization.

Even though the source code presents many original, creative solutions to overcome operating system limitations, the development of Parallel would have been far more difficult without the help of two remarkable open source applications, namely AutoScript and Windows Pager. In the spirit of these open-source projects, the full source code of Parallel will be published on its hosting website.

3.7 Future Development

Although the main design goals have been reached, there are still many ideas that have not been implemented in the current version of Parallel, but which are planned for future development.

The most important are: adding an intermediate layer of security to allow full privacy for all users, allowing users to customize their personal account and allow them to save their session (for example: the state of all opened windows). Also there currently is an issue regarding applications that only allow one instance to be opened at a time. Finally, another important task is to create a secure web server for automatic online update.

After hosting the project on a website, contributors will be welcomed to participate in the development of Parallel, enhance current functionality and also bring new ideas to the project.

Because no software solution is perfect, a specialized bug tracking web application has been created, to help better coordinate the development process, and monitor progress.

4 Competitive Analysis

Many virtual desktop infrastructure vendors claim that they offer zero clients of one form or another, but these are usually just thin clients that need additional hardware and software for streaming input and output [5].

The term zero-client is often misapplied to thin client endpoints in thin client vendor marketing materials. True zero client endpoints do no local processing and have no client operating systems, drivers, software, or storage. They are completely stateless and management-free. Zero clients mean zero endpoint management.

Parallel offers true zero client capabilities and has many advantages over its competitors. The following table illustrates the key differences between Parallel, Userful and Microsoft Windows Multipoint Server.

	Parallel	Userful	Multipoint
OS	Windows	Linux	Windows (x64)
No. Stations	Unlimited	11	11
Licensing	Free	Simple	Complex
Use Cur. OS	Yes	Purchase	No
Domain Join	Yes	Yes	License
Setup Time	5 min	30 min	4 hours
Performance	Fast	Fast	Slow
3D Graphics	Hardware	Software	No
Audio	Yes	Yes	No
USB Access	Yes	Yes	No

Table 2: Competitive Analysis

5 Conclusions

The use of zero-clients is a fundamental departure from current PC-based infrastructures. However, the potential benefits are immense; each zero-client can cost as much as 80 per cent less than a PC [6], and no further investments are necessary to establish the network infrastructure. Also, zero-clients use far less energy, and are more environmentally-friendly.

It is no longer the case that IT purchasers are solely interested in cost and performance. Environmental concerns are playing now an increasingly important role. As the cost of hardware decreases and the cost of service, energy and waste disposal rises, the environmental and economic advantages of zero-client computing grow.

Zero-clients are capable of replacing PCs everywhere. Universities and public institutions in general can benefit from this emerging technology, but in areas such as manufacturing and vertical markets, they can be especially useful. Zero-clients offer a bold new solution for improving the cost, flexibility, reliability, security and environmental impact of client server architectures [4].

Migrating to zero-client based solutions is now easy, and free of cost, thanks to Parallel, a program built with efficiency and high performance in mind.

Parallel eliminates the need for virtualization which degrades CPU processing. As in the personal computer model, the display adapter resides in the same computer as the application, which preserves performance because it eliminates the need for a network transmission protocol that injects delays due to network overhead.

Parallel thus allows more users to have access to feature rich applications without the need for hardware upgrades, and without requiring changes to existing software, providing numerous advantages over similar applications, and limitless possibilities.

References

[1] Frank Ohlhorst, *Replacing Desktop PCs with Zero-Client Solutions*, http://www.channelinsider.com/c/a/Networking/Replacing-Desktop-PCs-with-ZeroClient-Solutions-179077/
[2] Applica Inc., Zero Client, http://www.applica.com/zeroCtech.shtml
[3] Wise Technology Inc., Environmental Benefits of Thin Computing, White Paper, http://www.wise.com
[4] Digi International, *Zero-Client Computing*, White Paper, http://www.digi.com
[5] Pano Logic, Zero-Clients vs. Thin Clients, http://www.vmworld.com/servlet/JiveServlet/previewBody/4554-102-1-5753/PanoLogic_WP_Zero_vs_Thin_Clients-ZvTC-041410.pdf;jsessionid=A6E12B4360244333B4D336700C4
[6] Userful, official website, http://userful.com/
[7] Windows Multipoint Server, official website, http://www.microsoft.com/windows/multipoint

Mihai STANCU
"Lucian Blaga" University of Sibiu
Faculty of Sciences, Mathematics Department
10, Victoriei Bd., Sibiu, 550024
ROMANIA
Email: mihai.stancu@sobis.ro

PARTICIPANTS LIST

ALEXE Iulian
"Lucian Blaga" University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail: iulian.alexe@ulbsibiu.ro

AREZINA Jovo
Megatrend University
Faculty of Computer Science
Bulevar umetnosti 29, 11070 Belgrade
SERBIA
E-mail: jovo.arezina@gmail.com

AVDEEVA Ekaterina
Vladimir State University
Management and Computer Science in Technical and Economic Systems
Gorky St., 87, Vladimir
RUSSIA
E-mail: avdkaterina@rambler.ru

BALAN Oana
Petroleum-Gas University of Ploiesti
Informatics Department
no. 39 Bd. Bucuresti, 100680, Ploiesti
ROMANIA
E-mail: oanab_2005@yahoo.com

BOSTAN Adrian George
“Dunarea de Jos” University of Galati
Faculty of Computer Science
111 Domneasca St. , Galati
ROMANIA
E-mail: wait4.asecond@gmail.com

BOTA Florentin
"Lucian Blaga" University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail: botaflorentin@yahoo.com

BRAILEANU Vasile
"Dunarea de Jos" University of Galati
Faculty of Computer Science
111 Domneasca St. , Galati
ROMANIA
E-mail: vasilebraileanu@yahoo.com

BUCUR Bogdan Ion
Petroleum-Gas University of Ploiesti
Faculty of Letters and Sciences
Advanced Information Processing Technology Master
Ploieşti, Prahova
ROMANIA
E-mail: b.bucur@yahoo.com

BUHA Radu
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, 550012
ROMANIA
E-mail: braduadrian@yahoo.co.uk

BURYCK Ksenia
Kharkiv National University of Ecomomics
Faculty of Economic Informatics
61001, Kharkiv, pr. Lenina, 9a
UKRAINE
E-mail: mykitana@mail.ru

CAPUZZO Francesco
"Lucian Blaga" University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail: francesco.capuzzo@gmail.com

CIOROBEA Bogdan
Petroleum-Gas University of Ploiesti
Informatics Department
no. 39 Bd. Bucuresti, 100680, Ploiesti
ROMANIA
E-mail: bobo_crb@yahoo.com

CIOROGAR Ciprian
"Lucian Blaga" University
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail: contact@virtualpc.ro

DANOV Ivan
Technical University of Sofia
Faculty of Computer Systems and Control
8 St. Kliment Ohridski blvd, Sofia 1000
BULGARIA
E-mail: me@danov.eu

DODA Gheorghe
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu,550012
ROMANIA
E-mail: godlike_affairs@yahoo.com

DOROKHOV Mykhaylo
Kharkiv National University of Economics
Faculty of Economics Informatics, Chair of Informations Systems
Prospekt Lenina 9-A, Kharkiv, 61001
UKRAINE
E-mail: mikedorohov@meta.ua

FIERARU Mihai
Petroleum-Gas University of Ploiesti
Informatics Department
no. 39 Bd. Bucuresti, 100680, Ploiesti
ROMANIA
E-mail: mihai.vest@yahoo.com

FILCHEVA Magdalena
Technical University of Sofia
Computer Science and Technologies
BULGARIA
E-mail: magdalenafilcheva@abv.bg

HULEA Razvan
University of Bucharest, Faculty of Mathematics and Computer Science
Department of Applied Mathematics
Str. Academiei 14, Sector 1, 010014, Bucuresti
ROMANIA
E-mail: hulea.razvan89@gmail.com

IGNISCA Madalin
"Lucian Blaga" University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail: madalin.ignisca@gmail.com

IORDACHE Madalina
Petroleum-Gas University of Ploiesti
Informatics Department
no. 39 Bd. Bucuresti, 100680, Ploiesti
ROMANIA
E-mail: lilium_martagun@yahoo.com

IVAN Andrei
Transilvania University of Braşov
Faculty of Mathematics and Computer Science
Str. Iuliu Maniu nr. 50, Braşov
ROMÂNIA
E-mail: antraxx_cs@yahoo.com

KOTSEV Vladimir
Ruse University
Department of Informatics and Information Technologies
8 Studentska Str., Ruse
BULGARIA
E-mail: vladimir.kotsev@gmail.com

LAZAR Valentina
"Lucian Blaga" University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail: lyah_valy@yahoo.com

MANTA Alin
"Lucian Blaga" University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail: mantalin@gmail.com

MARINOV Petar
Technical University of Sofia
Faculty of Computer Systems
Sofia, blvd. Kl. Ohdiski 8
BULGARIA
E-mail: petjofy@hotmail.com

MILOSEVIC Marko
University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: markoub@gmail.com

MIREA Felix
"Lucian Blaga" University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail:: mirea_felix@yahoo.com

MIRONOV Kaloyan
University of Ruse
Informatics and information technology
#8 Studentska Str.
BULGARIA
E-mail: mail@kaloian.net

MORARIU Mihai
University of Bucharest
Faculty of Mathematics and Computer Science
Department of Applied Mathematics
Str. Academiei 14, Sector 1, 010014, Bucuresti
ROMANIA
E-mail: mihaimorariu@gmail.com

MORARIU Stelian
"Lucian Blaga" University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail: modev_st@yahoo.com

NEAGU Andrei
“Lucian Blaga” University of Sibiu
“Hermann Oberth” Faculty of Engineering
Department of Computer Science
4 Emil Cioran, Sibiu, Romania 550025
ROMANIA
E-mail: andrei.neagu@ropardo.ro

NECHIFOR Vasile Nicuşor
"Lucian Blaga" University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail: vasilenicusor@yahoo.com

NICOLESCU Dragos
University of Craiova
Faculty of Mathematics and Computer Science
Department of Computer Science
Str. A.I. Cuza, nr 13, Craiova
ROMANIA
E-mail: dragosm.nicolescu@yahoo.com

OBANCEA Dragoş Iulian
Transilvania University of Braşov
Faculty of Mathematics and Computer Science
Department of Computer Science
Street Iuliu Maniu, Nr. 50, Braşov
ROMÂNIA
E-mail: obancea_dragos@yahoo.com

PAMFILOIU Claudia Elena
"Lucian Blaga" University of Sibiu
Faculty of Economic Sciences
Accountancy and financial informatics
Calea Dumbravii nr.17
ROMANIA
E-mail: pamfiloiuclaudia@yahoo.com

PĂNESCU Adrian Tudor
“Alexandru Ioan Cuza” University of Iaşi
Department of Computer Science
16, General Berthelot, Iaşi, 700483
ROMÂNIA
E-mail: adrian.panescu@info.uaic.ro

RĂDOI Ovidiu
Transilvania University of Braşov
Faculty of Mathematics and Computer Science
Street Iuliu Maniu, Nr. 50, Braşov
ROMÂNIA
E-mail: oviradoi2003@yahoo.com

RADU Sorin Daniel
"Lucian Blaga" University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail: radu_s_o_r_i_n@yahoo.com

SIVKO Lidiya
Kharkiv National University of Ecomomics
Faculty of Economic Informatics
61001, Kharkiv, pr. Lenina, 9a
UKRAINE
E-mail: sneg_snegok@i.ua

STAN Alexandra Elena
"Lucian Blaga" University of Sibiu
Faculty of Economic Sciences
Informatics in Economy
Calea Dumbravii no.17
ROMANIA
E-mail: alexandra.stn@gmail.com

STÂNEA Paul
"Lucian Blaga" University of Sibiu
Faculty of Sciences
Department of Informatics
Dr. Ioan Raţiu St. No. 5 - 7
ROMÂNIA
E-mail: psb77black@yahoo.com

STANESCU Razvan
Petroleum-Gas University of Ploiesti
Informatics Department
no. 39 Bd. Bucuresti, 100680, Ploiesti
ROMANIA
E-mail: razzvan19@yahoo.com

STOIMENOV Dimitar
Technical University of Sofia
Computer systems and technologies
BULGARIA
E-mail: dimitar.ns@gmail.com

TASIC Jelena
University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: jelenatasicsd@yahoo.com

TUBA Ana
University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: anatuba89@gmail.com

TUDORICĂ Gabriel
“Lucian Blaga” University of Sibiu
Faculty of Sciences
Sibiu, Dr. Ioan Raţiu St. No. 5 - 7
ROMÂNIA
E-mail: office@eyenetworks.ro

VALJEVIC Igor
University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: igor.valjevic@gmail.com

VASILOV Ivan
Technical University of Sofia
Faculty of Computer Systems and Control
8 St. Kliment Ohridski blvd, Sofia 1000
BULGARIA
E-mail: vasilov.ivan@gmail.com

VOLOSINCU Bogdan
"Lucian Blaga" University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail: bogdan_voloshincu@yahoo.com

Sponsors (in alphabetic order):

CRISPUS PUB

EBS Romania – Expectations Beyond Software

PAN FOOD

ROPARDO

SCANDIA FOOD

SOBIS – Solution for Business

Top Tech

TRIDENT

Media partners:

Eveniment TV Sibiu