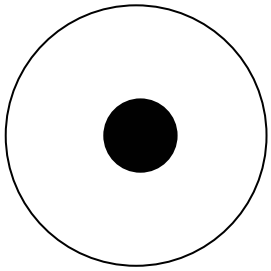
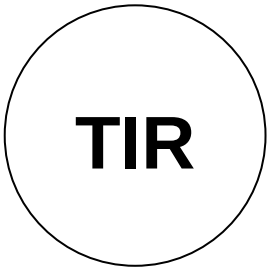




The IPSI BgD Transactions on Internet Research

Multi-, Inter-, and Trans-disciplinary Issues in Computer Science and Engineering

A publication of
IPSI Bgd Internet Research Society
New York, Frankfurt, Tokyo, Belgrade
January 2007 Volume 3 Number 1 (ISSN 1820-4503)

Table of Contents:

Pearls of Wisdom by Nobel Laureates:

Another View of Neural Networks
Cooper, Leon 3
TV is Dead - Long Live the WEB
Kroto, Harold 4
Challenges of Distance Learning
Richardson, Bob 4

Invited Paper:

Forgotten Ideas in Computer Architecture: It's Time to bring them Back!
Dennis, Jack 5

University of Belgrade Research Efforts:

On Reducing Overheads in CMP TLS Integrated Protocols
Radulović, B., Milan; Tomašević, V., Milo. 11
A Technique for Round-Trip Engineering of Behavioral UML Model Elements
Bojic, M., Dragan 18
Support for Knowledge Tests: Brief Summary of Regulations and Software
Bošnjaković, M., Andrija; Tartalja, I., Igor; and Protić, Ž., Jelica 25
Efficient Development and Maintenance of Enterprise Information Systems in Multicultural Developer Environment
Milovanović, M., Miloš; Milutinović, M., Veljko 30

***Regular Contributions*:**

Challenges in Geovisualization
Ragia, Lemonia; and Fiume, Eugene 35
Network Systems Intrusion: Concept, Detection, Decision, and Prevention
Pleskonjic, Dragan; Omerovic, Sanida; and Tomazic, Saso 40
Investigating Software Dependability Mechanisms for Robotic Applications
Laurent, T., Nana. 50
A Contribution to Development of Methods for E-business and E-commerce
Milutinovic, Dusan. 56

The IPSI BgD Internet Research Society

The Internet Research Society is an association of people with professional interest in the field of the Internet. All members will receive this TRANSACTIONS upon payment of the annual Society membership fee of €100 plus an annual subscription fee of €1000 (air mail printed matters delivery).

Member copies of Transactions are for personal use only

IPSI BGD TRANSACTIONS ON INTERNET RESEARCH

www.internetjournals.net

STAFF

Veljko Milutinovic, Editor-in-Chief	**Aleksandra Jankovic, Journal Manager**
Department of Computer Engineering University of Belgrade POB 35-54 Belgrade, Serbia (381) 64-1389281 (tel)	IPSI BgD Internet Research Society
vm@etf.bg.ac.yu	tir@internetjournals.net

EDITORIAL BOARD

Adeli, Hojjat	**Gonzalez, Victor**	**Milligan, Charles**
Ohio State University, Ohio,	University of Oviedo, Gijon,	Sun Microsystems, Colorado,
USA	Spain	USA
Blaisten-Barojas, Estela	**Janicic, Predrag**	**Milutinovic, Veljko**
George Mason University, Fairfax, Virginia,	The Faculty of Mathematics, Belgrade,	IPSI, Belgrade,
USA	Serbia	Serbia
Crisp, Bob	**Jutla, Dawn**	**Neuhold, Erich**
University of Arkansas, Fayetteville, Arkansas,	Sant Marry's University, Halifax,	Research Studios Austria, Vienna,
USA	Canada	Austria
Domenici, Andrea	**Karabeg, Dino**	**Piccardi, Massimo**
University of Pisa, Pisa,	Oslo University, Oslo,	Sydney University of Technology, Sydney,
Italy	Norway	Australia
Flynn, Michael	**Kiong, Tan Kok**	**Radenkovic, Bozidar**
Stanford University, Palo Alto, California,	National University of Singapore,	Faculty of Organizational Sciences, Belgrade,
USA	Singapore	Serbia
Fujii, Hironori	**Kovacevic, Branko**	**Rutledge, Chip**
Fujii Labs, M.I.T., Tokyo,	School of Electrical Engineering, Belgrade,	Purdue Discovery Park, Indiana,
Japan	Serbia	USA
Ganascia, Jean-Luc	**Frederic Patricelli**	**Takahashi, Ryuichi**
Paris University, Paris,	ICTEK Worldwide L'Aquila	Hiroshima University, Hiroshima,
France	Italy	Japan

Another View of Neural Networks

Cooper, Leon (1972 Nobel Laureate)

When interest in neural networks revived some fifteen years ago, few people believed that such systems would ever be of any use. Computers worked too well; it was felt that they could be programmed to perform any desired task.

Clearly, fashion has changed. Now, limitations of current computers in solving many problems involving difficult to define rules or complex pattern recognition are widely recognized; if anything, expectations for neural networks may be too high. The problem is no longer to convince anyone that neural networks might be useful, but rather to actually incorporate such networks into systems that solve real-world problems economically.

Neural networks are inspired by biological systems where large numbers of neurons, that individually function rather slowly and imperfectly, collectively perform tasks that even the largest computers have not been able to match. They are made of many relatively simple processors connected to one another by variable memory elements whose weights are adjusted by experience. They differ from the now standard Von Neumann computer in that they characteristically process information in a manner that is highly parallel rather than serial, and that they learn (memory element weights and thresholds are adjusted by experience) so that to a certain extent that can be said to program themselves. They differ from the usual artificial intelligence systems in that (since neural networks learn) the solution of real-world problems requires much less of the expensive and elaborate programming and knowledge engineering required for such artificial intelligence products as rule-based expert systems.

In their current state, neural networks are probably best at problems related to pattern recognition. Some existing neural network systems can efficiently and rapidly learn to separate enormously complex decision spaces. The problem of coordinating many neural networks, each a specialist in dividing some portion of the decision space, has also been solved. It is in these areas, therefore, that the first commercial uses will appear. Products that recognize characters, assembly line parts or signatures, that make complex decisions mimicking or improving on human experts (such as underwriters) that can diagnose engine or assembly line problems are in the prototype stage and/or are already fielded. One expects, further, that the pattern recognition ability coupled with, and feeding back and forth to rule-based systems (as has already been done in some simple applications) will finally result in machines that share our ability to learn and duplicate our processes of reasoning-machines that might be said to think.

The question is not whether but when.

Predicting the future, as we all know, is risky. Predicting the evolution of new technology is downright hazardous. Who in the 1930's would have said that among the consequences of the uncertainty principle would be transistors, silicon chips, and all of the vast array of solid state devices on which all modern computers depend? Or that superconductors would lead to extraordinarily sensitive detectors of magnetic fields now carried on many naval ships? Or in the late 19th century, that among the consequences of the research of Maxwell, Lorentz and Einstein, would be all that we call modern communication: radio, radar, etc.?

Accepting that risk, I would predict that neural networks will become standard components of what we today call computers. This will likely occur in a somewhat evolutionary manner: they will encroach gradually-board by board, intelligent components, that can be trained by humans in a language humans understand, into dumb machines-somewhat like neo-cortex came to dominate the reptilean brain. And, just as the 20th century is the century of automobiles, airplanes, telephones and computers, the 21st will be the century of intelligent machines. We will not only learn to live with these machines but, indeed, will wonder, one day, how we ever lived without them.

TV is dead - Long Live the WEB

Kroto, Harold (1996 Nobel Laureate)

Science, Engineering and Technology are as vital to our intellectual and cultural development (particularly our children's) as they are to our training to get along in the Modern World. Some efforts to redress the problems involved in the general Public awareness and understanding of science and Engineering (PAUSE) issues are being initiated via the Vega Science Trust (www.vega.org.uk), which aims to take advantage of the revolution in TV and Internet communications technology to improve matters. The best scientists and science communicators are being recorded and the programmes are being broadcast on BBC-TV and the Internet. Furthermore School/University outreach programmes are being developed and Vega is piloting ways in which members of the Science, Engineering and Technology (SET) community can, as individuals and groups, make important contributions. Excerpts from SET programmes will be presented. These efforts present a perspective on SET which places the cultural factors in the foreground and focuses on the intrinsic charisma of science which is hidden from many. It is now crucial that the society in general and the scientific community in particular accept that serious problems are involved in communicating science and the Internet as set to play a major role. Before the invention of the printing press there was only one book in the west - the bible - and it was hand-written by monks. After the invention the printing press book - writing and reading was democratized and this was truly the beginning of general education. In a similar way the birth of the Internet has democratized broadcasting - the broadcasting channels no longer control the dissemination of recorded material - individuals and groups of individuals can now do it themselves and so the Internet has enabled broadcasting to fulfill the promise it has always had - to be a superb educational medium.

Challenges of Distance Learning

Bob Richardson (1996 Nobel Laureate in Physics)

There is no longer question that the Internet and electronic communication are the major new tools for collaborative advances in the creation of new knowledge and in future learning. There are countless examples of highly successful professional courses taught of the Internet.

Similarly, international and multidisciplinary collaborations in scientific research based upon little contact other than through electronic communication dominate the scientific literature. Perhaps the most profound examples of distance collaboration in science are found in astronomy. The Hubble telescope has permitted astronomers to gather breathtaking images from the most remote observatory imaginable - one in orbit around the earth. A significant challenge remains. The challenge is to devise a remote mode for nonverbal communication about difficult concepts. In the shared creation of new ideas and knowledge, facial expressions and body gestures frequently play an important role in peer interactions. As the speed and bandwidth of electronic communication increase, we have the prospect that the important elements of human contact can be imitated. Without the development of sympathetic peer or mentor relationships, distance learning will remain quite sterile.

Forgotten Ideas in Computer Architecture: It's Time to bring them Back!

Dennis, Jack

Abstract—*Future computing systems will comprise multiple processors to achieve high performance. The issues surrounding the programming of such systems offer a major opportunity for innovative research in computer system architecture. The ideas of global virtual memory, functional programming, and data-cognizant control, which have long been ignored by mainstream computer architects, are relevant to addressing and solving the problems of formulating a satisfactory programming model for parallel computing, and developing computing systems that support such a model and the modular construction of software. The Fresh Breeze Project offers one approach to this goal.*

Index Terms—*Computer system architecture, functional programming, concurrent processing, parallel programming models, software modularity, virtual memory.*

1. Introduction

A revolution is in the making in the architecture of computer systems. Computer processing chips are being built with from two to hundreds of processor cores. Yet there is no generally accepted approach to writing programs that can exploit the parallelism offered by these devices. There is no agreement about addressing the "memory wall" – the problem raised by the fact that the time to fetch data from off-chip memory is now hundreds of times greater than the time to execute an instruction. A major weakness in the way parallel programs are currently written is that they are not "composable" – it is not possible to use a parallel program as a component in building a larger program without understanding and revising internal details of the component program – a violation of basic principles of modular programming.

This article reviews the field of computer system architecture based on my experiences as a participant in the field since the 1950s. It emphasizes concepts that, although they have had a degree of respect and acceptance in the past, have been ignored for many years by computer architects. Some of these concepts have a strong relevance to the current crisis in computer architecture: global virtual memory, functional programming, and data-cognizant control. Many projects and research efforts have contributed to a wealth of knowledge about these forgotten ideas – a gold mine waiting to be tapped and exploited. The time is right to look at them again.

Manuscript received January 2007. This work was supported in part by the U.S. National Science Foundation under Grant CNS-0509386.

J. Dennis is Professor Emeritus at the MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, Massachusetts, USA (e-mail: dennis@csail.mit.edu).

2. Virtual Memory

In the history of computer architecture no innovation has had the impact on programming that virtual memory and paging has had. Two ideas have contributed to the evolution of virtual memory. The better known is from the Atlas machine[15] which introduced the use of a relatively small main memory divided into fixed-size page frames loaded from a backing store on demand by a run-time control program. The need to load a new page was detected by failure to match the page number in a set of tag comparators, one for each page frame.

The second idea is the concept of segment, a contiguous area of memory space allocated to hold a particular data or code object. In the Rice University Computer, *code words* were introduced[14] that defined the limits of a data segment so run-time routines could move segments about in the main memory and provide contiguous space for new data segments.

The use of paging has become universal in commercial computers, and has made the programmer's task far easier than before. Before paging the programmer had to decide which data and code objects should reside in the main memory through the course of program execution, and include code to explicitly move pieces of code and data between main memory and the "backing store" (tape, drum, or disk) to meet the needs of the computation. (These programming gyrations have been reintroduced with the advent of the IBM Cell Processor.)

Paging implements a "virtual memory" in that the programmer writes as though programs and data are laid out in an address space much larger than the physical main memory of the computer. The paging mechanism operates so the "working set" of code and data objects resides in main memory, and those objects not

currently in active use are automatically relegated to the backing store. This makes it possible for the programmer to practice the most fundamental technique of modular programming -- calling a procedure module without the caller being concerned about the modules' ability to access the data it needs to perform its function. In other words, paging hardware has provided the support needed for dynamic management of the memory resource, a feature that relieves the programmer of the task of planning the static allocation of memory. This has become an accepted standard of computer system engineering.

Segmentation has had a less felicitous history. The concept was introduced to permit a data structure to grow (within a data segment) according to the needs of the specific computation being performed, not constrained by the presence in memory of other data objects. The idea of code words from the Rice University machine was adopted in the descriptors of the Burroughs B5000 system[7]. This feature allowed the automatic operating and scheduling program of the B5000 to load into main memory just those code and data segments needed to run the current constellation of user jobs. This opened up a new degree of power in modular programming because data objects (segments) of arbitrary size could be passed to and from a called procedure module.

The Multics computer system[3] carried the idea of segmentation further. Its memory management hardware provided each user process with the capability of linking to as many as 2^{18} data or code segments, each of which could hold as many as 2^{18} words of information. A segment could be created of arbitrary size, and could grow or shrink as computation progressed. Segments were divided into fixed-size pages to make storage management simple, and so only the active pages of segments need be kept in main memory. This effectively implemented a virtual memory space sufficiently large that there was no need for a separate means for accessing "files": A file in Multics is just a segment, so accessing a "file" was just like accessing data in "memory". These features of Multics gave its users a level of modular programming capability that did not exist before, and has not been seen since in commercial systems.

Multics did not however provide the most ideal environment for modular programming for at least two reasons: First, because the linking of segments was done independently for each process, addresses (e.g. representing data objects) could not be passed between processes without losing their meaning. Moreover, if the owner of a process would like it to run continuously, linking to new objects as necessary, it was necessary to invoke mechanisms to unlink unneeded segments from the process – a violation of modularity principles. More relevant to the theme of this article, Multics did not support any means for encapsulating a computation by several processes into a module that could be used as a component of a larger program. Encapsulation could not be applied to parallel programs. Overcoming these defects in Multics remains a challenge for future system architects.

3. *Functional Programming*

The main ideas of functional programming became of interest to computer scientists in the 1960s through the efforts of such workers as Landin, Strachey, Burge[5], and others, largely motivated by the mathematical logic of Alonzo Church[8]. Functional programming is a "declarative" style of expressing computation, in contrast to the imperative style characteristic of conventional programming languages. A declarative style expresses "what is to be done" rather than "how it is to be done". This means that functional languages do not use labels for memory locations, but only as names referring to values. The values to be calculated are defined in terms of intermediate values, which are, in turn, defined by other defined values, and so on, back to the input values for the calculation. Recursion is used as a natural way of defining sets of values generated by a series of similar computation steps, instead of iteratively "updating" loop variables. In this way, functional programs are much closer to the conventional expression of an algorithm in mathematical notation than is a program in, say, Fortran or C.

Early workers in functional programming concentrated on the expressiveness and semantics of functional notations for programs. The functional programming community also studied type systems for programming languages such that one could guarantee that a successfully compiled program would never encounter a type error during execution. These ideas regarding types have finally made it into popular languages such as Java.

One important property of functional programming languages is the absence of "side effects" of procedure calls. In procedural languages such as Pascal, C, and their successors, the procedure body may contain assignments to variables external to the procedure. (In some languages such assignments are limited to assignments to procedure arguments passed "by reference".) In a "pure" functional language such assignments cannot be made. This property gives purely functional languages several important benefits. One is that of "referential transparency" – an expression of the language has the same

meaning (represents the same value) regardless of where it appears within the body of a function. A related property is that of "implicit parallelism" – any set of separate subexpressions of a function body may be evaluated in parallel; that is, their order evaluation does not affect the result. Programming languages that permit side effects forego this useful property, and it is necessary to employ sophisticated analysis techniques to discover sections of code that may be correctly executed concurrently.

An often-mentioned limitation of functional programming is that side effects are used in programs to accomplish tasks that are not otherwise expressible – tasks involving the notion of "state". One class of such tasks arises in situations where a sequence of values is being processed to generate a result sequence, and each element of the result is dependent on all input values that have been seen so far. A program that performs such a task must maintain an internal state that holds whatever information about past input values is needed to calculate future results. A purely functional language can express such tasks if the power of stream data types[11] is included. So far, important languages such as ML and Haskell have not adopted this enhancement.

4. Data-Cognizant Control

Data flow concepts of computation led to a flurry of interest in the 1970s and 1980s. Data flow offers an appealing idea of parallelism: the availability of data initiates action on the data. In particular, the joint presence of two operands can signal the time to apply an operation to two input values. This seems conceptually simpler than the software synchronization gyrations needed to start a third process just when two others have completed subsidiary tasks. Indeed, data flow principles are applied in superscalar processors to indicate when an instruction may correctly issue to a function unit.

The right combination of control and data activated computation in a programming model can yield simplicity, elegance, and other benefits such as avoiding explicit locking and the problems of cache coherence. Two examples of combining control and data flow are part of current practice: One is message-passing where in the reception of a message a new activity is initiated; the other is the remote procedure call that starts a new process and provides its initial data in a single action. The concept can be made more universal with better architectural support.

5. Modular Construction of Software

Modularity in software concerns the ability to assemble large software systems from separately designed software components. Some principles for the practice of modular software construction were laid down many years ago. Parnas[17] identified "information hiding" as an essential element, and Boebert[4] emphasized context independence and the importance of using hierarchy. In Modula 2, Niklaus Wirth[18] introduced the practice of separating module inputs and outputs and precisely specifying their data types.

A weakness of these early ideas about program modules is that they failed to deal with data objects represented as linked structures in memory. Although Parnas understood the desirability of encapsulating representations and operations on data structures, it was only with Simula 67 and Clu that these ideas became incorporated in programming languages, and it became clear that a garbage-collected heap is essential to achieve the full benefit of support for modular software. The advent of Java makes these ideas finally available in a popular programming language.

Recently this author recognized a Secure Arguments principle[10], which requires that a software module not be able to modify input data in any way that would affect operation of another module presented with the same input concurrently. This is clearly important in a system supporting concurrent execution of program modules, where two modules developed independently might operate simultaneously using the same input object. Yet it is also important in sequential programming because it should not make any difference which of two independent modules is executed first when they share only input data.

A summary of these ideas has been presented as six principles of modular software construction[10]:

> **Information Hiding Principle:** The user of a module must not need to know anything about the internal mechanism of the module to make effective use of it.
>
> **Invariant Behavior Principle:** The functional behavior of a module must be independent of the site or context from which it is invoked.
>
> **Data Generality Principle:** The interface to a module must be capable of passing any data object an application may require.
>
> **Secure Arguments Principle:** The interface to a module must not allow side-effects on arguments supplied to the interface.
>
> **Recursive Construction Principle:** A program constructed from modules must

be useable as a component in building larger programs or modules.

Resource Management Principle: Resource management for program modules must be performed by the computer system and not by individual program modules.

Although it seems obvious that the architecture of a computer system would affect the ability of its users to practice modular software construction, computer architects have paid little heed to these principles. A module must be a unit of software that has well-defined interfaces with other modules and implements functionality independent of the context of use. To satisfy the Resource Management Principle, the system must implement resource management rather than permitting modules to make private decisions about use of shared resources. To satisfy the Data Generality Principle, it must be possible to pass arbitrary data structures held in heap storage as module arguments and results. This implies the necessity of a global address space. The Secure Arguments Principle implies that the language in which modules are written should not permit "side effects" through input data.

6. ARCHITECTURE EVOLUTION

Since the introduction of RISC architecture in around 1980, the main driving forces in computer architecture have been performance of programs written for the sequential programming model, and the ability to execute legacy code. This led to the ever-increasing complexity of processor architecture in a quest to execute more instructions per clock cycle. The added complexity usually increased power consumption; that was not an issue because a system would have only one processor chip and would include lots of memory and communication logic that would still dominate the power equation. In fact, any scheme that would utilize silicon area to achieve even a slight (single-thread) performance gain was worth the investment: register renaming and wide instruction windows to increase the amount of instruction-level parallelism the hardware could discover in the instruction stream; logic to by-pass use of the register file for results that could be immediately used by successor instructions; and mechanisms to ensure that results are not committed too early when instructions are issued out of their order in the instruction stream. The schemes included mechanisms such as branch prediction that permitted the processor to continue executing instructions following a branch instruction even though it might turn out that the branch is taken and the speculative execution has no value. This process of buying small performance gains for significant investments of silicon area has now run out of steam.

Users interested in high performance for scientific computations have, since the "Attack of the Killer Micros"[6], relied on the most economical way of achieving supercomputer levels of performance: connecting large numbers of commercial microprocessor chips together and using message-passing models to write parallel applications. This approach has never been acceptable in commercial data processing because running legacy software packages has been a fundamental requirement. Instead, the client/server model allowed users to spread their workloads over multiple machines without disturbing the underlying software paradigm.

Now the tables have turned. The speed of logic in future silicon devices is no longer expected to increase with each new generation of processor chip. The myriads of transistors it is now possible to fabricate in one chip must be put to better use than heroic attempts to improve single thread performance. Instead, the way forward is through concurrency, using multiple threads of computing to carry out multiple activities simultaneously. Power consumption has also become a big issue. Complex superscalar processors are now seen as an exorbitant waste of power and silicon area for the performance achieved. Using multiple copies of simpler processors is clearly the better choice; the pressing question is how to program them.

7. IMPEDIMENTS TO PARALLEL PROGRAMMING

Much has been made of the difficulties of writing parallel programs. Taking a sequential program and finding the parts that may be executed concurrently is a challenging task for a person, and the search for automatic means for uncovering opportunities for parallel computing has had disappointing results. On the other hand, experience has shown that parallel computers can be successfully used: most computing tasks can be analyzed and programmed to exploit the parallel execution capabilities of multiprocessor systems. However, there are at least four issues: First is the programming effort required to coordinate the activities of multiple threads of computing. Second is the cost in performance due to the additional instructions and processor cycles needed to implement the coordination. Third, the programming of thread coordination is rife with opportunities for errors that can lead to data races and nondeterminate behavior. Finally, data structures with shared access from several parts of a program are often a source of complexity because the program parts are running in different addressing environments and

their interaction requires translation of each communication into the address space of the receiver.

These problems all arise from the absence of a satisfactory *program execution model* for parallel computing. The application program interfaces (API) provided by multi-core chip running conventional operating systems and providing library software for parallel computation violate many principles of modular programming. When one attempts to use a parallel program as component of a larger software unit, one finds that:

- Processors are committed.
- Data structures are distributed.
- Files may be used, but only their names may be used in an interface.
- The procedure call is an inadequate module interface, but the best available.
- Side effects are rampant.
- Naming conflicts arise.

A common attitude is that these difficulties can be overcome through improvements in software: better languages, smarter compilers, and advanced run-time systems. Yet even the vendors of the new multi-core chips admit that new programming models are needed for parallel computing and are looking to the research community for ideas.

Thus there is a new search for better ways of expressing parallelism. However, the search often presumes that good solutions will be found in more advanced and sophisticated software tools. I believe this search is futile because the source of the difficulties is the inadequacies of the application programming interface (API) presented by current multiprocessor systems and their operating software.

8. THE OPPORTUNITY

The theme of this article is that the difficulties of programming parallel computations can be eliminated by choosing a good programming model and supporting it with matching system architecture.

> **It is possible to construct systems that exploit the parallelism present in applications and honor all six principles of modular software construction … and they can have superior performance.**

A first requirement is that all threads of control use names (addresses) for objects that have the same meaning. A second requirement is that processing resources not be committed by a program module. Processors must be virtualized so they are available to perform any work that is ready to be done. A third requirement is that the means of thread coordination not present the programmer with choices that can result in needlessly nondeterminate behavior. Means that combine data flow and control flow concepts can provide a solution.

The need for a major change is appreciated by leaders in industry. Dr. John McCalpin of IBM has said[16]:

> **"Hardware must support dramatically superior programming models to enable effective use of parallel systems."**

Workers on shared memory multiprocessor systems have long been concerned with the "memory model" presented by their systems to the user. In contrast, I believe that designing a memory model is solving the wrong problem. Computer systems are built to run programs, and it is the program execution model that determines whether a program behaves correctly according to the semantics of the language in which it is expressed. A memory model is relevant only as a component of a program execution model with which it must be consistent.

9. THE FRESH BREEZE PROJECT

An example of a forward-looking project is the Fresh Breeze Project being pursued by the author and students at the MIT Computer Science and Artificial Intelligence Laboratory. This project is evaluating a multi-core chip architecture that aims to support efficient parallel execution of large programs constructed hierarchically from modular components. A Fresh Breeze computer system will honor all six principles for modular software construction. To achieve this goal the Fresh Breeze architecture will use simultaneous multithreading processors, and provide a global memory consisting of fixed-size chunks that are created and initialized by threads, but become read-only once they are shared with other threads. The Fresh Breeze multithread program execution model is an extension of the Cilk model[12], with hardware-implemented futures to build producer/consumer style compositions of modules, and a special *guard* memory object to provide a transaction capability. A Fresh Breeze program runs with determinate behavior unless it contains uses of guards for transaction processing applications.

The Fresh Breeze architecture provides an alternate machine-level view of computation – one consistent with principles of functional programming: Starting with given data, compute new values and store in fresh memory cells; don't overwrite the given data – it may be in use by other concurrent activities. Achieving this requires the flexible and agile handling of memory and processor allocation that is built into the Fresh Breeze architecture.

Further benefits of the Fresh Breeze system architecture include:

- The Fresh Breeze execution model does not permit creation of cycles of chunks, so reference count garbage collection may be used, realizing rapid, distributed identification of unused memory chunks.
- Conventional files are supported as structured collections of memory chunks, and so may be freely passed between program modules.
- Fine-grain management of threads is supported by an on-chip thread scheduler that provides flexible allocation of processing resources.
- Many functions performed by the kernel of a conventional operating system are performed by Fresh Breeze hardware, eliminating a large source of software overhead.

The Fresh Breeze multithread processor will, we believe, achieve competitive, even superior, performance for the area and power used.

10. Conclusion

The time is ripe for many innovative projects to explore new design spaces for computer architecture. The tools are available for constructing prototypes at modest cost, including the FPGA prototyping board developed by the RAMP Project[1]. A recent Berkeley Report[2] observes that the best choice along the line of simple to complex is not independent of the application domain and context of use, so there are many design points that would be worthwhile to explore. The Fresh Breeze Project makes some radical choices. Other design points may be capable of achieving similar goals.

There is no longer an excuse for computer architecture research to evaluate minimal adjustments to conventional processor designs. Changes are needed in the nature of our computing infrastructure. Let us seize this opportunity to build a new foundation for computing, one that will have the power to change the way we construct software for the better. The coming years should see a blossoming of innovative work in computer system architecture.

Acknowledgment

The ideas presented here have evolved through research efforts with students and colleagues in the Computation Structures Group of the MIT Computer Science and Artificial Intelligence Laboratory, and with professional, academic and industrial colleagues throughout the world.

References

[1] Arvind, K. Asanovic, D. Chiou, J. C. Hoe, C. Kozyrakis, M. Oskin, D. Patterson, J. Rabacy, and J. Wawrzynek. RAMP: Research Accelerator for Multiple Processors - A Community Vision for a Shared Experimental HW/SW Platform. Technical Report UCBCSD-05-1412, University of California, Berkeley, September 2005.

[2] Asanovic, K., et al. *The Landscape of Parallel Computing Research: A View from Berkeley.* Technical Report UCB/EECS-2006-183. University of California at Berkeley, 2006.

[3] Bensoussan, A. Clingen, C. T., and Daley, R. C. The Multics virtual memory. In *Proceedings of the Second Symposium on Operating System Principles*, ACM, 1969, pp. 484-492.

[4] Boebert, W. E. Toward a modular programming system. In *Proceedings of a National Symposium: Modular Programming.* Cambridge, MA: Information Systems Press, 1968.

[5] Burge, W. H. Recursive Programming Techniques. Reading, Massachusetts: Addison-Wesley, 1975.

[6] Brooks, Eugene. Attack of the Killer Micros. Presentation at Supercomputing 89. Lawrence Livermore National Laboratory.

[7] Burroughs Corporation. The Descriptor--a definition of the B5000 information processing system, Detroit, Michigan, 1961.

[8] Church, A.. *The Calculi of Lambda Conversion*. (AM-6: Annals of Mathematics Studies) Princeton University Press, 1941.

[9] Dennis, J. B. Fresh Breeze: A multiprocessor chip architecture guided by modular programming principles. *ACM SIGARCH Computer Architecture News 31*, 1 (March 2003) pp. 7-15.

[10] Dennis, J. B. A parallel program execution model supporting modular software construction. In *Third Working Conference on Massively Parallel Programming Models*, IEEE Computer Society, 1998, pp. 50-60.

[11] Dennis, J. B. Stream data types for signal processing. In *Advances in Dataflow Architecture and Multithreading*. J.-L. Gaudiot and L. Bic, eds., IEEE Computer Society, 1995.

[12] Frigo, M., Leiserson, C., and Randall, K. H. The implementation of the Cilk-5 multithreaded language. In *Proceedings of the ACM SIGPLAN '98 Conference on Programming language design and Implementation. ACM SIGPLAN Notices 33*, 5 (May1998).

[13] Gosling, J., Joy, B., and Steele, G. *The Java Language Specification.* Addison-Wesley, 1996.

[14] Iliffe, J. K., and Jodeit, J. G. A dynamic storage allocation scheme. *Computer Journal 5* (3): pp. 200 - 209, October 1962.

[15] Kilburn, T., Edwards, D.B.G., Lanigani, M.J., and. Sumner, F. H One-Level Storage System. *IRE Trans. Electronic Computers EC-11,* April 1962.

[16] McCalpin, John D. Perspectives on the Memory Wall. Powerpoint presentation, February 18, 2004. IBM, Austin, TX.

[17] Parnas, D. L. Information distribution aspects of design methodology. In *Information Processing 71.* North-Holland, 1972, pp 339-344.

[18] Wirth, N. The development of procedural programming languages: personal contributions and perspectives. In *Lecture Notes in Computer Science, Volume 1897: Modular Programming Languages.* Springer-Verlag, 2000, pp 1-10.

On Reducing Overheads in CMP TLS Integrated Protocols

Radulović, B., Milan; Tomašević, V., Milo

Abstract—*The paper primarily tries to identify the main obstacles for performance and complexity improvements in CMPs (speculative chip multiprocessors) with TLS (thread level speculation). It is focused on an analysis of the integrated speculation and coherence protocols in the state-of-the-art CMPs and identifies four areas where the improvements are promising: hardware overhead, software overhead, bursty traffic, and replacement policy. After an overview of each aspect, some ideas for reducing the identified overheads are outlined. Finally, the paper concludes with a very brief sketch of an innovative proposal which employs the lessons learned the previous analysis.*

Index Terms—*coherence protocols, single chip multiprocessors, thread-level speculation*

1. INTRODUCTION

The rapid technology advances in the past decade resulted in emerging of single chip multiprocessors (CMP). Their wide acceptance depends not only on the efficiency for parallel applications, but also on their ability to execute sequential applications in a cost-effective way. For this purpose, CMPs predominantly employ the thread-level speculation (TLS) technique [1].

The CMP architectures with TLS support can be classified into three main groups. The approaches from the first group are completely oriented towards exploiting speculative parallelism, e.g., Multiscalar (ARB and SVC schemes) [1,2], Multiplex [3], Trace processor [4], SM processor [5], MAJC [6], MP98 [7] and MIT MAP [8]. In these proposals application threads can communicate both through registers and the shared memory. It has been shown that hardware and software support for inter-register communication in these architectures is quite acceptable and effective to provide correct speculative execution. In addition, some of these systems, such as Multiscalar, Multiplex, Trace processor, MIT MAP or SM processor, have sufficient hardware support that enables them to deliver high performance by handling sequential binaries without a need for full recompilation of the source code. However, when these systems run a parallel application or a multiprogrammed workload, large amount of speculative hardware and software support remains unutilized.

The second group includes mostly generic CMP architectures with only minimal support for speculative execution, e.g., Hydra [9], STAMPede [10], and SP [11]. These systems restrict the inter-thread communication to occur through memory only. Studies about impact of communication latency on overall performance of speculative CMPs argued that a fast communication scheme between processor cores may not be required and that inter-thread communication through the memory is fast enough to minimize the performance impact from communication delays [9,10,11,12]. The limitation of inter-thread communication through the memory simplifies the design but the need for source code recompilation is still a disadvantage.

IACOMA [12], Atlas [13] and SpecCMP [14] architectures are the representatives of the third group of CMPs with TLS support. They combine the best features of previous two approaches:

- inter-thread communication both through registers and memory, and operation on sequential binaries without need for source recompilation as in the first group,
- modest hardware support for speculative execution and generic enough CMP architecture as in the second group.

The third group of CMPs with TLS support has simpler design and modest hardware/software support for speculative execution compared to CMPs from the first group.

In speculative CMPs, correct speculation handling is usually integrated into protocols for coherence maintenance [15]. Several kinds of overheads or can be noticed in these protocols such as:

- Hardware overhead,
- Software overhead,
- Bursty traffic on thread commit,
- Inappropriate replacement policy.

Our intention is to propose a speculative CMP architecture similar to the systems from the third group since they have a modest hardware and software support for the speculative execution and, also, with a support for the inter-thread communication both through registers and

Manuscript received December 12, 2006.

Milan Radulović is with T-mobile, Podgorica, Montenegro (e-mail: radulovic_milan@yahoo.com). (contact person)

Milo Tomašević is with School of Electrical Engineering, University of Belgrade, Serbia (e-mail: mvt@etf.bg.ac.yu).

memory. The goal is to alleviate some of these overheads and to improve over existing solutions of the same kind. For that reason, in the following sections a brief analysis of overheads observed in Hydra [9], STAMPede [10], SP [11], IACOMA [12], Atlas [13] and SpecCMP [14] is presented. In addition, SVC scheme of Multiscalar employs an important speculative and, therefore, it is also included in this analysis [2].

2. *HARDWARE OVERHEADS*

IACOMA, STAMPede, SpecCMP, Multiscalar (SVC scheme) and Hydra assign a number of state bits to L1 data cache lines/words as a part of the hardware support for speculative execution. In IACOMA and STAMPede each private L1-cache word and line, respectively, are augmented with additional 6 bits, while SpecCMP has even 8 additional bits assigned to each L1-cache word. Tags of each data cache 32-byte line in Hydra also include additional bits to record the state necessary for speculation. The first two bits, Modified and Pre-Invalidate, augment the basic cache coherence scheme, while the other two sets of bits, Read-by-Word and Written-by-Word, allow the detection of RAW violation using the write bus mechanism. SVC scheme incurs 6-bit overhead plus a pointer for each L1-cache line. The pointer identifies the processor that has the next copy/version, if any, in the Version Ordering List (VOL) for a particular line [2].

SP uses the memory buffer in the run-time dependence checking. Namely, all of the target store entries, from current and the predecessor threads, are stored in the memory buffer. The memory buffer is used as a write-back buffer for the non-target store data. Each entry in the memory buffer includes the address tag and the data field as well as additional bits and bit fields such as: valid bit, alias count, target store distance vector (N bits) for address and target store distance vector (N bits) for data (N is the number of thread processing units) [11].

Also, all speculative CMP architectures exploit different hardware mechanisms for resolving memory data dependences and buffering of speculative states. This support adds on already present hardware overheads.

The disambiguating mechanism in IACOMA is implemented through the memory disambiguation table (MDT) and a related logic that checks for data dependence violations. The MDT is analogous to a directory in a shared-memory multiprocessor. It is a centralized approach that keeps its entries on cache line basis, while the information is maintained on per-word basis. It augments overall bit overhead since each word in MDT has Load and Store bit for each processor. Also, it is possible that the MDT runs out of entries causing a stall of speculative thread while trying to insert a new entry in MDT. By using the values from MDT, the additional Check-on-store logic determines whether any successor thread has performed an unsafe load causing memory dependence violation [12].

The hardware overhead in SpecCMP is caused by complexity of control logic required for operations of the cache controller: ownership probing and data transfer for data forwarding between processor cores, violation detection and state transition. Since the management of speculative state is performed on per-word basis, the increase in area overhead of applied cache directory is significant in comparison to original MSI cache directory. The estimation of the delay of logic on critical paths and additional area overhead caused by added state bits in L1 caches has shown that area overhead occupies more than half of the total delay for many protocol operations. However, the delay caused by accessing and comparing cache tags is higher than the area overhead. The critical path latency is increased by 11% when protocol operations are performed in parallel with tag comparison [14].

The Hydra employs a set of write buffers, rather than the L1-caches, to hold the speculative writes until they can be safely committed into the L2 cache. Hence, the shared L2 cache is guaranteed to hold the non-speculative data only. One write buffer is assigned to each speculative thread. In case when a write buffer has to be drained to L2 cache, the processor core sends the message to the buffer controller to initiate the procedure. There are more sets of buffers than processor cores in Hydra in order to allow continuity of speculative execution when those buffers drain their contents into the shared L2 cache. Although inclusion of write buffers simplifies the protocol, they may become full and stall the speculative threads. Also, a coprocessor is assigned to each processor core to control the thread sequencing The coprocessor has several hardware mechanisms to support speculation and to simplify cache coherence scheme, but it incurs the additional area overhead. It has several control registers, a set of duplicate L2-cache buffer tags, a state machine that tracks the current thread sequence and the interrupt logic to initiate software handlers [9].

The centralized logic called the Version Control Logic (VCL) is applied in SVC scheme as a hardware support for speculation. Each cache line includes a pointer that identifies the processor core that has the next version in the Version Ordering List (VOL) for that line. The VCL uses the bus request, the program order among the tasks and the VOL for appropriate response for each L1 cache when cache misses issue a bus request. Hence, each cache line is updated based on its initial state, the bus request and the VCL response.

The Atlas is a CMP that engages aggressive

speculation techniques to enable the dynamic parallelization of sequential binaries. Thread speculation and data value/control prediction are combined to enable a processor to execute dependent threads in parallel. This architecture is critically dependent on performance of applied sophisticated data value/control predictor. It is implemented together with the global predictor with hardware structures added to each processor core or with modification of already existed hardware support inside each processor core. The inclusion of this mechanism resulted in an area as well as a run-time overhead [13].

3. SOFTWARE OVERHEADS

Hydra uses a coprocessor as a hardware/software interface to control the thread sequencing in the system. These simple "speculation coprocessors" consist of several control registers, a set of duplicate secondary cache buffer tags, a state machine to track the current thread sequencing, and an interrupt logic that can start software handlers to control thread sequencing if necessary. Also, Hydra requires source recompilation, which is a serious problem when source code is not available. The register-level coherence is also handled by a software support, which incurs additional time penalty [9].

Both STAMPede and SP require source recompilation to extract thread-level parallelism. STAMPede uses software speculation handlers and sophisticated compiler technology to support speculative execution [10]. SP architecture heavily relies on compiler to identify speculative threads and to generate an efficient threaded code. It successfully applies both classical and innovative complier techniques for program analysis and transformation in order to exploit more parallelism in programs [11].

4. BURSTY TRAFFIC ON THREAD COMMIT

IACOMA L1 caches work in a restricted write-back mode during the speculative execution and they are not allowed to displace modified lines. However, when a speculative thread acquires non-speculative status, modified lines can be displaced from L1 caches and they switch to write-through mode. When a thread completes and before it commits, any remaining modified cache line is flushed to memory causing the bursty traffic on interconnect. This may increase the time to commit the thread [12]. The same situation appears also in SpecCMP [14].

The STAMPede also performs writes of non-speculative contents on thread completion [10], which results in same problems as in SpecCMP and IACOMA. SP's thread pipelining execution model drains data from memory buffer to the L2 cache during the write-back stage causing bursty traffic [11]. The same action is performed in Atlas on non-speculative thread completion. The write buffer associated to each CPU has to be flushed to the shared L2 cache by broadcasting the data values out to L2 cache and all speculative nodes causing the bursty traffic on an interconnect as well as an increase of thread commit time [12].

5. REPLACEMENT POLICY

The IACOMA speculative protocol allows only non-speculative thread to displace an updated cache word from L1 cache, while any other speculative thread stalls on that occasion. To keep system simple, IACOMA did not include hardware support that allows the committed lines to remain in L1 cache after a new speculative thread starts on a processor [12].

The SpecCMP temporarily holds the speculative data in L1 cache of a speculative thread. When a speculative thread acquires non-speculative status it is allowed to store data to the shared L2 cache and to replace the cache word found either in shared or modified state [14].

A cache line cannot be evicted from the Speculative Data Cache of a processor core in Atlas while that processor runs a speculative thread. In case when a cache line has to be replaced and only if cache lines with active speculation bits are available for replacement, the corresponding speculative thread must stall until it becomes non-speculative. The active speculation bits added to each cache line enable the detection of data dependency violation. If the speculative cache lines are evicted, the processor would not be able to track data dependences anymore and the speculative execution fails [13].

If a speculative thread tries to evict a cache line with the read bits set, the corresponding processor core in Hydra is stalled until the thread becomes either the head thread or is restarted. However, a small victim buffer is added to data cache in each processor core to prevent its stalling until the victim cache is full. The write buffers are added between each processor core and shared L2 cache to collect all writes made by processors during speculative execution. The buffer is drained to the shared L2 cache only when a speculative thread acquires the non-speculative status. They may fill up during speculative execution, so the corresponding threads will be stalled (unless they are restarted) until they become the non-speculative [9].

The STAMPede generates a flush each time a speculative write accesses a dirty cache line. This action is performed in order to ensure that only up-to-date copy of the given cache line is not corrupted with the speculative write. When an epoch tries to replace a speculative cache line from the L1 cache during speculative execution, it is treated as a dependence violation. Two schemes are possible: first, that allows the epoch to proceed and to signal the violation, and

second, that suspends the epoch until it becomes non-speculative and, then, allows the replacement of given cache line [10].

SP uses memory buffers to save store addresses and data for run-time data dependence checking, as well as to save uncommitted produced data. The memory buffer has a fixed size and when it overflows the corresponding speculative thread must be stalled until all of its predecessors are completed. When it becomes the head thread, it obtains non-speculative status and it is allowed to resume the execution. Then, the data from the memory buffer can be saved into the next memory level [11].

6. IMPROVEMENT AVENUES

Based on lessons learned from previous brief analysis, some avenues for an improvement in each of four areas can be summarized:

1. Hardware overhead can be reduced by using L1 data caches to buffer speculative data until they are safe to be saved in the shared L2 cache, and by decreasing the number of additional bits for each cache word in L1 data caches as a support for resolving inter-thread data dependence violations. Also, it would implement simpler hardware mechanisms for thread control and sequencing in each processor core.

2. Software overhead can be reduced if the source recompilation is avoided during partitioning the sequential programs into threads. Avoiding speculation handlers also leads to reducing the software overhead.

3. The bursty traffic during thread commit incurred in most of speculative CMPs causes a delay in issuing a new thread to the core which executed the committed thread. Thread commits can be made more efficient by retaining the modified data in L1 caches until their replacement or intended modification. That way, the delay in issuing a new thread caused by bursty traffic on thread commits can be neglected.

4. Replacement policies applied in current CMP architectures are based on classical temporal history of either cache lines or words that are candidates for replacement and they do not care about their current coherence/speculation states. A new replacement policy should first consider the cache word memory state instead of temporal history of use. However, among the cache words that are in the same memory state the replacement policy can follow classical FIFO, LRU or random policy as a secondary criterion.

7. BRIEF SKETCH OF THE PROPOSAL

With the experience gained from the previous analysis a proposal is made which attempts to achieve a cost-effective solution by removing or reducing the overheads observed in HW/SW support for speculation in current CMP architectures along the recognized improvement avenues. This section very briefly describes the basic elements of the proposal.

1. CMP architecture

Underlying speculative CMP architecture is similar to IACOMA [13], Atlas [14] and SpecCMP [15] and consists of four processor cores, each with private L1 data and instruction caches connected by a snoopy shared bus, while all cores share the unified L2 on-chip cache (Fig. 1). This architecture supports the inter-processor communication both through registers and memory. The hardware support for register communication includes a snoopy shared bus for transferring register values between cores and local scoreboards for keeping the status of data in registers. A hardware mechanism to support thread-level memory speculation is integrated into each processing unit's cache controller.

2. Speculation support

The speculative threads are limited to inner-most loop iterations. The status of non-speculative thread moves from one thread to its immediate successor and so on after a thread completes. It is presumed that threads strictly commit in order to keep the sequential semantics. The thread has to wait to reach the non-speculative status before it can be retired and a new thread to be initiated on the same processor.

The sequential application starts execution on one core, and then, when it reaches a loop entry point the multiple threads are spawned on other cores. Hence, there is always one non-speculative thread, while all its successors are speculative threads. After the last iteration is completed, any iteration that was speculatively spawned after the last one is squashed.

The speculative state is kept in private L1 caches while the shared L2 cache keeps the sequential state (Fig. 1). Speculative data can be committed to the shared L2 cache only after thread becomes a non-speculative. Speculative threads get the required data from appropriate producer thread. In addition, when a successor speculative thread reads a cache word before a predecessor thread writes to the same cache word, the data dependence violation (RAW hazard) is detected. Then, the thread that caused the violation as well as all its successors, that might use data produced by violating thread, has to be squashed and restarted.

3. Thread identification and manipulation

The thread identification in a sequential application is achieved in software during a compilation For this purpose the modified binary annotator based on IACOMA annotation tool is used [16, 17]. The modifications of the IACOMA binary annotator are related to the classification

of writes to the loop-live registers. Those writes can be divided into non-final writes (NFW), final writes (FW) [16], and possibly final writes (PFW) (in enhanced version) [17]. NFW is one that is definitely followed by at least one write to the same register, while FW is definitely the last one. The PFW is one that might be either final or non-final depending on the control flow of the current iteration. The annotator extracts multiple threads from sequential binaries without the need to recompile the source code.

4. Register-level protocols

Two versions of register-level protocol are proposed Snoopy Inter-Register Communication (SIC) protocol [16], and its extended version SIC (ESIC) [17]. The register communication between threads in both of them can be producer-initiated and consumer-initiated.

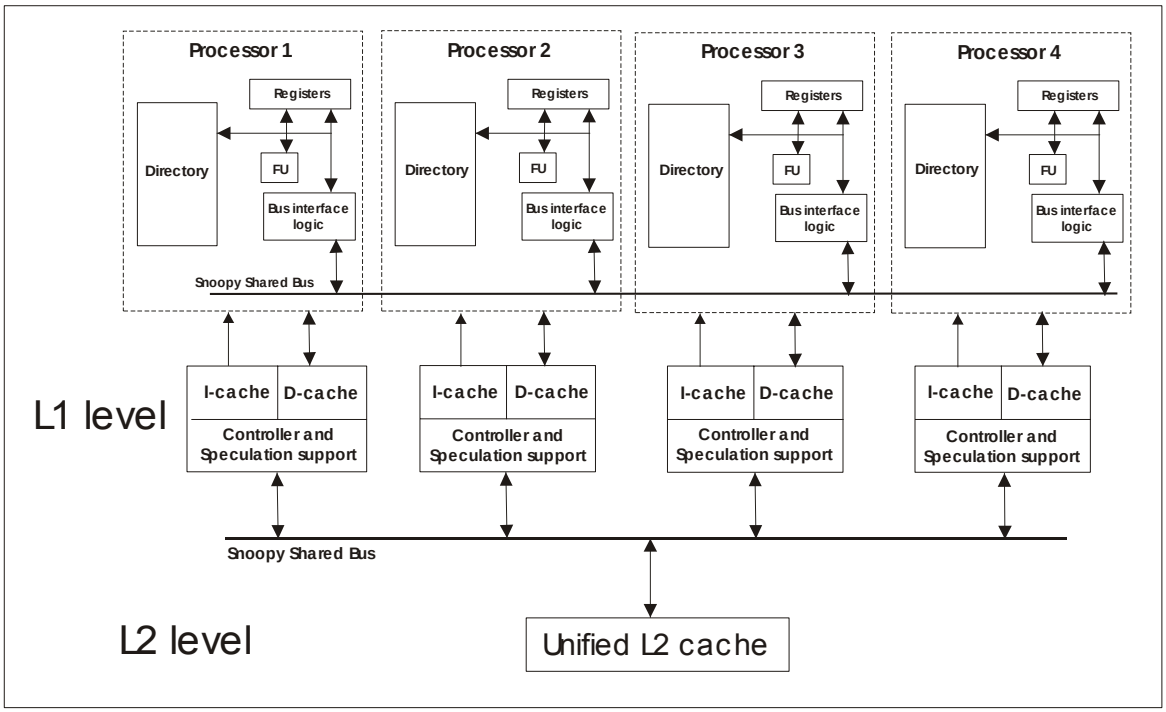


Fig. 1. CMP architecture

A loop-live register value in the SIC protocol can be found in one of the following states: Invalid (INV), Valid-Unsafe (VU), Valid-Safe (VS), and Last Copy (LC). Other registers can be either in Invalid (INV) or Valid-Safe (VS) states. A read miss for a loop-live register causes a consumer-initiated inter-thread communication. All possible suppliers, i.e., predecessors (non-speculative thread or/and earlier speculative threads) participate in distributed bus arbitration which chooses the latest predecessor. If there is no supplier available at the moment, the consumer thread blocks. FW request to a loop-live register incurs the producer-initiated communication by sending the value to the immediate successor thread. If the successor was blocked waiting for this particular value, it continues the execution.

In handling of read misses to a non-looplive registers Read Snarfing is employed in order to decrease a number of misses and to validate the other registers more quickly. Last copy problem is managed by means of a separate state.

However, the SIC baseline scheme does not resolve the problem of consumer thread blocking when requested register value has not been produced yet. Another proposal, the enhanced SIC protocol (ESIC), aims to obtain further performance gain preventing a consumer thread blocking in SIC scheme by aggressively and speculatively forwarding of possibly final register values. Although the ESIC protocol introduces two additional states, Valid-Possibly-Safe (VPS) and Valid-Possibly-Safe-Forwarded (VPSF), hardware and software support for this proposal are very slightly more demanding.

5. Memory-level protocols

The memory communication, buffering of speculative data and data dependence detection in this architecture is performed through L1 write-back data caches with added support for speculative execution. Compared to register dependences it is very difficult to identify memory dependences from the binaries. Therefore, we decided to include a hardware support that is fully responsible for identification and managing of memory dependencies as well as for recovering the execution. The proposal for speculative execution respects the strict sequential semantics both in thread retiring, and a new thread initiation.

This proposal for cache coherence protocol

with TLS support relies on snoopy cache coherence principles [15]. The baseline scheme modifies the snooping bus-based cache coherence protocol to support speculative versioning for proposed CMP architecture.

The proposed integrated coherence and speculation protocol is called Speculation Integrated with Snoopy Coherence (SISC) protocol. The memory state is managed on a per-word basis since previous work has shown that memory state management on a per-line basis could result in false dependence detection that causes the unnecessary threads squash and worse performance [2, 13, 15].

Three variants of speculative cache coherence protocol are proposed: first, a write-invalidate SISC protocol as the baseline scheme, second, an enhanced SISC variant, and, third, a write-update based SISC protocol.

The baseline scheme of SISC write-invalidate protocol uses 11 states to keep track of data status. It enables consumer-initiated inter-thread communication only. The correct value is supplied by most recent speculative thread, non-speculative thread or shared L2 cache.

The committed versions in baseline SISC protocol can remain in the caches after the thread that created the data has been committed. This way, the possible bursty bus traffic that may increase the time to commit the thread and delay initialization of a new thread to that processor is avoided. A modified committed cache word is written back to L2 cache when it is accessed next time either on local thread read/write request or on remote request(s) from other thread(s). The cache words found in stale, shared or modified committed state remains in the same state when a speculative thread reaches non-speculative status. Then, the thread can be retired and a new thread initiated on the same core.

Following the analogy with register-level protocols, since the baseline SISC protocol does not support producer initiated communication on memory level, it is extended with new state Shared-Unsafe (SU) in order to support non-demand inter-thread communication on write hits and misses. If a cache word is either in shared or stale state on a non-speculative write, or in any shared or speculatively modified state on a speculative write, it is forwarded to successor threads on non-demand basis.

Third version of the cache coherence protocol with TLS support in proposed CMP architecture is based on write-update protocol. It uses 9 states and an additional stale bit for each cache word in L1 cache. The inter-thread memory communication in this version can be both producer-initiated and consumer-initiated. In SISC write-update protocol whenever a shared cache word is written to the corresponding L1 cache, its value is updated in the L1 caches of all successor threads that also have a given cache word in shared state.

In all three variants of SISC protocol, an improved replacement policy is employed. It considers the cache word memory state as primary criterion for eviction instead of usual temporal history of reference. The first candidates for replacement are the words in non-speculative states, because the replacement of a speculative word would causes a speculative thread squash and restart. It allows a speculative thread to evict from its L1 cache the modified committed cache words that are only up-to-date copies and that have been speculatively read or written. Also, clean words that have not been yet speculatively either read or write can be replaced. Hence, the processor stall will be avoided and the speculative execution will continue. However, among the clean cache words that are in the same state standard FIFO, LRU or random policy can be considered as the secondary criterion.

8. CONCLUSION

Current CMPs with TLS support have been analyzed for possible improvements. It resulted in identification of overheads in four areas: hardware complexity, software support, bursty traffic on thread commit and replacement policy. Based on the results of the previous analysis, a proposal of CMP architecture with protocols for register-level and memory-level communication is made which tries to reduce those overheads.

This proposal decreases the directory overhead by using fewer bits both in register and memory protocols. Also, SISC protocols use the distributed arbitration for finding the appropriate supplier thread with modest hardware support. The software support doesn't include speculation handlers nor requires source recompilation. SISC protocols retain the non-speculative data in L1 caches on thread commits over time until their replacement in order to minimize the delay in issuing new threads. They also implement an improved replacement policy that reduces thread squashing and stalling. Future evaluation study should reveal the quantitative performance effects of proposed optimizations.

REFERENCES

[1] Sohi, G.S., Breach, S., Vijaykumar, T.N., "Multiscalar Processors", *Proc. of 22nd ISCA*, May 1992, pp. 414-425.

[2] Gopal, S., Vijaykumar, T., Smith, J. E., Sohi, G. S., "Speculative Versioning Cache", *Proc. of the 4th HPCA*, February 1998.

[3] Ooi, C.L., Kim, S.W., Park, L., Eigenmann, R., Falsafi, B., Vijaykumar, T.N., *"Multiplex: Unifying Conventional and Speculative Thread-Level Parallelism on a Chip-Multiprocessor"*, ICS-15, June 2001.

[4] Rotenberg, E., et al, "Trace Processors", *Micro-30*, December 1997, pp. 68-74,.

[5] P.Marcuello, A. Gonzalez, "Control and data dependence speculation in multithreaded processors", *Proc. of the 4th HPCA*, February 1998.

[6] Tremblay, M., “MAJC™: An Architecture for the New Millennium”, *Proc. Hot Chips 11*, Aug. 1999, pp. 275-288.
[7] Matsushita, S., et al., “Merlot: A Single-Chip Tightly Coupled Four-Way Multi-Thread Processor”, *Proc. Cool Chips III Symp.,* Apr. 2000, pp. 63-74.
[8] Keckler, S., et al., “Exploiting Fine-Grain Thread Level Parallelism on the MIT ALU Processor”, *Proc. of 25th ISCA*, ACM press, Jun.-Jul. 1998, pp. 306 -317.
[9] Hammond, L., et al., “The Stanford Hydra CMP”, *IEEE Micro*, 2000, pp. 71-84.
[10] Steffan, J.G., Colohan, C.B., Zhai, A., Mowry, T.C., “A Scalable Approach to Thread-Level Speculation”, *ISCA 2000*, June 2000.
[11] Tsai, J.Y., Huang, J., Amlo, C., Lilja, D. J., Yew, P-C., “The Superthreaded Processor Architecture”, *IEEE Trans. on Computers*, September 1999. pp. 881-902
[12] Krishnan, V., Torrellas, J., “A Chip-Multiprocessor Architecture with Speculative Multithreading.” *IEEE Trans. on Computers,* September 1999, pp. 866-880.
[13] Codrescu, L., Scott Wills, D., “Architecture of the Atlas Chip-Multiprocessor: Dynamically Parallelizing Irregular Applications”, *1999 IEEE International Conference on Computer Design,* October 1999.
[14] Yanagawa, Y., et al., “Complexity Analysis of a Cache Controller for Speculative Multithreading Chip Multiprocessors”, *HiPC - International Conference on High Performance Computing*, December 2003.
[15] Tomašević, M., Milutinović, V., The Cache Coherence Problem in Shared Memory Multiprocessors: Hardware Solutions, *IEEE Computer Society Press*, 1993.
[16] Radulović, M., Tomašević, M., “A Proposal for Register-level Communication in a Speculative Chip Multiprocessor“, *XLIX ETRAN Conference,* June 2005.
[17] Radulović, M., Tomašević, M., “An Aggressive Register-level Communication in a Speculative Chip Multiprocessor”, *IEEE EUROCON2005 Conference,* November 2005.

A Technique for Round-Trip Engineering of Behavioral UML Model Elements

Bojic, M., Dragan

Abstract—*A set of specific requirements, both functional and non-functional, to support round-trip engineering of behavioral UML model elements is derived by studying existing solutions of round-trip engineering in object-oriented modeling. This paper elaborates a use case- and test-driven approach named URCA to satisfy stated requirements. Experimental results gained with a prototype that implements the core technique are also given, along with a discussion of some questions requiring further investigation.*

Index Terms—object-oriented modeling, round-trip engineering

1. Introduction

In contemporary OO modeling tools round-trip engineering of static model elements (classes and their relationships) is supported to a reasonable level. Typical usage scenarios are: (1) building an initial model by reverse engineering program code using a static analyzer and (2) maintaining consistency between the model and the code in both directions. The first scenario is necessary if the application was initially built without modeling.

In this paper, the possible scenario of incorporating behavioral UML modeling elements in the round-trip engineering process is analyzed. The importance of having behavioral aspect of the system modeled has been discussed elsewhere, [6], [7]. Here, it is enough to state that many development methodologies, and specifically the Unified Development Process (RUP) [5] are use-case driven, and use cases are seen as an important integrating mechanism for the model as well as the process.

Proposed in a paper is a method named URCA (Use-case Driven Model Recovery by means of Formal Concept Analysis) that, in author's opinion, satisfy general requirements stated above and impose a minimal overhead to the overall round-trip engineering process. Also presented are experiences with a prototype tool that supports the core technique.

Manuscript received December 24, 2006. This work was supported in part by the German D.A.A.D. international project Software Engineering Education and Reverse Engineering.

D. M. Bojic is with the Electrical Engineering Department, University of Belgrade, Serbia (e-mail: bojic@etf.bg.ac.yu).

2. Problem Statement

Round trip engineering is defined by Wikipedia (www.wikipedia.org) as a functionality of software development tools that provides generation of models from source code (reverse engineering) and generation of source code from models; this way, existing source code can be converted into a model, subjected to software engineering methods, and them be converted back. For a mathematical treatment of this notion see e.g. [4].

Based on the definition, one can identify the following high-level usage scenarios:

1. Building the initial model from the code.
2. Updating the model from the code.
3. Updating the code from the model.

A conceptual toolset depicted in Figure 1 supports scenario-based round-trip engineering. Each element of the toolset will be explained as we elaborate high level usage scenarios.

2.1 Building the initial model from the code

The prerequisite for performing dynamic analysis of an application is to be able to compile it and run it. This is not a serious limitation, especially in the context of framework based development. The initial code is generated in the development environment, and it represents a complete application that can be compiled and executed. In any case, the user should be allowed to initially build only the static model, and to perform a complete reverse engineering once the application can be executed.

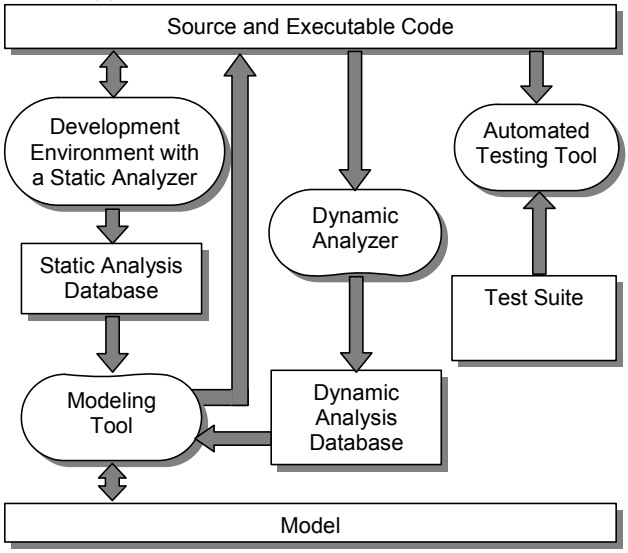


Figure 1: A toolset to support scenario-based

round-trip engineering

Now let us consider the complete process assuming that initially there is no model. The user first builds the executable code of the application. Then he or she devises a set of test cases. Some prescriptions are given in [5], but a general assumption is that for reverse engineering purposes a minimal set of test cases would be devised to relate only to the most important functionality.

Using the modeling tool, the user creates an empty Test Model as prescribed by RUP. For each test case the user updates the Test Model by adding the particular test case and one or more use cases that are traceable to that test case (only model elements for those use cases are entered, and not the relations between them). Using the testing tool, user creates a test script for each test case and enters it in the test suite. The test tool should support test scripting and should also support capture/playback, so initial script is recorded by just executing the application. The dynamic analyzer supervises every execution of the application, extracting and recording data (further discussed in section 3) in the dynamic analysis database.

When all test cases are defined and recorded, the initial model of the application is obtained "with a single click on a button", first performing static analysis to obtain static model elements, then performing processing of execution trace data to obtain behavioral elements. In the context of RUP model template, static elements are grouped in layers and packages and then placed in the Design Model. All other required RUP template elements result from processing of execution traces. These are:

- The Use Case Model, which contains actors, and abstract, concrete and included use cases. For each use case U there is a diagram "Local View" depicting all actors and all other use cases associated with U (Figure 2). There is also a diagram "Global View" representing all actors and all concrete use cases of the system.
- The Design Model, which contains use case realizations; for each realization there is a traceability use case diagram depicting <<realize>> relationship between that realization and the corresponding use case (Figure 3). Also for each realization there is a diagram of participating classes representing all classes (or their roles) that take part in a realization of the corresponding use case (Figure 4). One or more interaction diagrams are also attached to a realization, depicting possible sequences of interaction between objects that belong to that realization. The Process Model, The Business Use Case Model, the Business Objects Model, the Analysis Model and the Process Model are optional in the standard template and will not be further discussed.

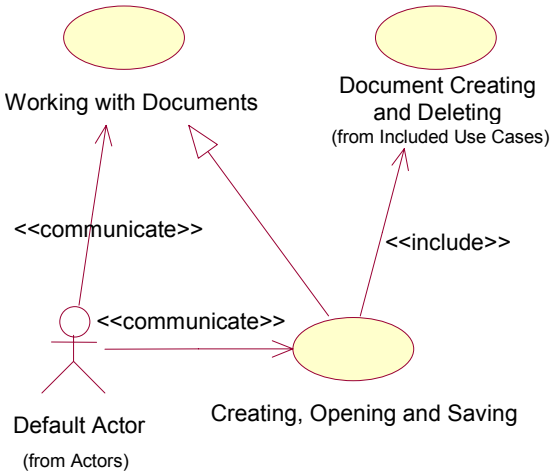


Figure 2: A sample Use Case Local View diagram

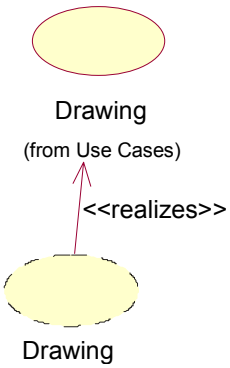


Figure 3: A sample Traceability diagram

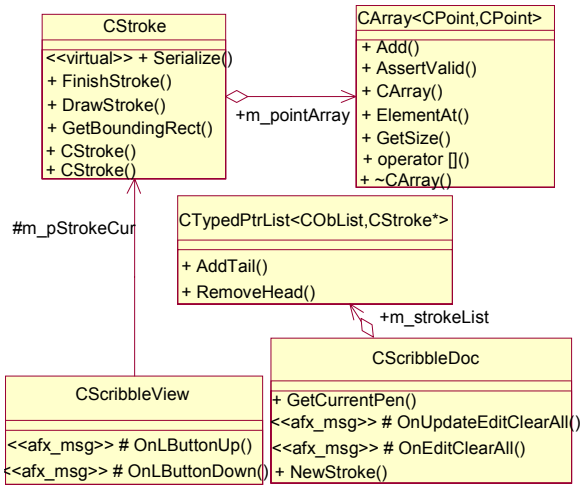


Figure 4: A sample Participating Classes diagram

2.2 Updating the model from the code

There are several concrete scenarios of updating the model from the code:

(1) The user may update only the static parts of the model, because such operation is inexpensive. The behavioral part would, of course, be left unchanged and progressively inaccurate. The inaccuracy could be partially detected by model checking (e.g. an object in an interaction diagram whose class does not exist in the model etc).

(2) Let us first introduce the term "to invalidate a test case". This term is not related to dynamic

analysis data, which are certainly outdated after the first change to the source code, but to failure of a test script to execute entirely. If a change in the source code does not invalidate any test case, and that condition can hold to a certain level of changes, then obtaining an accurate behavioral model is “one click on a button” operation, but of a long duration. The test automation tool should rerun all test scripts from the given test suite to update the dynamic analysis database. The behavioral part of the model is then built in the same way as in scenario 1, as a result processing execution trace. In an Intranet work team configuration, one can even imagine a dedicated server that constantly polls a version control server, builds a new version when it becomes available and executes test cases, thus providing a fresh dynamic analysis database to clients for a transparent update of their models.

(3) If changes in the code invalidate one or more test cases, scenario (2) can be applied to a certain extent with a graceful degradation of model preciseness. That is, test cases that fail will not be incorporated in the analysis, resulting in a model lacking certain interactions.

Ultimately, the Test Model should be updated by replacing or adding test cases and recording new scripts for those updated test cases. Dynamic analysis data should also be updated by running those tests.

2.3 Updating the code from the model

This activity is entirely driven from static elements of the model, and is not influenced in any way by behavioral elements. It should be noted that this process in general does not need any synchronizing source code comments.

Finally, a round-trip engineering feature needs to be automatic and self-configurable to the highest level possible, robust, and of acceptable performance, even transparent to the user. And that may go to the expense of approximating a model as far as it does not interfere with code generation.

3. A STUDY OF EXISTING SOLUTIONS

Let us consider, for example, how round trip engineering feature is implemented in Rational Rose, a popular UML modeling tool. The code analysis is based on static source code analysis, as well as with other main stream OO modeling tools. That means that they can accurate recover only static modeling views, like class diagrams. The main usage scenarios from our problem statement are scenarios 2 and 3, which means that the user is expected to manually create Initial application model. There is, however, support for scenario 1, but it lacks mechanisms for logically grouping model elements in subsystems, packages etc apart from physical structure of the project code. The obtained model is somewhat approximate, but it is not an obstacle because the elements that are used to regenerate code from model are obtained with sufficient correctness. During analysis, a code is populated by comments used for synchronization between the model and the code. Regenerating the model from the code and vice versa is initiated with a single command. Performance is quite satisfactory. Some other tools, notably Together, have even smoother support for round-trip engineering: there is an instant synchronization between the code and the model, without the need for a user to initiate it, and also there are no comments in code cluttering the view.

There exist several approaches to dynamically analyze the software system and produce interaction diagrams in the context of program understanding, reverse engineering, animating and debugging. However, neither of them specifically addresses round-trip engineering. What follows is a brief description of each of related methods, along with a comment on their potential strengths and weaknesses in the context of round trip engineering.

Scene [7] is a tool that enables automatic instrumentation, dynamic analysis and generation of scenario diagrams (in fact object interaction diagrams, in which messages represent object creation, function calls and returns). Scene also enables browsing documentation from a certain points in a diagram (from an object to its class declaration, from a message to a call in a source, etc, to various other documentation). The diagram abstraction is done using static and “physical” criteria: representing modules as objects, and using several operations to divide a diagram that exceeds the maximum size. One of them is call compression, that is, the exclusion from a diagram of all messages from the point of the call to the point of the corresponding return. The Scene method seems to be robust without requiring much or any preconfiguring, but lacks the notion of use case realization, and poses extra requirements on the user interface of a modeling tool.

IsVis [6] is a tool to support browsing and analyzing execution scenarios of large applications. The interesting feature is a support to the user to identify recurring patterns of interaction. Patterns are parts of interaction diagrams that can be specified by replacing some messages with a wildcard to exclude them from a matching process. The user has a visual aid for identifying patterns in the form of global overview of the complete interaction sequence named Information Mural. The tool automatically finds other occurrences of a pattern once the user specifies the first one. The collection of all such identified patterns is termed Program Model, and is used to abstract and structure interaction diagrams. Compared to URCA

approach, IsVis also uses related functionality as a grouping criterion (its patterns roughly correspond to URCA concepts), but uses manual approach to structure the model. Manual approach would make initial reverse engineering more tedious, and also impose a requirement to update the Program Model with each change in code.

SCED [12] tool illustrates an approach to automatically synthesize state diagrams from object interactions. This technique seems potentially suitable for application in round-trip engineering.

Systä [11], [13] presents an experimental Shimba tool that uses SCED and Rigi [9] to combine information obtained from static and dynamic analysis and to create various views in analyzing Java software. The event trace information included in the scenario diagrams is used to build high-level dynamic views applying two techniques: state diagrams and pattern matching. State diagrams are generated automatically for selected objects to view their overall behavior. String matching algorithms are applied to search behavioral patterns from the scenario diagrams, and present them as repetition constructs or subscenarios.

4. THE PROPOSED SOLUTION

URCA method [1], [2] is grounded on formal concept analysis [3] to group functions of the analyzed system in a hierarchical structure using a criterion of common behavior. So-called context relation is constructed on the basis of collected dynamic analysis data (function execution counts). It essentially relates each function F to a use case U if and only if there exist at least one test case that executes F at least once and is traceable to U. Dynamic analysis data are preprocessed in a suitable way:

- All irrelevant functions (essentially those that are not user defined and those that have different execution counts in two different executions of the same test case) are filtered out.
- To account for extended use cases, an operation on test data termed differentiating (essentially subtracting execution counts) is defined. Data for most test cases are for example differentiated with “app. start and exit” one.

On the basis of context relation, the concept analysis derives a concept lattice – a form of an acyclic digraph. Nodes are termed concepts. A concept is generally defined with a set of use cases and a set of related functions, and is interpreted as a UML collaboration element (that realizes those use cases). Remark: in the lattice, only one appearance of each use case (one that belongs to the highest concept) and one appearance of each function (one that belongs to the lowest concept) is taken into account, as prescribed by concept analysis. In a normal situation, there will be at most one use case per concept (in the opposite case, test cases were not discriminatory enough). Concepts that contain an empty set of use cases correspond to common functionality not “visible” from the outside, so one can define one internal function for each of those concepts.

An edge in a lattice between a pair of concepts represents generalize (in case a parent concept contains an abstract use case) or includes relations between use cases that correspond to those concepts. Extends relation is not presented in a concept lattice, but is determined on the basis on test data differing operation.

To reduce the overall number of concepts, each concept that has an empty set of user defined use cases, and the cardinality of the set of its functions below a certain threshold value, can be eliminated from the lattice. In that case, its functions are taken into account in each of successor concepts; its successors are directly related to each of their predecessors.

Figure 5 depicts the concept lattice obtained for Scribble application from Microsoft Developer Network Library. Concepts are named after the corresponding use cases, except concepts marked with #. These are for included use cases and are named after contained functions and their classes. Concept marked with □ corresponds to abstract use case, those without marks to concrete use cases. Twelve test cases in total were executed.

Actors are introduced in a model by relating a set of functions to each actor, in a similar way that IsVis [6] technique does. Manual configuration and update of this information during round trip engineering has been avoided by introducing a default actor, as explained in section 4.1.

Space limitation prevents us from explaining how each element of RUP template is generated from the concept lattice, but the following discussion will concentrate on interaction diagrams.

4.1 Generating interaction diagrams

Profiling information is, of course, not sufficient for this purpose. Therefore, URCA uses program trace in the form of function call tree, as exported from MuTek Bug Trapper tool (BT), and depicted in Figure 6. BT operation relies not on code instrumentation, but on architectural features of Intel processors. BT supports tracing debug versions of library code, providing information about parameter values and time marks, defining tracing on/off triggers in code, etc with negligible performance penalties.

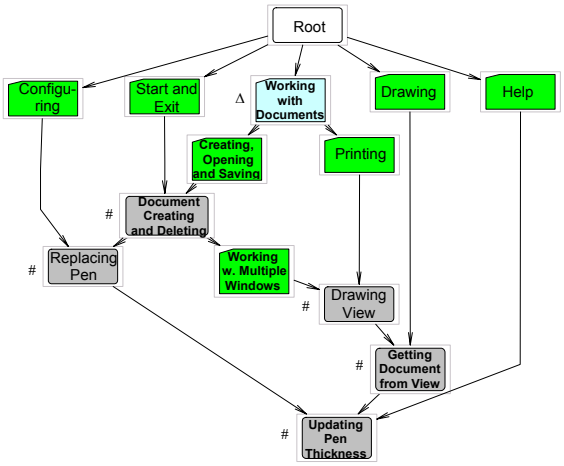


Figure 5: The concept lattice for Scribble

Nothing is required concerning the use of Bug Trapper when writing code. BT needs only the code to be compiled with debugging information included. BT provides a superset of profiler's information, therefore it is sufficient to use BT as a dynamic analyzer. Not all test cases used for constructing a lattice need to be depicted in a model with diagrams, but only those revealing the most interesting interactions.

```
void f2(){}
void f1(){ f2(); }
void f3(){ f1(); }
main() {
    int i;
    f3();
    for(i=0; i<2; i++)
        { f1(); }
    f2();
}
```

```
main( )
  f3( )
    f1( )
      f2( )
  f1( )
    f2( )
  f1( )
    f2( )
  f2( )
```

time flow

Figure 6: A sample program and its call trace

Interaction diagrams, in general, represent objects and their interactions in the form of messages – directed arcs from one object to another. Figure 7 presents an example of an interaction diagram in a form of the sequence diagram, as generated by URCA.

A collaboration diagram is another form of interaction diagram and is automatically generated from a sequence diagram and vice versa by Rose.

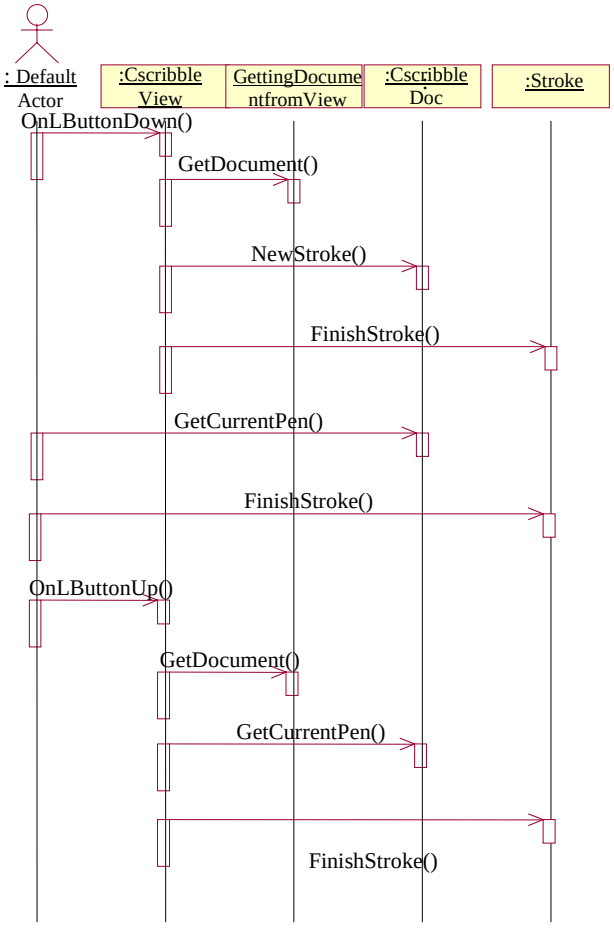


Figure 7: A sequence diagram generated by URCA

Each diagram is connected to one use case realization and the trace of one test case can generate more than one diagram. There is a discussion in [11] about abstracting interaction diagrams in a vertical way, by choosing the level of granularity at which objects are presented, and in horizontal way, by replacing a sequence of messages with a representative.

In the proposed method, diagram objects are either unique class instances, or representatives of related use case realizations (for example "Getting Document From View" object in Figure 7), or actor instances. Messages correspond to function calls, and horizontal abstraction is realized with the following operations on call traces:

• The replacement of references to every function not contained in the lattice by a communication with a default actor instance. Of course, the user can introduce more actors and assign some of those filtered out functions (and other functions) to them to obtain a more precise model. For example, Figure 8(a) and 8(b) represent traces obtained from the one in Figure 6 by filtering out f3() and main(), respectively.

• The elimination of repeating calling sequences. That is, if there exists a pair of nodes in a call tree with the same list of predecessors (starting from the root), the node that is chronologically later is eliminated if it has no

successors. The elimination of all such sequences is done in two passes through the trace, [3]. Figure 8(c) shows a call trace obtained from the one in Figure 8(b), where the calling sequence f1()-f2() has been eliminated.

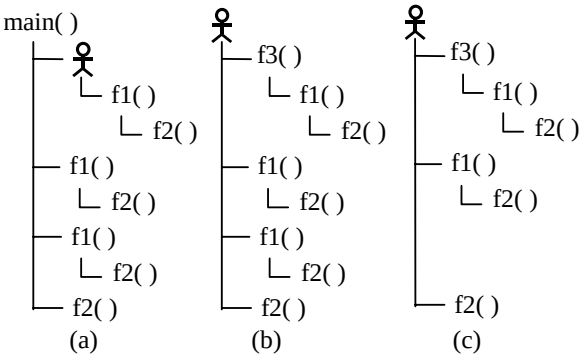


Figure 8: Operations on call traces

The principles of generating interaction diagrams based on call traces and the concept lattice are as follows: We analyze each pair of successive nodes A and B connected with a link in the call trace, and determine their corresponding concepts CA and CB, unless one or both of them are actor instances.

If CA and CB both exist and are the same, we place class instances or class utility instances for A and B on a diagram attached to that concept, and a message link between them. If CA and CB both exist but are not the same, we represent this interaction on diagrams attached to both CA and CB, placing a CB representative on a diagram attached to CA, and a CA representative on a diagram attached to CB. If one of concepts, say CA do not exist, and the other exists, then we place an actor instance A communicating with an object for B on a diagram attached to CB. If both A and B correspond to actors, we place the <<communticates>> relation between them in Global View use case diagram. Note that Rose does not provide special means to represent "included" or "extended" interactions on a diagram.

5. A Case Study

A prototype tool named URCA supports a core process (that is, reverse engineering without test automation). It runs on Windows and requires a Microsoft Visual C++ or Borland C++ profilers and MuTek Bug Trapper tool, along with Rational Rose to view the resulting model. The model is built from a script file that URCA automatically generates.

As a small but illustrative URCA usage case, let us consider the reverse engineering of UML model for Wordpad application from Microsoft development network source code examples. Wordpad consists of 925 functions (excluding MFC framework and other library functions) and about 10.000 lines of source code. 26 use cases could be devised (5 abstract and others concrete) and 22 test cases exercising them, which cover about 68% of functions. The resulting lattice contains 35 concepts, 5 for abstract use cases, 16 for concrete and 15 for included ones. Some use cases were grouped in the same concept for example "Insert line of text", "Delete line of text", and "Edit line of text" either the use cases should be reformulated or the test made more discriminatory). Distribution of functions per "included" use cases is as follows: 0 with 0 functions, 3 with 1, 4 with 2, 1 with 3, and others with more than 3 and less than 27. So we could cut a half of "included" use cases by eliminating those concepts that contain less than 3 functions. The largest potential number of objects on any interaction diagram is 12.

Another URCA usage example was a partial restructuring of XCTL — an application for controlling laboratory equipment for the analysis of semiconductor structures and their pictorial representation at the Max Planck Institute, and also a reference software reengineering project at Humboldt University Berlin. XCTL is based on a very complex problem domain and consists of about 60 custom views and dialogs, and about 44000 LOC in more than 70 C++ source files using Windows native system interface. XCTL GUI subsystem was successfully ported to Microsoft Foundation Classes framework from older technology using URCA methodology.

6. Conclusions

After establishing that a round-trip engineering feature needs to be automatic and self-configurable to the highest level possible, robust, and of acceptable performance, even transparent to the user, the URCA method is proposed in this paper as a possible solution for round-trip engineering of full models according to Unified Process template.

One of the questions that have to be further investigated is a support for consistency checking of manually created model elements, which form an abstraction of a real system (for example, the Analysis Model). The possible way to resolve a problem is to define traceability links to those elements and apply an approach similar to software reflexion modeling [10].

Another issue is defining further horizontal and vertical abstractions for behavioral model elements. The new abstractions would probably include static model elements.

References

[1] Bojić, D., Eisenbarth, T., Koschke, R., Simon, D., Velasevic, D, "Addendum to Locating Features in Source Code", *IEEE*, Transactions on Software Engineering, Vol. 30, No. 2, February 2004, p. 140

[2] Bojić , D., "An Approach to Reverse Engineering of Use Cases", PhD Thesis, *University of Belgrade, Faculty of Electrical Engineering*, Bulevar kralja Aleksandra 73, 11000 Belgrade, Yugoslavia, 2001.

[3] Burmeister , P., "Formal Concept Analysis with ConImp: Introduction to the Basic Features", *Arbeitsgruppe Allgemeine Algebra und Diskrete Mathematik,*

Technische Hochschule Darmstadt, Schloßgartenstr. 7, 64289 Darmstadt, Germany, 1998
[4] Henriksson, A., Larsson, H, "A definition of round trip engineering", Dept. of Computer and Information Science, Linköpings Universitet, SE-581 83 Linköping, Sweden, 2003.
[5] Jacobson, I., Booch, G., Rumbaugh, J., "The Unified Software Development Process", *Addison-Wesley*, 1999
[6] Jerding, D, Rugaber, S., "Using visualization for architectural localization and extraction", *IEEE*, Proc. of the 4th Working Conference on Reverse Engineering (WCRE'97), pages 56–65.
[7] Koskimies, K., Mössenböck, H. "Scene: Using scenario diagrams and active text for illustrating object-oriented programs", *IEEE*, Proc. of International Conference on Software Engineering (ICSE '96), pages 366–375.
[8] Koskinen, J., Kettunen, M, Systä T, "Profile based approach to support comprehension of Software Behavior", 1st int. conf on program comprehension, Greece, June 2006, pp. 212–224.
[9] Müller, H., Orgun, M., Tilley S.R., Uhl J.S., "A Reverse Engineering Approach To Subsystem Structure Identification", *Software Maintenance: Research and Practice*, 54., December 1993, pp. 181-204.
[10] Murphy, G., Notkin, D., Sullivan, K., "Software Reflexion Models Bridging the Gap between Source and High-Level Models", *ACM*, Proceedings of the ACM SIGSOFT '95, pp. 18-28.
[11] Systä T., "On the Relationships between Static and Dynamic Models in Reverse Engineering Java Software", *IEEE*, Proceedings the 6th Working Conference on Reverse Engineering (WCRE99), 1999, pp. 304-313
[12] Systä, T., Koskimies, K., "Extracting state diagrams from legacy systems", ECOOP '97. pp 110-114.
[13] Systä T., "Understanding the Behavior of Java Programs", *IEEE*, Proc. of the 7th Working Conference on Reverse Engineering (WCRE 2000), ,http://www.cs.tut.fi/~tsysta/publications.html

Dragan M. Bojić (M'93) received a PhD degree in electrical engineering and computer science from the University of Belgrade in 2001. He is an assistant professor at the Faculty of Electrical Engineering, University of Belgrade. His research interests focus on software engineering techniques and tools, and e learning.

Support for Knowledge Tests: Brief Summary of Regulations and Software

Bošnjaković, M., Andrija; Tartalja, I., Igor; and Protić, Ž., Jelica

Abstract—Intention of this paper is to brief the reader with overall status of the support for knowledge tests in contemporary software. Need and multiplicity of ways to ask a question and to administer a knowledge test have lead to development of numerous systems with different output formats, thus pushing the need for standardization to front. In order to classify relevant terms, two main criteria are taken into account – whether a term refers to a regulation or to a software product and term's generality. Additional criterion is applied to classify products: free of charge (even open source) or commercial. Examples are given for all of the classes constructed using these criteria. Also provided are list of further readings and brief comparison of relevant features in most prominent software tools, based on the proposed question-type taxonomy.

Index Terms—*CAA, IMS QTI, knowledge test, LMS, question, quiz, SCORM, VLE.*

1. Introduction

REGARDLESS of the field, need for assessment of human resources is ubiquitous within academia and industry. During this process, psychological, especially knowledge, tests are often used as a time-saving tool. There are three common phases during their life cycle: assembly, administration, and scoring. The term assembly denotes creation and/or selection of items for a test, administration is presentation of some or all items contained in the test to the examinees, while scoring is associating some, most often numerical, data with examinees' responses. The concept has been basically the same for decades, what changes over years is the increase in use of computers to automate the process.

One of important parts of e-learning is knowledge assessment, which belongs both to education and psychology. Within this context, tests are often referred to "knowledge tests" or "quizzes", items are referred to as "questions" or "problems", and examinees are referred to as "students" or "learners". In this paper, the first of the two options will be used for each of three terms.

This paper is created as a first part of broader survey conducted at School of Electrical Engineering at University of Belgrade during research and development that resulted in the assessment software called *test* [1], which is made up of modules that help automate described phases of knowledge test's life cycle.

Manuscript received December 15, 2006.

A.M.Bosnjakovic, I.I.Tartalja, and J.Z.Protic are with the School of Electrical Engineering, University of Belgrade, Serbia (e-mails: andrija@etf.bg.ac.yu, tartalja@rcub.bg.ac.yu, jeca@sezampro.yu, respectively).

2. Problem Statement

There are many ways to create and to ask a question. There are many ways to select questions for a test and decide on the order of their appearance. There are numerous related software systems in existence, only simple listing of which would take a lot more than one journal paper. That is why there are many question types available in the existing software, some of which were impossible to use in a paper-and-pencil test, mostly due to their interactivity. Software products that support creation of diverse question types and various test assembly and/or administration methods are producing custom output, which can hardly be used by some other similar tool. This plethora of output formats caused the situation where most tools are not interoperable, which further lead to introduction of standards and specifications that regulate format of questions and tests (e-learning content in general).

Primary goal of this paper is to briefly inform the reader about key products and regulations in the field of e-learning and/or computer-based knowledge testing, rather than to provide in-depth analysis of subtopics. Additional goal is to compare support for most frequent question types in related software. Although important, test administration features will be only mentioned, while scoring will not be discussed at all in this paper, as it is most often trivial for knowledge tests.

3. Organization of the Presentation

This section lists criteria used to classify terms in this presentation, and then lists appropriate classes under meaningful combinations of criteria values, used to organize the paper.

3.1. Criteria

One has to distinguish a term that refers to a regulation (a standard or a specification) from a term that refers to particular software that is compliant or not to a regulation (criterion: *term nature*). Furthermore, both standards and tools can be focused only at questions and tests or to learning content in general (criterion: *generality*). Finally, software tools and systems can be either commercial or free to use and modify (criterion: *price*).

3.2. Classification

The first criterion classifies term in question as either *software* or *regulation*. These two classes are further divided using the other two criteria.

Applying the generality criterion to regulations (standards and specifications) classifies them to those specialized only to question and test representation (class: *specialized regulations*) and to those that cover more general area of describing learning content (class: *general regulations*). Applying the generality criterion to software products introduces other two classes. Tests can be created, previewed and/or administered through a Computer Aided Assessment (CAA) tool, specialized for one or more of these actions (class: *CAA software*); frequently, such tools have support for item parameters and their analysis, which are not as important to educators as to psychologists and psychometricians. Another common way is to use assessment component of e-learning software systems, which are most often called Virtual Learning Environment (VLE) or Learning Management System (LMS). These systems are designed to help instructors (teachers or skilled professionals) in the management of educational content for their learners (students or company employees) and help learners to use the content accordingly to their needs. Within scope of this paper, the term VLE will be used when referring to this kind of software (class: *VLE software*).

Finally, the price criterion is applicable only to software products, as regulations are inherently open to public and ones described in this paper are freely available. Software products can be further classified as *commercial* and *free*, which yields total of 4 classes pertaining to software.

4. Regulations and Example Software

This section introduces the most prominent relevant regulations and software.

4.1. Specialized standards and specifications

Regulations that focus only on question and test representation are almost always a subset of more general documentation.

4.1.1. IMS QTI

IMS Question and Test Interoperability [2] is a specification that was introduced and is constantly developed by IMS Global Learning Consortium (IMS GLC). It is a specification for a standard way of sharing assessment data, such as description of questions and tests and reporting of results. QTI deals with the structure of assessment related information rather than the way it is going to be rendered using some particular software. The QTI specification is now being implemented within a number of assessment systems and virtual learning environments. IMS GLC is maintaining a lot of other specifications related to learning content [3]. Although this specification is quite mature (the current version is 2.1), support for it in current software products is at a relatively early stage, as most products that support QTI are compliant with previous version (1.2) and many of the tools are still under development.

4.2. General standards and specifications

There are several regulations that refer to format of e-learning content. It is worth noting that these are neither complementary nor mutually exclusive.

4.2.1. SCORM

Sharable Content Object Reference Model (SCORM) [4] was created in 2000, as a result of cooperation between government, academia, and industry and is maintained by Advanced Distributed Learning (ADL), which is a part of U.S. Ministry of Defense. It consolidates other regulations, such as IMS specifications [3], AICC recommendations [5], and IEEE 1484 standards (LOM, ECMAScript API, Data model) [6], into unified reference model that enables interoperability, accessibility, and reusability of Web-based learning content. Among other things, it covers packaging of e-learning content (Content Aggregation Model) and the way data are exchanged at runtime between the content and software using it (Run-Time Environment). Similarly to QTI, it does not imply content rendering (unless the content itself contains such information), as it only defines rules on e-learning content storing and exchanging. ADL certified 178 products to be compliant to SCORM 1.2 or SCORM 2004 [7]. This growing number and agreement between ADL, IMS and IEEE [8], further establish SCORM as the industry standard.

4.3. Commercial VLE software

This class includes VLE software that is not freely available, i.e. that one (an institution or an individual) must pay for in order to use it.

4.3.1. Blackboard and WebCT products

Blackboard Inc. is a company that offers several commercial software products. Besides VLE software, the company offers tools that manage transactions related to functioning of an educational institution. After acquisition of the rival WebCT company, Blackboard Inc. became the leading provider of e-learning solutions in the USA, in terms of total users. Having in mind that the given software's e-learning features do not differ substantially from the ones found in other similar software [9], Blackboard software stands out mostly with its transactional capabilities.

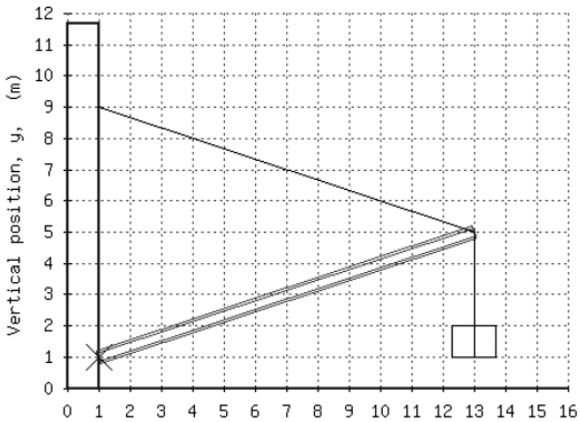
4.4. Free VLE software

Global trend of more and more really usable free and/or open source software impacts this kind of software like all other types. That is why majority of examples presented are in this section. It is worth noting that, besides the examples given here, there are several other solutions in this class, each having respectable list of implemented features and growing user community.

4.4.1. Moodle

Designed by Martin Dougiamas, this open source system has been under constant development for several years, with participation from

A crate with a mass of 155.5 kg is suspended from the end of a uniform boom with mass of 89.5 kg. The upper end of the boom is supported by a cable attached to the wall and the lower end by a pivot (marked X) on the same wall. Calculate the tension in the cable.



A crate with a mass of 177.5 kg is suspended from the end of a uniform boom with mass of 88.5 kg. The upper end of the boom is supported by a cable attached to the wall and the lower end by a pivot (marked X) on the same wall. Calculate the tension in the cable.

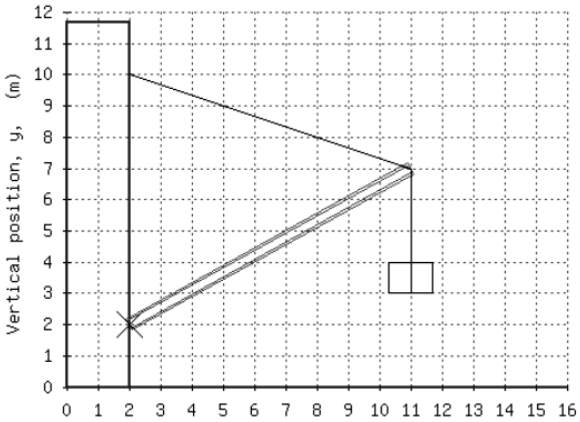


Figure 1. Two versions of the same question. The values of parameters differ in each version of a same question, as they are randomly selected from predefined ranges. Question text and associated image are consistent with these parameters in each version. System uses predefined formulas to automatically generate the result based on the concept that a student must comprehend in order to answer correctly and, optionally, results based on frequent concept misunderstandings. It compares student's response with these result(s) and provides appropriate feedback. This approach "encourages collaboration between students on a conceptual level, but prevents blind copying of answers" [13].

numerous community members and supported by Moodle.com company. Original idea beyond this system is constructivism – approach to learning where knowledge is actively constructed by the learner, not passively received from the environment [10]. The name of the system stands for Modular Object-Oriented Dynamic Learning Environment, which reflects its key attributes – modularity and ease of extension and/or customization. There are more than 100 modules which add some functionality to the system, and the usage statistics [11] are impressive: approximately 50,000 downloads during one month and nearly 20,000 known installations with total of nearly 8 million users and nearly 6 million quiz questions. Information on best practices can be found in [12] and at the Moodle.org community discussion groups. Ease of use and constant improvements make this system the first choice for most educators.

4.4.2. LON-CAPA

Nearly two decade long development of this system is lead by LON-CAPA Academic Consortium, headed by Michigan State University and University of Illinois at Urbana-Champaign. LON-CAPA stands for the LearningOnline Network with Computer-Assisted Personalized Approach, which denotes software's key features: distributed content repository and personalized questions (Figure 1). Moreover, system is utilizing “one source-multiple target” concept, rendering the questions appropriately for the given output format (desktop computer, PDA, bubble-sheet, etc.). Parts of this system are in use for more than a decade, and the distributed repository holds almost 250,000 resources, 80,000 out of which are questions (as of Fall 2006). LON-CAPA related work has produced a long list of publications [14]. Due to its personalized approach, it is best suited for sciences such as physics and mathematics, where concept comprehension is more important than plain memorization of facts.

4.4.3. ILIAS

The name stands for Integrated Learning, Information and Co-Operative Work System, which is being developed since year 1997 by University of Cologne. ILIAS supports a lot of different authentication methods and course design based on learning objectives. Data in different formats ($LaTeX$, OpenOffice.org, etc.) can be imported into the learning content database. ILIAS is also known as the first free SCORM conformant software [15]. Also interesting are extended forms of most frequent question types; for instance, short answer questions may have associated a set of terms that helps automate scoring – student's answer is compared against this set. This system is interesting to users having lots of existing content in popular formats; also, institutions having user authentication system for other purpose can use it to authenticate ILIAS users.

4.5. Commercial CAA Tools

The history of specialized assessment software is long in terms of computer science, and there are many examples. Here are presented only two representative products.

4.5.1. Questionmark Perception

Questionmark is a privately owned company from Norwalk, CT, USA, and have started developing CAA software in year 1988. The company offers a set of test assembly, administration, and reporting tools. Perception Authoring Manager, specialized in question creation and test assembly with support for most frequent question types, has feature-rich question editor and an option to

Question type	Brief description	1	2	3	4	5	6	7	8
multiple choice single response	question text (stem) and several answers (alternatives), one of which is correct, one can be selected	ü	ü	ü	ü	ü	ü	ü	ü
multiple choice multiple response	question text (stem) and several answers (alternatives), zero or more are correct, more than one can be selected	ü	ü	ü	ü	ü	ü	ü	ü
term comparison short answer	student provides a short text answer, and that text is compared against one (or more) possible correct answers	ü	ü	ü	ü	ü	ü	ü	ü
regular expression short answer	student provides a short text answer, correct answer is formed using regular expressions	û	û	ü	ü	ü	û	û	û
cloze (fill in the blanks)	question text contains empty text fields for the user to fill in (fields are most often treated as short-answer questions)	ü	ü	ü	ü	ü	ü	û	ü
cloze (general)	contains several questions of various formats embedded within the text of one composite question	û	û	ü	û	ü	û	û	û
matching items	student is presented with two lists of terms, and should match terms correspondent in a given context	ü	ü	ü	ü	ü	ü	û	ü
numerical	short-answer question that expects a numerical value instead of a word, correct answer is defined by range	û	ü	ü	ü	ü	ü	û	û
calculated (simple)	similar to numerical type, values in question text and correct answer are parameters that vary in each instance	û	ü	ü	ü	û	û	û	û
calculated (complex)	extension of simple calculated type, values in question text and associated pictures consistently vary in each instance	û	û	û	ü	û	û	û	û

Table 1. Overview of most often supported question types with their brief description. Numbers in the heading row pertain to: (1) Blackboard (all products), (2) Web-CT (all products), (3) Moodle 1.6.3, (4) LON-CAPA 2.1, (5) ILIAS 3.7.0, (6) Perception 4, (7) FastTEST Professional 2, and (8) Hot Potatoes 6.

import questions in IMS QTI format. Along with other tools, such as connectors to SCORM conformant, Blackboard or WebCT systems, Perception represents a rounded set of tools, serving more than 15,000 users who author content and millions of users who use it in some way. The company is providing a list of white papers [16] that contain information useful both for introductory and in-depth reading on assessment. Perception is set of feature rich software tools, useful to all users who want to create and/or modify SCORM, QTI and MS Office content.

4.5.2. ASC products

Assessment Systems Corporation (ASC) was founded in 1979 by David Weiss, psychometrics professor at the Minnesota State University; company's products of most relevance to this paper are the historical DOS software MicroCAT, and its successors, FastTEST Professional and FastTEST. The first two support both computerized adaptive testing (CAT) and paper-and-pencil tests, while FastTEST is focused only at the latter. CAT is the kind of testing where the sequence of questions adapts to the examinee's previous responses. Along with other features, MicroCAT has support for MCATL (Minnesota Computerized Adaptive Testing Language), an authoring language specifically designed for specifying adaptive tests. MicroCAT was the first software to support CAT. ASC provides software and books that cover many areas of computerized testing [17]. FastTEST Professional offers different ways and precise control of test administration, as well as support for most frequent question types.

4.6. Free CAA Tools

Although they are not yet mature to be compared with commercial systems, some free assessment software tools are worth discussing, even at this stage of their development.

4.6.1. Hot Potatoes

Since 1997, development team affiliated with the University of Victoria Humanities Computing and Media Centre is working on this set of tools, use of which is free for most non-commercial purposes [18]. All the tools produce interactive educational exercises viewable in most browsers. The package contains several separate tools for different types of learning material (specific purpose enclosed in parenthesis): JCloze (fill in the gap exercises), JCross (crosswords), JMix (jumbled-sentence exercises), JMatch (matching and ordering exercises), and JQuiz (quizzes made of some of most frequent question types). All of the exercises use JavaScript and HTML for their functionality, and the content can be made to comply with the Dublin Core metadata standard [19]. Having in mind supported types of exercises, these tools are most suitable for language learning and purposes where assessments should preferably be in some presentational format attractive for younger examinees.

5. FURTHER READINGS

The above list is only a top of an iceberg, when compared to constantly increasing list of relevant products. For more information on terms described in this paper, interested reader is first directed to their respective sites. Regarding standards and specifications, further readings include already mentioned AICC recommendations and

white papers [5] and IEEE 1484 standards [6]. This list is rounded with efforts conducted by CETIS [20] and JISC [21]. Detailed description of most contemporary VLE software, along with option for direct comparison of several systems' features at a time is available at [9].

6. Discussion of Approaches

Although it should not be the sole criterion when choosing a system for question authoring, it is good to know what question types are available in which software. Table 1 provides very coarse overview, as only the most frequent question types are listed. Most of the tools have some kind of expansion possibilities, so that new or existing question types can be easily added. Moodle and LON-CAPA are good example: both of them utilize Java molecular editor, a graphical editor where a student can construct molecular structure [22]. Once completely drawn, this structure is converted to a string, which is further processed as a response to short answer question.

Regarding support for test administration, most common features supported in CAA and VLE software are: defining the time during which a test will be available, randomization of questions within a test and answer options within a multiple choice question, feedback after a question is answered and/or a test is completed, etc.

Most of the described software products support Internet/Web based or computer/desktop based testing. There is also an approach that is somewhere in between this two concepts – use specialized desktop software to create questions and tests, while using Internet only to deliver the content to the public. One example of this approach is already mentioned *test* software [1]. An interesting feature of this software is automatic test assembly, based on predefined question selection criteria, such as question difficulty, number of questions in given category, etc.

7. Conclusions

Table 1 indicates that VLE software has become comparable with specialized CAA tools, in terms of diversity of supported question types. Communities gathered around open source solutions are constantly contributing to adding new features to these solutions, which further leads to the fact that most features in free VLE software are at least comparable to ones of commercial VLE software.

Free VLE software will probably substantially increase its part in the e-learning market share, as all the crucial components are beginning to fit into place: quantity of available content is increasing due to standardization, features are improving day by day, and there are a lot of small privately owned companies that support developers and offer user support and/or training for free VLE software.

Diverse question types supported in current software contribute to better assessment of student's knowledge. However, these types are only a part of a much larger list given in [23], which opens horizons for future development. Having in mind easy extensibility of the most of relevant software, it is only a matter of time until all existing and, perhaps, some new question types will be implemented and made available, thus boosting the shift already present in education: from traditional testing towards Internet based testing.

Acknowledgment

The authors would like to thank Professor Veljko Milutinovic for his continuous support and encouragement to publish the results of their work.

References

Unless otherwise noted, URLs given in this section are last retrieved on December 15, 2006.

[1] Protic, J., Bojic, D., and Tartalja I., „*test*: Tools for evaluation of students‘ tests - a development experience“, *31st ASEE/IEEE Frontiers in Education Conference*, Reno, Nevada, USA, 2001, pp. F3A.6-F3A.12.
[2] IMS Global Learning Consortium, “IMS Question & Test Interoperability Specification," http://www.imsglobal.org/question/, 29.9.2006.
[3] IMS Specifications, Specifications, http://www.imsglobal.org/specifications.html, 12.12.2006.
[4] Advanced Distributed Learning, SCORM 2004 3rd Edition, http://www.adlnet.gov/scorm/index.cfm, 12.12.2006.
[5] Aviation Industry CBT Committee, Index of Available AICC Publications, http://www.aicc.org/pages/downdocs-index.htm.
[6] IEEE Learning Technology Standards Committee, http://ltsc.ieee.org/.
[7] Advanced Distributed Learning, SCORM Certified Products, http://www.adlnet.gov/scorm/certified/index.cfm, 5.12.2006.
[8] Advanced Distributed Learning, "ADL Reaches Agreement with IMS and IEEE on Use of Specification/Standards Work in SCORM® 2004 3rd Edition," http://www.adlnet.gov/news/articles/384.cfm, 16.11.2006.
[9] EduTools Course Management System Comparisons, http://www.edutools.info/static.jsp?pj=4.
[10] Dougiamas, M., "A Journey into Constructivism," http://dougiamas.com/writing/constructivism.html, November 1998.
[11] Dougiamas, M., et al., "Moodle statistics," http://moodle.org/stats/.
[12] Cole, J., "Using Moodle," *O'Reilly*, Cambridge, MA, USA, July 2005.
[13] LON-CAPA Academic Consortium, "Examples of Individualizing Problems," http://www.loncapa.org/problems.html.
[14] LON-CAPA Academic Consortium, "Publications connected to the LON-CAPA project," http://www.loncapa.org/publications.html.
[15] Kunkel M., et al., ILIAS History, http://www.ilias.de/ios/info-e.html, 6.9.2006.
[16] Questionmark, White Papers, http://www.questionmark.com/us/whitepapers/index.htm.
[17] Assessment Systems Corporation, http://www.assess.com/.
[18] Hot Potatoes, http://hotpot.uvic.ca/.
[19] Dublin Core Metadata Initiative, http://www.dublincore.org/.
[20] Centre for Educational Technology & Interoperability Standards (CETIS), http://www.cetis.ac.uk/.
[21] Joint Information Systems Committee (JISC), http://www.jisc.ac.uk/.
[22] Ertl, P., "Java molecular editor," http://www.molinspiration.com/jme/doc/index.html, October 2004.
[23] IMS Global Learning Consortium, IMS QTI Implementation Guide, http://www.imsglobal.org/question/qtiv2p1pd2/imsqti_implv2p1pd2.html, 14.10.2005.

Efficient Development and Maintenance of Enterprise Information Systems in Multicultural Developer Environment

Milovanović, M., Miloš; Milutinović, M., Veljko

Abstract— Development of enterprise systems has following characteristics: deadlines are always too short, there is no final specification, requirements change in the development phase and nobody knows the whole business process exactly (at least those who know the business process don't know how to specify software which would support that process). On the other side, developers in big international corporations come from different countries; have different education and culture background, as well as their way of software development. In such environment it is extremely important that each part of the system is very flexible, generic and parameterized so runtime changes can be handled instantly. This paper gives a view on serious problems in the development process and the methodology for effective problem solving. There are several frameworks that help developers in building the enterprise systems, but if a developer is not aware of these problems, framework cannot prevent him or her from making the similar mistakes. Methodology presented in this paper is completely independent from the technology.

Index Terms—Information systems, development process, MDD, OOP, OOD, generic applications

1. INTRODUCTION

DEVELOPMENT of enterprise information systems is a challenging job with variety of problems which can appear anytime. For example, web portals, which are very popular nowadays, can be extremely difficult to develop because they should be often integrated with some old legacy software system. Usually nobody knows how that legacy software works exactly, which makes a lot of trouble to developers.

The biggest problem in the development of web portals is that they are, by default, huge; so, at the first sight, they can be developed very fast and a lot of parts can be developed in parallel and later on easily integrated. The following sentence is very popular in the management population, but each developer should be alarmed when he or she hears something like it. .

"Development of that software is **EASY**; it is very **SIMILAR** to something we did before; we should **ONLY** add some **SMALL** changes."

Magic four words and its usage. The sentence given above and its variations are frequently used by the managers, but developers should be suspicious when they hear them. Whenever you hear a word EASY, SIMILAR, ONLY or SMALL, it is better you look closer into your software specification (if it exists).

The second biggest problem is that nobody knows all details about the business which should be supported with the web portal. This can cause many changes in the system requirements in the development phase. This usually happens in the presentation layer, when some manager or tester sees that something is displayed in the wrong place, or some functionality should be disabled in some cases Bearing this in mind, we can conclude the following:

"Everything that is displayed on the screen should be parameterized because there is a high probability that it can be changed."

A simple example is when application is developed in English language and later it is required that application has Multilanguage support.

The third very important problem is a communication problem between users – managers – developers. This problem appears because in big multinational companies there are people from different countries, with different education and culture. This can cause significant problems in communication. Unified modeling language (UML) tries to resolve these problems but still it does not solve all of them.

In the environment like this, a developer is forced to write highly parameterized code, which can be easily customized during the whole software lifecycle. This paper presents the methodology for effective software development,, which is highly flexible, generic, reusable and above all, it is extremely easy to build the new software parts on top of the old legacy software systems. This methodology makes it easy to build high quality software on top of software and databases with significant design problems.

2. *Significant Problems In Development Of Enterprise Systems*

This chapter presents characteristics of the development process, which can cause a lot of difficulties to the developers. Each developer should be aware of these problems and be prepared to solve them if they appear.

2.1 Rapid application development

As it was mentioned before, managers often say that something is easy, that it can be done fast and if possible, finished within an unreasonable timeframe (very often, when asked for a deadline, managers reply: "The deadline is yesterday!")... This is very common requirement and can cause many problems. Even if something can be done "quick and dirty" developers should avoid this development style because they can regret it in later phases. Figure 2 shows the progress of the software project in the typical scenario and in the scenario using our methodology.

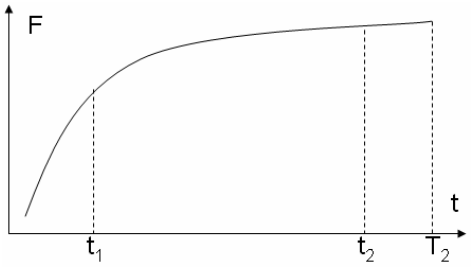


Fig. 2a. Progress of the software project in case of "Quick and Dirty" development style. Each software process, roughly speaking, has tree phases: setup phase, central development phase and final tuning phase. In case of a "quick and dirty" development style we can notice that in the first phase everything looks excellent because lot of functionality is developed fast. But in later phases, it becomes very difficult and time consuming to make even a small change.

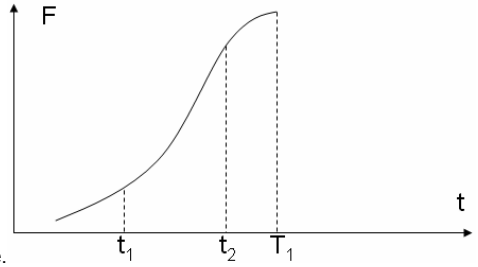


Fig. 2b. Progress of the software project using our methodology. In our methodology, a development phase can be much shorter. The only problem is that in the early phases of development there is no significant progress from the manager's point of view so they become suspicious and require quick and dirty development style. Phase 1 (before t_1) is used for the development of the framework and modeling of the software. Phase 2 (between t_1 and t_2) is a phase of the exploitation of framework potentials and most of the functionalities are implemented in that phase. Phase 3 (after t_3) is used for the final tuning and needs to be done manually.

As it is shown in the figure 2.a., it seems that quick and dirty approach is better, because it is faster in the first phase, but later our methodology proves to be much better because everything is implemented clean, and modeled properly so later variety of functionalities can be implemented extremely fast. In case of a "quick and dirty" style, in the later phases not only development cannot progress, also in many cases some code parts need to be rewritten from the scratch, which is very inefficient and time consuming.

2.2 Requirements are changed during development phase

Very frequent scenario is that manager notices that some "small" requirement should be added in the development phase, and even in the test phase. This can happen because managers don't know the business process in detail or don't know much about the capabilities of their developers so they give just rough specification in the beginning and later add more details. Typical example is that if some table is displayed, then some new column should be added or removed, or a number of displayed rows should be changed or some simple functionality should be added.

2.3 Extension of the existing system

Typical requirement is that the existing system should be extended with new functionality. Nowadays all companies want to have web portals for doing eBusiness. This is very tough requirement because existing systems can be very old and their design can be obsolete for a long time, even technology which was used for that old system can be abandoned. So, a developer needs to be extremely careful, needs to make strict interface to the old system and to make communication protocol with the old system highly parameterized, as will be described later

2.4 Fancy and user friendly front end

The last, but the most important thing is that a user interface needs to have a fancy design and to be user friendly. An example of the typical enterprise web portal is presented in the figure 3.

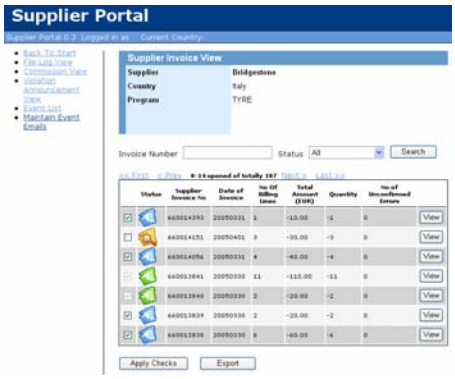


Fig. 3. An example of the user interface of the modern enterprise web portals. Variety of data is displayed and a lot of functionalities are offered to the end user.

Modern enterprise systems have plenty of functionalities and a nice design. It is very difficult to develop and maintain both of these characteristics, due to the problems mentioned above. An old legacy software system can hide behind the nice fancy user interface and many changes should be performed in the new

application in order to make the whole system working properly. That's why a new application needs to be extremely flexible.

3. PROPOSED SOLUTION

We propose software architecture presented in the figure 4. Proposed architecture has four major layers: Presentation Layer, Business Logic, Framework, View Layer and Database. Purpose and responsibility of each layer will be presented in this chapter.

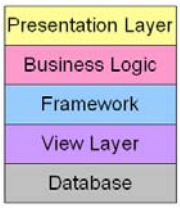


Fig. 4. Software architecture. This software architecture is suitable to meet all requirements for building the enterprise information system and to successfully handle all mentioned problems.

1) View layer

View layer is the first layer on top of the Database. This layer is extremely important because it separates the code from the database. This is useful because database is shared and its structure can be changed anytime, so it is very good development style if we make those changes transparent for the code. Also, database can have significant problems in the design and it is much better to fix some problems first and then build the application on top of the clean database. Following transformations of the database structure can be very useful and make the rest of the code much better.

a) Merging structures in the database

It often happens that attributes of one entity are spread over several tables. In that case it is much better to join those tables first, using view, and then build application which sees one entity in one view/table.

Fig. 5. Joining tables in the database. If one entity is spread over several tables, it is much better to join those tables using view first and then build the application which sees one entity in one view/table.

b) Extraction of key entities

Very common problem is that the database design is not normalized. It means that several entities are joined and stored in one single table. In that case it is much better to use views to extract these entities first, to have clean model and then build application on top of it. Figure 6 presents this situation.

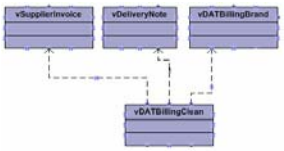


Fig. 6a. Extraction of entities from a single table. In case that database has a design problem in which several entities are joined and stored in the single table, using views can extract those entities.

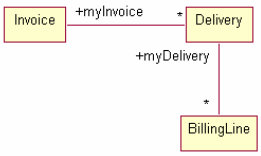


Fig. 6b. UML model achieved using views. After we perform the extraction of entities using views we will have clean model of software, and each entity will be stored in the single table. Code which should be built on top of that database will be much cleaner.

c) Recycling the data

Common case is that in the database we have tables with huge number of columns (we once saw a table with 78 columns!) and we don't need most of them. Also, some column values can be packed or some abbreviations can be stored (legacy data). In those cases it is much better if data are refined first before they go to the application. Views can help us for that. Figure 7. presents view layer on top of table layer in the database.

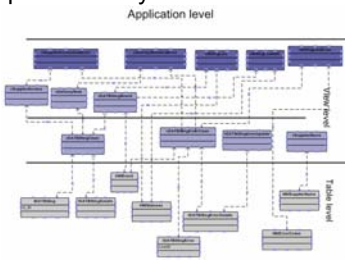


Fig. 7. View layer. Even we have just a few tables in the database, view layer can be big and also contains a sub layers. Each layer will perform one of the transformations mentioned in this section.

2) Framework

Our ultimate goal is to develop the application which looks like presented in the figure 8. We can notice there are a lot of elements of the presentation layer and a lot of functionalities. To put them all together we will add one more abstract layer which will speed up the later development.

Fig. 8. Ultimate application. Our final goal is to design the application which looks like one presented on the picture. Obviously we have in here elements of the presentational layer (data and images) and the business logic (buttons).

Responsibility of this layer is to be a one more abstract layer which will help in the rapid development of business logic and presentation

layer. Upper layers will be developed by customizing required look and feel and required functionalities in this framework.

Two major classes in the framework are `TableView` and `Command`. The idea is that class `TableView` should be responsible for drawing of its items and class `Command` should be responsible for the proper behavior. Common case is that for each row presented in the table several functions can be executed. That's why we developed class `TableView` as the container for objects of the class `Command`. In that way a class `TableView` will receive several object of the class `Command` and for each of them will draw a button next to each row. When the button is pressed, a `TableView` object will detect which button is presented and a proper command will be executed. UML diagram is presented in the figure 9.

Fig. 10. UML modes of the base classes in framework. Classes `TableView` and `Command` are abstract classes. All classes responsible for the presentation will be derived from the class `TableView` and all classes responsible for the behavior (business logic) will be derived from the class `Command`.

The `TableView` class is responsible for the presentation and class `Command` is responsible for the behavior. UML diagram of the derived classes is presented in the figure 11.

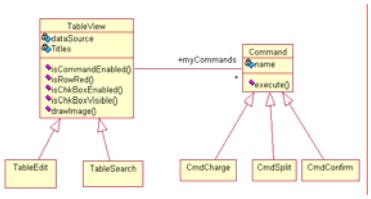


Fig. 11. UML Class diagram. Left side of the diagram is reserved for the classes responsible for the presentation and the right side is reserved for the classes responsible for the behavior. Design of classes `TableView` and `Command` is completely orthogonal. It means that each derived class from the presentation part can be combined with the each `Command` class.

This way we have the strict separation between presentation and the business logic. If we want to add new functionality we just add new class and everything else remains the same. Also, if something in the presentation should be changed, a new class should be derived and everything in the behavior part remains the same.

3) Business logic

Central part of the business logic layer is the class `Command` which implements Design Pattern Command. This abstract class has the abstract method `Command::Execute` which should be redefined in each derived class. Class implements the design pattern Command which allows easy development of functionalities: logging, security protection, unto and all other functionality related to command execution.

4) Presentation Layer

Base class `TableView` is central in the presentation part. That class is responsible for the look and feel of the system. Other classes like `TableEdit` or `TableSearch can be derived` from that class , and they will have a specific look and feel for the specific functionality they support. We noticed in the discussion on the problems that **every human visible item should be parameterized**. This layer and the class `TableView` provides that. It contains methods which control if any element will be presented and how it will be presented. By default, all elements are visible but in the derived classes the way of presentation can be changed.

5) Database Manager

Often requirement is that system should be independent from the specific RDBMS. It means that code should be able to use Orcale, MS Sql Server, SQLite and other databases. To achieve that, another layer needs to be introduced. That layer will hide specific database from the rest of the system.

4. Conclusion

This approach is successfully applied in the development of information system for Ford. Our assumptions about possible problems were correct and proposed architecture solved all of them. We proved that development of enterprise applications using described architecture has many advantages:

- Development is fast. Only the first phase seems to be slow. In other phases, the development is extremely fast and required functionalities are developed also very fast.
- Changes in the requirements during the development phase are handled easily.
- Problems related to the database design are solved efficiently.
- Problems in the communication with other developer can be handled easily because architecture is very flexible with two levels of indirection: View Layer and Framework.

Acknowledgment

We would like to thank especially to Dragan Milićev, Dejan Marjanovic and Nebojša Pirićanac, excellent developers and computer scientists, for teaching me how to build high quality software for enterprize sistems.

References

[1] Milićev, D., *Object Oriented Modeling in Language UML*, Mikro knjiga, Belgrade, 2001.

[2] Lazarević, B., Marjanović, Z., Aničić, N., Babarogić, S., *Database systems*, Faculty of Organizational Sciences, Belgrade, 2003.

[3] Gamma, E., Helm, R., Johnson, R., Vlissides, J., *Design Patterns*, Addison-Wesley, 1995.

[4] Robinson, S., Allen, K., Cornes, O., *Professional programming*, *C#.Net*, Cet, Belgrade, 2005.

Milovanović M. Miloš was born in Kragujevac, Serbia in 1980. He is a PhD student at Barcelona Supercomputing Center, Barcelona, Spain. He graduated at the Faculty of Electrical Engineering (ETF), University of Belgrade in 2004. He was teaching and research assistant at ETF from 2004 till 2006, where he was teaching six courses related to Computer Science and was involved in several international projects. He attended variety of international programming contests and won lot of awards. The most significant awards were silver medal at International Olympiad in Informatics (IOI) in Turkey in 1999 and 5th place in Nokia Code Master Competition among 700 contestants.

Challenges in Geovisualization

Ragia, Lemonia; and Fiume, Eugene

Abstract—The increasing complexity and pervasiveness of geographical information in many walks of life is giving rise to some difficult challenges in the growing field of geovisualization. We take stock of these challenges and argue that mature technologies taken from areas such as databases and computer graphics may offer partial resolution of these challenges.

Index Terms—computer graphics, databases, geovisualization.

1. INTRODUCTION

THE cartographic representation is the traditional method to depict geographical information. This information is represented classically by two-dimensional maps. The visualization of geographical information has become increasingly important due to the escalating need for better and more realistic viewing, more flexible data models, and more diverse data integration, in a wide variety of applications. Geovisualization has become the common term for the visualization of geographical information. It refers not only to the visualization of spatial data but also to the use of new techniques for displaying information that has been extracted following data analysis.

Considerable attention has been devoted to evaluating the existing tools and examining new geovisualization methods. Geographical Information Systems (GIS) are now connected directly to visualization tools. Three groups have predominantly collaborated to develop tools and to investigate new visualization methods: the Association for Computer Machinery's Special Interest Group on Graphics (ACM SIGGRAPH) (www.siggraph.org/~rhyne/carto), the ICA Commission on Visualization and Virtual Environments(www.geovista.psu.edu/sites/icavis) and GeoVRML which is the Virtual Reality Modeling Language for geographical data (www.geovrml.org). Many GIS companies are working on more complete and effective scene representations and a wide variety of visualization systems are already in existence.

Although there has been considerable progress, there remain limitations and some areas need further investigation. Gahegan [5] points out that there are barriers to the development of effective exploratory visualization tools. We believe high performance computer graphics tools and its techniques can contribute greatly to geovisualization and that greater research effort should be devoted to this endeavour.

The effectiveness and use of computer graphics in geovisualization is receiving increased attention. A proposal for virtual environment technology for geovisualization and the need for 3D dynamic displays is described in [10]. The importance of simple color-based highlighting styles is presented in [15], and the importance of 3D computer graphics techniques for geovisualization is also denoted in [3].

In this position paper, we give an overview of recent progress in the area of geovisualization and strongly argue for the need for increased attention to applying techniques from 3D computer graphics to this field.

2. STATE OF THE ART

A variety of different GIS systems or tools exist that support access, data analysis and geovisualization for many activities in many application domains. Recent improvements have largely concentrated on the following issues.

- *Cartographic support*. Many more graphical cartographic features are now available in real systems. Levels of detail may be represented by symbols, colors, annotation and text. Rules can be specified to depict the geometry, natural phenomena, physical objects and relationships among spatial features. Multilayer representations are supported by many systems, and every layer has its meaning and relevance in the real world.
- *Multiple data formats*. A wide variety of data formats can be created through the use of different methods for data acquisition. Terrestrial, aerial and subterranean formats may be used in the same or different systems. Many tools are available to convert among formats or to allow systems to interoperate using different formats. For example, different image formats may be supported and different representations of spatial data such as raster and vector are often allowed.
- *Performance and Scalability*. Several tools have been developed to improve the general quality of visualization and to support system scalability. Particularly when dealing with large data sets, new functions make it

Ragia, Lemonia; and Fiume, Eugene; Department of Computer Science; University of Toronto

possible to zoom, to navigate, to fly-by, and sometimes to animate large and complex scenes [9]. Better performance has been achieved using advanced application program interfaces and more careful attention to systems-level issues such as temporal coherence, caching and predictive data prefetching [13].

- *Metadata and Documentation*. Documentation can be supported by more refined tools that record information about the sources of the data, descriptions of individual steps during the data acquisition process and annotations of the general characteristics of the data. The geospatial metadata record includes a catalog of elements describing the characteristics of each data set. Normally this is embodied as a file presented as an XML document that includes all this information and supports every data set [7].
- *Database.* Many vendors and companies that develop GIS's have devoted considerable effort to the use and management of spatial data within a database. Such systems use some spatial database management functions and some provide interfaces to database management products. They are typically based on the relational data model which mainly includes tables with attributes, the description of their instances and some relationships between the geographical objects [19]. There is work that provides integrity constraints derived from the primitives and spatial relationships in spatial databases [1].
- *Increasing use of the third dimension.* A GIS traditionally parameterizes geographical data such as a height-field as a function of two rectangular or angular dimensions. With the increasing diversity and volume of geospatial data, GIS's must now incorporate fully three- or four-dimensional spatiotemporal parameterizations. As mentioned, many systems currently represent a third dimension as an attribute or value that is seen as a function of the other two dimensions. Furthermore, ad hoc solutions are often employed. For example, the visualization of 3D contours and isosurfaces is often flattened into two dimensions for viewing [8]. Several approaches to 3D geovisualization are emerging. One approach, for instance, integrates web-based 2D and 3D technologies for architectural visualization [18]. In another approach 3D modeling is used in geological applications [16, 17].
- *Visualization of texture.* Natural objects have textured surfaces that derive from phenomena such as vegetation or terrain. It is necessary to model surfaces more realistically to enhance human analysis and processing of geospatial data. Some work presents texturing techniques for terrain visualization applications [2], and others have demonstrated how synthetic texture can be generated [12].

3. GRAPHICAL LIMITATIONS

All Geographic Information Systems are mainly geared to a particular application. A quick count reveals about twenty categories for Geographic Information Systems, including soil information systems, urban information systems, and so on. Most of them engage in collecting, analyzing and visualizing data, but they still have considerable limitations for the visualization of spatial information. As mentioned in some recent research work, there is a need to improve the interoperability of three dimensional geovisualization components. Computer graphics rendering libraries and standards evolved independently of geospatial data models. This has resulted in inefficiencies associated with geovisualization. We identify the following key areas of geovisualization that require considerably more work:

1. **Time series animation**. Animation must become a first-class primitive in a geovisualization system as it can depict different forms of temporal variation of spatial data. We can identify the several categories of change: a) measured or simulated evolutionary data changes due to external events such as an earthquake, growth, erosion, or the construction of a new street; b) data changes due to inconsistencies in spatial or temporal data sources arising from combining disparate data sets; c) data changes in the representation due to generalization, enhancement or abstraction. There is some work using computer animation for time series data taken from satellites [11], but traditional tools for geovisualization do not usually incorporate tools to create and manipulate time series animation. For example, the changes of the sea shorelines cannot easily be depicted as a function of tides or seasons.
2. **3D Geovisualization and Extensibility**. Users must be able to create their own features, change scale, rotate objects, etc. The techniques and tools have to be interactive. There is a need for better modeling systems in order to reconstruct data in 3D. Traditionally, spatial data is parameterized two dimensionally but a great deal of data is now three dimensional. Manipulating such data interactively and realistically will naturally require higher-performance hardware and software. However, it will also require more general and powerful geovisualization tools. For

example, the generalization of topological relationships to 3-D for geological purposes is not supported at this point and requires further research.

3. **Combining a database management system with a graphics display system**. The functionality of such a combination is to develop interactive, application-independent data exploration and visualization tools. Such a technology would not rely on special data transformations or formats. Such variations would be accounted for in a data description language. Furthermore, it is widely accepted that commercial geographic information systems are not user-friendly. Just as an appropriate abstraction is needed for data, the user-interaction layer must also be abstracted from the hard-core functionality of any underlying GIS. Although this idea was presented some years ago [6], there has been little progress in this direction. There is a prototype [4 that combines database technology and visualization to support the visualization of ecological data but since the geographical data sets are so complex it is very difficult to develop a general tool with the level of abstraction we are advocating.
4. **Texture mapping.** While (pseudo-)coloring a two-dimensional surface is a fairly well known operation, there has been relatively little research on the more aggressive use of planar two-dimensional or volumetric three-dimensional textures to geovisualization. The former can provide details of the surface structure of an object, allowing easy disambiguation among, for example, varieties of stone, wood, vegetation, etc. On the other hand, a volumetric texture can represent interior structure, such as sedimentary layering, and veining. Volumetric textures can also represent densities of phenomena that have no easily characterized shape, such as mist, smoke, clouds, bodies of water, etc. Texturing provides important visual cues for human observers to analyse and classify, often without prior automatic segmentation.
5. **Distributed Geovisualization**. With the rapid proliferation of data on the Internet, a data set pertaining to the same location can become available from different sources taken at different times. Huge amounts of data of the same location can now come from satellites, aerial images, sensor networks, etc., but that may have been relevant to different applications such as urban planning, geology, health, econometrics and demographics. Such data sets are often stored and managed locally and users often have privileged access only to only one. The integration and concurrent exploration of these data sets can yield considerable insights into the real geospatial structure of a region. However, there are no tools that are capable of automatically extracting and visualizing data located in distributed, disparate databases [14]. We use the term distributed geovisualization to refer to the orchestrated use of tools and data that may be distributed across many applications, workstations and data servers.
6. **Integration of Geovisualization and Geocomputation**. The constantly increasing computational power and rendering speed of commodity workstations is bringing us to the point where geovisualization could occur in tandem with geocomputations such as data integration and analysis. Furthermore, the advent of efficient physical simulations, while still time consuming, afford a further opportunity for interactive visualization. Virtually everything that was once thought of as an off-line process is now a candidate for online, interactive implementation integration. This will allow much more dynamic kinds of data exploration. For example, the analysis of a specific observable phenomenon such as a forest fire or rock slide may be coupled to a simulation interactively to permit a broader exploration of its underlying causes or predictions of its future behaviour. The importance of exploratory data analysis will thus require much more dynamic methods that can help to discover relationships and hidden parts interactively.

4. TOOLS FROM COMPUTER GRAPHICS

Although still a very active field of research, the fundamental tools of computer graphics has matured over the past decade into a powerful, stable, high-performance technology. While its main applications have been to film and video, video games, virtual reality, scientific visualization and medical imaging, it also has considerable potential more generally in the area of information visualization. Computer graphics is the science and technology of computer-assisted visual communication, and is responsible for performing a transformation from geometric and physical models to sequences of images. The input primitives for such a process can take on many forms, provided there is a way to transform these primitives into a well understood spatial representation. Furthermore, the underlying algorithms of computer graphics are typically independent of form factor and image resolution, although there exist many customizations to specific display systems and architectures such as large-format display systems, CAVEs, desktop workstations, laptops, handheld devices and embedded systems. The use of more advanced computer graphics techniques appears to be an obvious one for geovisualization, and yet there has been relatively little progress in this area.

We outline a few basic technologies of computer graphics that we feel could be useful in geovisualization.

1. *Color and Shading*. Any object that can be given a color may also be assigned other geometric properties such as a locally-varying surface normal and material properties (water, reflectivity, material composition). With the addition of one or more light sources, this would immediately permit the computation of optically relevant information such as surface reflectance and transmission and shadows.
2. *Geometric modeling, transformation and viewing.* A wide array of modeling tools exist to manipulate geometric representations. Furthermore, graphics systems provide easy workflow to view, clip or slice through models.
3. *Texture mapping*. The introduction of variations in surface coloration through a texture map will allow the realistic rendering of terrains, vegetation and other two-dimensionally parameterized phenomena. Adding three-dimensional textures would allow the computation of illumination in gaseous or fluid environments using multiple scattering.
4. *Bump mapping and displacement mapping*. Textures may also be used to create "bumps" by perturbing local surface normals, and they can also displace the actual geometry according to a height function. This will add more realistic viewing while still maintaining interactivity.
5. *Realistic lighting*. Many computer graphics systems are capable of modeling a wide range of artificial and natural lighting conditions.
6. *Image quality*. Rendering algorithms in graphics are preoccupied with producing high quality images. The use of such technology for text and graphics would automatically lever built-in image quality features such as anti-aliasing and higher quality sampling schemes.
7. *Multiscale representations and levels of detail*. Because computer graphics has a long tradition of needing to work interactively on a wide array of devices, the data representations that are used are characterized at different levels of detail. This affords storage and computation budgets that are amenable to the display environment. Furthermore, level-of-detail algorithms are often employed within the compution of a single image to handle issues such as varying data density across the image. For example, in regions of high perspective magnification such as in the foreground, a greater degree of detail will be needed than in the background, or toward a horizon line.
8. *Animation*. The full power of stop-frame, keyframe, kinematic and dynamic animation techniques would be available to those attempting to perform geovisualization. There is little reason to reinvent the wheel.
9. *Visual Simulation*. A wide array of simulation technologies has been developed that are amenable to interactive viewing. This would provide an attractive workflow for visualization applications in which data analysis, simulation and visualization could occur simultaneously.
10. *Hardware acceleration*. It is important for geovisualization to "surf" the wave of cheap, high-quality graphics acceleration cards that are available virtually as commodity components. Graphics systems are usually able to provide fast, accelerated viewing at interactive rates for screen resolution and then send the same scenes to offline renderers for very-high quality rendering for film or high-definition video.

5. CONCLUDING REMARKS

The time, we believe, is clearly right for much more dramatic exploitation of computer graphics technology in the display of geospatial information. Computer graphics has largely responded to the challenges of computer animation, entertainment, and scientific visualization. Researchers in GIS have developed their own visualization techniques in isolation that have proven to be somewhat effective but do not go far enough. There is a tremendous potential to adopt the tools and techniques that have been developed over the years in computer graphics to let users explore the full potential of spatial data. This paper has explored some techniques that we feel will be of interest to geovisualization. We have also strongly advocated the need for greater interactivity, for cleaner data abstraction, and for better schemata for data integration.

REFERENCES

[1] Borges, K. A. V., Davis, C. A. Jr., Laender, A. H. F., "Integrity Constraints in Spatial Databases", Idea Group Publishing, Database integrity: challenges and solutions, 2002, pp. 144-171.

[2] Döllner, J., Baummann, K., Hinrichs, K., "Texturing Techniques for Terrain Visualization", IEEE Computer Society Press, in proceedings of the conference on Visualization, 2000, pp. 227-235

[3] Döllner, J., "Geovisualization and Real Time 3D Computer Graphics".

[4] Falkenberg, C., Kulkarni, R., "Case Study: Using Spatial Access Methods to Support the Visualization of Environemental Data", IEEE Computer Society, Proceedings of the 6th IEEE Visualization Conference, 1995, pp. 400-403.

[5] Gahegan, M., "Four barriers to the development of effective exploratory visualization tools for the geosciences", Taylor & Francis, International Journal for

Geographical Information Science Vol. 13. No 4. 1999, pp. 289-309.
[6] Garrett, M. T., Foley, J. D., “Graphics Programming Using a Database System with Dependency Declarations”, ACM Transactions on Graphics, Vol. 1, No. 2, 1982, pp. 109-128.
[7] Lehto, L., “XML in Web-based Geospatial Applications”, 3rd AGILE Conference on Geographic Information Science, 2000, pp 162-167.
[8] Lopes, A., Brodlie, K., “Interactive Approaches to Contouring and Isosurfacing for Geovisualization”, Elsevier, In Exploring Geovisualization, Dykes, J., MacEachren, A. M., Kraak, M.-J., (eds.) on behalf of the International Cartographic Accociation, pp. 345-361.
[9] Ma, K. L., Stompel, A., Bielak., J., Ghattas, O., Kim, E. J., “Visualizing Very Large – Scale Earthquake Simulations”, IEEE Computer Society, ACM/IEEE SC Conference, 2003, pp. 48.
[10] MacEachern, A. M., Edsall, R., Haug, D., Baxter, R., Otto, G., Masters, R., Fuhrmann, S., Qian, L., “Virtual Environments for Geographic Visualization: Potential and Challenges”, ACM Press, In Proceedings of the Workshop on New Paradigms in information Visualization and Manipulation in Conjunction with the Eighth ACM International Conference on information and Knowledge Management, 1999, pp. 35-40.
[11] Meisner, R., E., Bittner, M., Dech, S. W., “Computer Animation of Remote Sensing-Based Time Series Data Sets”, IEEE Geoscience and Remote Sensing Society, IEEE Transactions of Geoscience and Remote Sensing, Vol. 37, Issue 2, Part 2, 1999 pp. 1100 – 1106.
[12] Rachev, B., Ilieva, D., “An Approach for Texture SDI Visualization”, Proceedings of the 4th International Conference on Computer Systems and Technologies - CompSysTech 2003, pp. 124-129.
[13] Rhyne, T.M., MacEachren, A., “Visualizing Geospatial Data”, in the ACM SIGGRAPH 31st International Conference on Computer Graphics and Interactive Techniques Course Notes, 2004,
[14] Rhyne, T.M., “Visualization Viewpoints”, IEEE Computer Graphics and Applications 2004, pp. 13-17
[15] Robinson, A. C., “Highlighting Techniques to Support Geovisualization”, Proceedings of the ICA Workshop on Geovisualization and Visual Analytics, 2006
[16] Shumilov, S., Thomsen, A., Cremers, A.B., B. Koos, B., “Management and Visualization of Large, Complex and Time-dependent 3D Objects in Distributed GIS”, Proceedings of the 10th ACM international symposium on Advances in geographic information systems, 2002, pp. 113-118.
[17] Sprague, K. B., De Kemp, E. A., “Interpretive Tools for 3D Structural Geological Modelling Part II: Surface Design from Sparse Spatial Data”, GeoInformatica, Vol. 9, No. 1, 2005, pp. 5-32.
[18] Yan W., “Integrating Web 2D and 3D Technologies for Architectural Visualization, Applications of SCG and X3D/VRML in Environmental Behavior Simulation”, Web3D , Columbia, Maryland, 2006, pp.37-45.
[19] Zeiler, M., Modeling Our World: The ESRI Guide to Geodatabase Design (book).

Network Systems Intrusion: Concept, Detection, Decision, and Prevention

Pleskonjic, Dragan; Omerovic, Sanida; and Tomazic, Saso

Abstract — ***This paper analyzes concepts for intrusion detection processes; building decision making (DM) criteria on the bases of intrusion detection, and prevention based on DM as a last level of protection in computer systems and networks. The second part of the paper discusses a practical implementation for Intrusion Detection and Prevention Systems (IDPS), based on Wireless technology (WIDPS). Basically paper concentrates on the problems/answers of how to differentiate between legal and illegal access, i.e. intrusion and what are the key and root causes of this difference. Two issues are differenced: finding the set of concepts needed for detection and a set of criteria for DM in IDPS. Paper concludes with achieved results and future goals related to automated DM process in wireless technology.***

Index Terms — ***agent, artificial intelligence, concept, decision making, detection, intrusion, prevention, sensor, server, system, wireless network.***

1. INTRODUCTION

THE problem of intrusion in computer systems and networks has been around for decades. Unfortunately, that problem is growing as the variety of different threats is increasing every day, especially with the massive usage of computers, distributed systems and the Internet. Wireless and mobile networks give us new challenges with respect to IDPS, as their nature is to spread their network signals around without known exact boundaries. Wireless systems, which are competitive by their price/performance ratio, provide an ability of anytime, anywhere connectivity. At the same time, this means that intrusion can happen anytime, anywhere in the network.

Since the phases for both attack and response are layered processes (later explained in Section 2.2), paper presents the IDPS as a DM System in a form of hierarchical set of the following six layers: Unstructured data, Data Retrieval, Data Modeling and Analysis, Concept Layer, Decision Making, and Decision [23].

Manuscript received February 25, 2007.

Dragan Pleskonjic is with the Finsoft Ltd, UK (e-mail: dragan.pleskonjic@finsoft.com). Sanida Omerovic and Saso Tomazic are with the Faculty of Electrical Engineering, University of Ljubljana, Slovenia (e-mails: sanida.omerovic@lkn1.fe.uni-lj.si, saso.tomazic@fe.uni-lj.si).

In order to populate all six layers with useful data that will lead to automatic intrusion prevention, one needs to answer the following 7 W's questions [26]:

- Why? - Why does one need an IDPS in one's computer system/network?
- Who? - Who sets the rules for IDPS criteria (the owner of the computer, the network, the programs, the processed data, or services offered on the network)?
- When? - When to react if detection of an intrusion is established (processing time for detection or prevention and corrective action)?
- Where? - Where are the places where computer systems, network, programs, data, and services are most vulnerable?
- What action/state? - What are we protecting and from whom? What are the necessary actions to be taken when detection of intrusion happens?
- (W)How? - How we can differentiate between attack and legal access? How to build a model an ideal IDPS (what are the necessary concepts)? How to implement such a system? At the end of the paper, a proposition of a practical implementation for WIDPS is given.
- Which? - Which ones are the possible implementations of IDPS, based on the 7 W's criteria?

Related to answering the first W's question "Why" one can consider WIDPS as a superset of traditional IDPS designed for wired networks. This is a consequence of the fact that the sets of threats, attacks and vulnerabilities of wired networks are only a subset of those for wireless networks. In other words, wireless networks are vulnerable to all intrusion types possible in wired networks, but, in addition, there are numerous possible intrusions that are wireless specific.

"Wired" means that the user's computer is physically attached and that wire concentration exists; an intruder/attacker needs to plug directly into the network. In wired networks there is network security perimeter. On the other hand, in wireless a network the intruder can stay anywhere in an area covered by the signal and intrudes, and remains unseen. There is no exact "border" between an internal and an external network. This causes a loss of the exact

classification between insider and outsider attacks which is one of the important concepts in classic IDPSs.

Sometimes people assume that host based systems prevent insider attacks and network based systems prevent outsider tasks. Authors may not agree with this in practice, but it is the case that, as soon as one adds a Wi-Fi signal, the border of defense becomes unclear and is not sharply defined.

The problem one has is the case of classic IDPS is that they cannot provide an adequate and satisfying level of protection for wireless networks. One needs WIDPS or at least necessary extensions to current systems in order to cover wireless based threats, attacks and vulnerabilities.

The answers for the rest of the W's questions are structured in the shape of DM layers as follows: Intrusion Detection, Decision Support, and Prevention. Then follows a proposal for WIDS, based on those DM layers and Artificial Intelligence (AI). Paper concludes with future challenges for WIDPS and IDPS in general.

The aim of this paper is to stress the various possibilities of WIDPS, since wireless technology is increasingly present in today's networks and therefore in people's life in general. Only the overview of WIDPS solution is presented. Full documentation is possible on request from the authors.

2. *Decision Making System in IDPS*

In this section, foundations of our IDPS solution are established, through the layers of a DM System. Our main goal is to determine where the weak spots of IDPS are and how to improve them.

The observation of processes in layers is not a new thing in science. One can take a look in a Six Dimensional Information World [29], OSI Network model, TCP/IP model, Semantic Web layered model [24], Layered Mobile Agent Architecture [36] etc, and see that different processes can be distinguished into several different hierarchical layers.

What is the process of DM in general and can it be applied to IDPS? In [23], [30] the authors presented the idea that every process which has Unstructured data and some Question as Input, and Decision on the output, can be seen as DM Process, as shown at Figure 1.

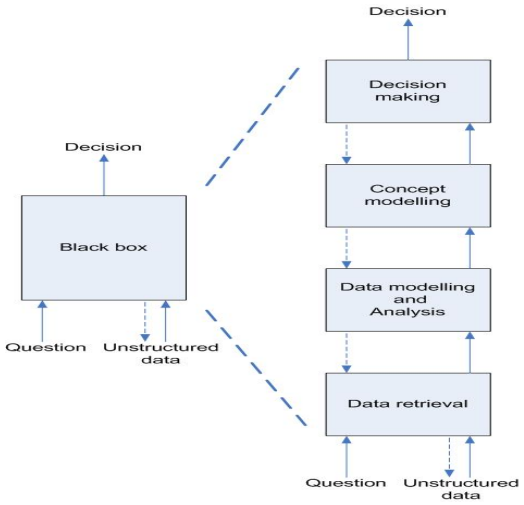


Figure 1. DM System Layers

Now let's discuss whether this model can be used for IDPS.

Generally speaking, the first step in IDPS is monitoring communications and/or behavior and collecting data. At the beginning of the process one has Unstructured Data. There is no exact method to differentiate the data of interest for IDPS. The majorities of data are, at first, not of interest and can be treated as noise in terms of their importance for IDPS. Usually that data represent regular use of the system; only small amounts of data and behavior can be part of illegal actions and intrusions (actually in real systems, on occasions this can be much more than a couple of percent, but still not a major part of communications).

The next layer in DM is Data Retrieval. Applied in IDPS, retrieval should be done in such a way as to extract the data of interest and not to overload the processor or to consume too many other resources, as that can be problem for itself. Also, efficient and reliable algorithms are necessary for this part. Moreover, in real time systems (which IDPS is considered to be) this problem can be very significant.

Usually IDPS deals with huge amounts of data. One of the first and most important questions is how to effectively do Data Modeling and Analysis, which also represents the next layer in a DM system. This should help one to understand and differentiate the data that are important for detection and appropriate prevention and what data are not useful for further analysis.

After the data are modeled and analyzed, one must construct concepts and build them into the Concept Layer. The concept layer in a DM system represents data structured in a way of conceptual hierarchy, where it is possible to add new concepts in the course of a simulation of process of learning, and also delete concepts which are no longer valid. Data are stored as concepts, so that that previously learned

knowledge (stored in some form of concept hierarchy) can be used to prepare for significant points and mimic human thinking, coupled with fast data analysis and processing to support the next stage – the DM process. One needs the Concept layer in IDPS in order to automate the detection and prevention processes with maximum possible accuracy.

Usually, it is hard to make all parts of a DM process clear. What is certain is that one must conform to the DM criteria, based both on the concepts and applications using the DM System.

This analogy with human DM (which is based on both previously learned knowledge and everyday perception) has been an important topic in science for many years now. Among possible mechanisms to achieve an automated process of DM are: Neural Networks, Expert Systems, Fuzzy logic [31] [32] [33] [34], etc.

IDPS is observed as a layered architecture made of three components: Intrusion detection, Decision support, and Prevention. It is possible to present IDPS and DM layers in parallel, as shown at Figure 2. Intrusion detection consists of the first four layers of DM, and Decision support consists out of last two layers of DM. Prevention depends both on the Decision support layer and the present technology available.

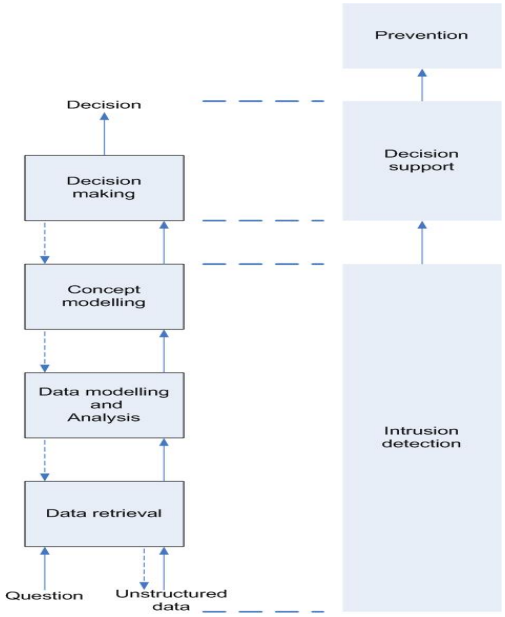


Figure 2. Parallel presentation DM and IDPS layers

In the next section, all three IDPS layers are explained in greater detail. The system proposed in this and some of other papers intend to employ mechanisms of automated DM as much as possible in order to make IDPS more human-aware and less like rigid computer-aware processes.

2.1 Intrusion Detection: Definition and Concepts

The potential threats to networks are numerous. Denial of Service (DoS), session hijacking, and sniffing are just a small sample of the potential attacks. While many of the attacks against wireless networks are similar to those against wired networks, 802.11 networks are generally subject to more threats.

Another problem is related to the open nature of Wireless Local Area Network (WLANs). Due to the propagation characteristics of wireless networks, there is limited control from where a signal can be accessed. This leads to a situation where, unlike wired networks, a hacker can manipulate or eavesdrop on the network from uncontrolled locations or geographical areas which were not intended to be served when the network was implemented.

Problems with network and especially WLAN security are well known, but solutions for those problems have not been coming quickly. That is especially the case with standardization. In order to satisfy market needs, some vendors started their own developments and proprietary standards. But this approach has not satisfied a wide base of customers. Work has also been directed toward confidentiality and authentication problems. On the other hand, problems with respect to some threats and resulting intrusions to corporate and organization networks, using wireless have not yet been addressed properly for a long time. There are some solutions existing but they are not directed to toward this area. At present in the area of intrusion protection there is an extension of Snort IDS named Snort Wireless [http://www.snort-wireless.org/], which is an open source solution. There are also vendors who are addressing different aspects of security, including wireless networks. But there are very few solutions specifically directed to WIDS. Namely, those solutions that are coming from: Airdefense, Enterasys, Network Chemistry, IBM, Cisco and WIDZ (rather waggish on first glance, but actually a serious approach). There are also other partial solutions.

WLANs can also create backdoors to wired networks. Many organizations spend a great deal of money on wired network security with extensive investments in firewalls, Virtual Private Networks (VPNs), and other security-enhancing technologies. A single unauthorized (rogue) wireless access point (WAP) connected to a wired network has the potential to create a backdoor to the wired network, circumventing the wired network security and thereby allowing a hacker to effortlessly gain access to a closed network. A wireless policy can help combat these threats. Fortunately, it is never too late to develop a policy, although an early adoption approach is mostly highly effective.

2.1.1. Intrusion definition

Before defining the intrusion, a list of specific attacks and vulnerabilities in Wireless Systems is presented. In order to be protected from the intruder, one must be fully aware of these two issues.

Wireless specific attacks and vulnerabilities:

- Easy access to 802.11 networks
- Unauthorized ("rogue") access points
- Unauthorized use of service
- Denial-of-service vulnerability
- MAC address spoofing and session hijacking
- Relatively easy traffic analysis and eavesdropping.

Various kinds of threats and actions of malicious people:

- Employing unauthorized access points (APs) – "rogue" or bogus APs that are designed to steal the association and login credentials
- War Driving - Probe requests which do not have the ESSID field set in the probe
- Flooding - attempts to flood the AP with associations
- Monkey / Hacker jacks
- Null probes
- Null associations
- Floods etc.

In the early 1980s Jim Anderson [35] was responsible for some of the starts in intrusion detection. Anderson defines an intrusion as any unauthorized attempt to access, manipulate, modify, or destroy information, or to render a system unreliable or unusable. This rather classic definition is accepted across the community and successfully covers important aspects of intrusion detection.

The definition of an intrusion depends upon the perspective from which it is seen (if one is attacker, the owner of the data, or an administrator, etc). It is very hard to differentiate WHAT an intrusion can be, especially because one can not predict the future intrusions and give them an irregular access label, before they have happened.

That is why is important to set the concepts of the intrusion, so one can define IDPS, no matter what is situation arises.

2.1.2. Intrusion concepts (IC)

In order to create an IDPS, various approaches can be taken. In this paper, IDPS is presented through the layers of a DM System, since authors believe this is the most efficient method to model a real life situation from the perspective of a network security environment.

An IDPS can be classified (according to various criteria) into different concepts: detection model, scope of the protection, time of attack, and type of response. The cases of IDPS concepts are given below.

As a detection model i.e. what is detected:

- Misuse detection i.e. signature based approaches
- Anomaly detection.

Scope of the protection (or by deployment) i.e. where detected:

- Network Based
- Host Based
- Application Based.

When an attack is detected:

- Real time
- After the fact i.e. after attack happened.

Type of response:

- Active
- Passive.

Based on these concepts, input data from an IDPS can be quite varied: traffic network parameters, application time, data transfer rate, etc., depending which concept is applied. After doing Data Retrieval, one proceeds to Data Modeling and Analysis are processed next. Basically system is doing filtering through all parameters that user receives from the network. Data which are crucial for the IC are further modeled in such a way from which concepts out of them. The rest of these data, which are not correlated with the IC, are discarded. A full description of how concepts are defined, built, populated and updated is given in the practical example of WIDS in Section 3.

2.2 Decision support in IDPS: Detection and Decision

After building the IC filtered from retrieved data next step in IDPS is Decision Support, namely Detection and Decision.

2.2.1. Intrusion definition

Detection in the IDPS is based on the IC. If one is able to detect the IC, then one is also able to detect the attack. For efficient detection of the intruder, one must analyze the attack anatomy. One approach is:

- Reconnaissance
- Exploitation
- Reinforcement
- Covering Tracks.

After the successful Detection of the intrusion, a proper Decision must be made.

2.2.2. Decision

In an IDPS Decision Support can be fully automatically or include human interaction. After the user has been recognized as an attacker, the system should be able to decide the following:

- Close the connection between the user and the system, so there is no dataflow
- Locate the physical resources of the intruder
- Learn how the attack happened and take self-learning steps so that a future attack of a similar kind can be blocked.

As stated earlier, after Decision Support, the next layer in IDPS is Prevention. This layer depends upon both the Decision support layer and the present technology available.

2.2.3. Prevention

Most of today's IDS systems will prevent an attack by alerting, dropping the offending packets, terminating the session for TCP and UDP based attacks, and dynamically establishing firewall rules that can keep the source of the threat off the network indefinitely or for a configurable period of time. Known sources of attacks can be stopped from even entering the network by enabling "Black Lists," while key corporate resources or trusted networks are always allowed to pass via "White Lists."

The phases for the response/attack prevention [27] are:

- Preparation
- Identification
- Containment
- Eradication
- Recovery
- Follow up and
- A loop for network self learning.

In general, the essence of Prevention is to protect individual/company data and provide Confidentiality, Integrity and Availability (the so called "CIA triad"). Viewing Prevention as a layered process (like Intrusion Detection and Decision Support) helps to better differentiate Intrusion weak spots and to builds a strong foundation of Prevention actions.

3. WIDPS

This section presents a full technical overview of our WIDPS solution based on the idea of DM layers. The letter W is used here to denote wireless, which is one of the growing network infrastructures at the present time.

It has been known that wireless networks suffer from all known wired intrusion types as well as some new intrusion types that are not specific for wired networks. In order to prevent these, a new approach is proposed, so one can detect and prevent intrusions of this type of network automatically.

3.1. The new idea and solution

When it comes to WIDPS, two problems arise:

- Wireless networks intrusions
- The kind of solutions which are intended to address typical problems in currently available IDPSs categories, which are briefly described above (misuse/anomaly detection, host/network, etc.) and which offer the resolution of such problems.

The idea is to create a WIPDS with a high degree of autonomy in tracking suspicious activity, detecting intrusions, making conclusions/decisions, and launching protective action (prevention) against the intruder.

Today's IDPS are mostly directed toward for wired networks. These systems are used in fixed and stable networks with "traffic concentration" units and their function is based on intrusion signs and rules. What the above mentioned systems lack are:

- What should be done about new attacks (different from the ones for which one know rules/signs)?
- How should the situation be handled where node is not based on a fixed physical place – which is case with wireless networks?

There is no "traffic concentration" in wireless networks (especially ad-hoc ones, but also in those with infrastructure too). In another words, there are no dedicated places where one can put IDPS agents (An agent is a program that performs some information gathering or processing task in the background). One needs a system that will support:

- IDPS agents in clients (wireless clients) for joint and local detection
- Cooperation between IDPS agents
- Self-learning capabilities
- Self-decision and alerts
- Self-defense against intrusion attempts (network administrators are not always present in the moment of intrusion).

This can be very complex in multidimensional and distributed system. Parties that will benefit from increased wireless security are:

- End user using wireless and mobile networks with increased freedom of work
- WISP (Wireless Internet Service Providers)
- Wireless software and hardware vendors, also where including in other wireless software and hardware products are included in other products
- The IT community in general (in this way one will encourage the average user to have more confidence in using wireless networks).

The use of Neural Network and Fuzzy logic in DM Systems (and also in WIDPS) has been considered as the Holy Grail for long time. With current developments in both areas possibility arises to consider their unity again and to create feasible system that combines them in order to give a more powerful solutions for the protection of wireless networks.

3.2. Architecture

The core of our WIDS proposed solutions are Agents [20]. In WIDS, agents are responsible for:

- Intrusion detection
- Decision support
- Prevention.

The architecture of our WIDPS solution is shown in Figure 3. WIDPS agents are connected (through secure channels) to each other and also through Access points. In that way Agents can efficiently exchange information and respond synchrously when an intrusion happens.

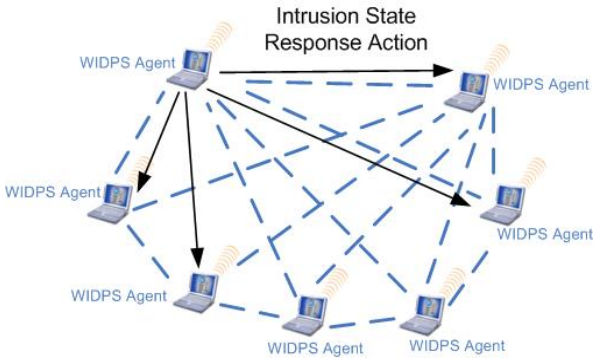


Figure 3. WIDPS Agents in ad-hoc wireless network [20]

This kind of solution is proposed for ad-hoc networks but not for the infrastructure mode. Some of this approach could be used to create basic principles for building a WIDS.

In order to protect any type of networks, the most important two concepts that need to be considered are:

- Intrusion Detection and Decision Concepts – A reactive approach
- Intrusion Prevention Concepts - Proactive action.

These two are quite different approaches, but should be very tightly coupled.

First, Intrusion Detection and Decision Concepts are discussed. The follow-up section is dedicated to Intrusion Prevention Concepts.

3.3. Intrusion Detection and Decision Support in WIDPS

For the WIDPS solution presented in this section, the following scenario related to Intrusion Detection Concepts is considered.

If one says that 0 is a proven intrusion, and 1 is a proven legal access, one can form limits in the following way:

- A is highest limit of decision where access is automatically classified as an intrusion
- B is lowest limit of decision where access is automatically classified as a legal access.

So, one can say that the scope is as shown at Figure 4:

- [0-A] is to deny access
- [A-B] requires human or artificial intelligence intervention
- [B-1] is to allow access

A and B are movable, where A ≤ B.

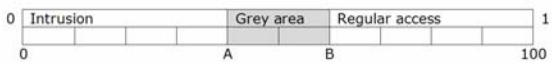


Figure 4. Example of the decision scale

One approach to making automated decisions is to define a scale with a range [0% – 100%] with thresholds at 40% and 60%. The system will consider as an intrusion everything in the range [0%-40%] and deny access. Also, the system will allow access for [61%-100%]. In the "grey area" which is, in this case in interval of [41%-60%], it will ask for assistance.

If one is talking about a WIDS Agent, assistance can be required from:

- A WIDS Sensor
- A WIDS Server
- Neighboring WIDS Agents
- And also from humans (for example from system administrator or response center personal). In that case it is not an automated, but a semi automated DM.

If assistance is required, suspicious activity will be held until an answer is received or a certain amount of time has passed. A good visual presentation of this is a slider with two movable notches representing low and high thresholds.

In [20], a mechanism has been proposed as shown at Figure 5 (slightly modified to fit also infrastructure mode also):

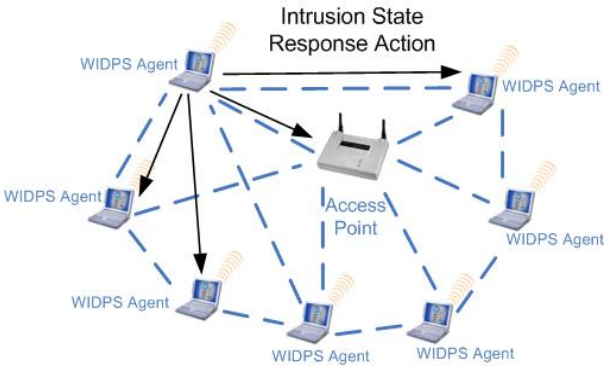


Figure 5. WIDS Agents in mixed (ad-hoc and infrastructure mode) wireless network

This mechanism considers:

- Data collection
- Detection (local and cooperative)
- Intrusion Response (local and global).

The DM processes in a WIDPS can be through the Conceptual Model WIDS Agent, as shown in Figure 6.

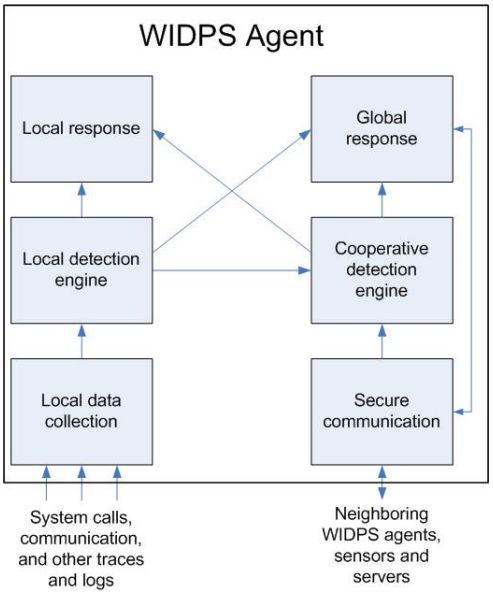


Figure 6. Conceptual model for a WIDPS Agent in an ad-hoc network [14]

The basic input in the Conceptual Model for a WIDS Agents is data that comes from an outside system and data that comes from the neighboring WIDS agents. These two types of data are processed in parallel inside the WIDS Agents. System data are gathered in the local data collection. Further, they are processed with a local detection engine and sent both to a cooperative detective engine and a local support medium.

Data from the neighboring WIDS agents are gathered through secure communications (and vice versa, specific WIDS agents also send data that they collect to other WIDS agents). Further, neighboring data are processed into the cooperative detection engine. The cooperative detection engine sends processed data to the local response medium and to the global response medium.

In several papers related to the WIDPS area [6] [8][9] authors propose an implementation of multiple layer and multiple dimensional systems, which consist of several key elements of IDS and IPS. These kinds of systems have been widely implemented at several levels in wireless and mobile networks. The essence of multilayer and multidimensional systems is the local response of particular system elements, as well as the existence of a global system response (in the case when attack happens). This multilayer system would have components such as:

- Intelligent agents and sensors (intrusion detection)
- Servers (central points for gathering and intelligent data processing, data exchange among other servers, agents, and sensors).

all with the purpose for generating proper prevention actions before any damage in the system occurs.

One of the purposes in the proposed the WIDPS is cross network connections and mutual accessing of distributed data bases. This way of organizing a WIDPS increases the partitioning of database and decreases the so called “zero time” (time counted after first appearance of new type of intrusion until the time when system is ready to protect its data). Having this property, a system is also resistant to a variety of new types of attacks.

The following a section is dedicated to a second type of Concepts which are necessary for the WIDPS solution presented, namely Intrusion Prevention Concepts.

3.4. Prevention in WIDPS

When suspicious activity is detected or an activity is classified as an intrusion, action should be launched. This action is classified as a local response meaning protecting the local host where the WIDS Agents are based or offering even wider protection in the network (if possible). Preventive action can also be defined as action to protect network(s) i.e. other neighboring parts of a network. Further, action can be extended to a wider network area and a higher level of security measures initiated.

In order to differentiate more precisely what intrusion is and what it is not, sensitivity and specificity definitions related to the WIDPS solution are presented next.

3.4.1 Sensitivity, specificity, and accuracy

In a WIDPS sensitivity is observed according to [25]. Therefore, one has four possible cases of attack detection: Intrusion Correctly Detected, False Alarm, Intrusion Missed, and Integrity Correctly Detected. One can make a table of all four states, and define them as: True Positive, False Positive, False Negative, and True Negative, as shown at Table 1.

		Intrusion	
		+	**-**
IDPS	**+**	**TP**	**FP**
Response	**-**	**FN**	**TN**

TP = True Positive = “Intrusion Correctly Detected”

FP = False Positive = “False Alarm”

FN = False Negative = “Intrusion Missed”

TN = True Negative = “Integrity Correctly Detected”

Table 1. Matrix of intrusions and detection

3.4.1.1. Sensitivity

Sensitivity is defined as the true positive rate (for example, the fraction of intrusions that are detected by the IDPS) [25]. Mathematically, sensitivity is expressed as follows:

True Positives / (True Positives + False Negatives)

The false negative rate is equal to 1 minus the sensitivity. The more sensitive an WIDPS is, the less likely it is to miss actual intrusions.

Sensitive WIDPSs are useful for identifying attacks on areas of the network that are easy to fix or should never be missed. Sensitive tests are more useful for "screening"; that is, when you need to rule out anything that might even remotely represent an intrusion. Among sensitive WIDPSs, negative results have more inherent value than positive results do.

3.4.1.2. Specificity

Mathematically, specificity is expressed as follows [25]:

True Negatives / (True Negatives + False Positives)

True negatives represent a WIDPS that is correctly reporting that there are no intrusions. False positives occur when an IDPS mistakenly reports an intrusion when there actually is none. The false positive rate is equal to 1 minus the specificity.

Specific IDPSs offer the greatest utility to the network administrator. For these programs, positive results are more useful than negative results. Specific tests are useful when consequences for false positive results are serious.

3.4.1.3. Accuracy

Often, a trade-off occurs between sensitivity and specificity that varies on a continuum dependent on an arbitrary cutoff point. A cutoff for abnormality can be chosen liberally or conservatively.

However, there are situations when one needs to spend the extra money to achieve both a high sensitivity and a high specificity. Accuracy is a term that encompasses both specificity and sensitivity. Accuracy is the proportion of all WIDPS results (positive and negative) that are correct.

For example, you might need a high-accuracy WIDPS in an area of the network such as the HTTP (Web) Server in Figure 7, which represents a typical corporate network and its protection.

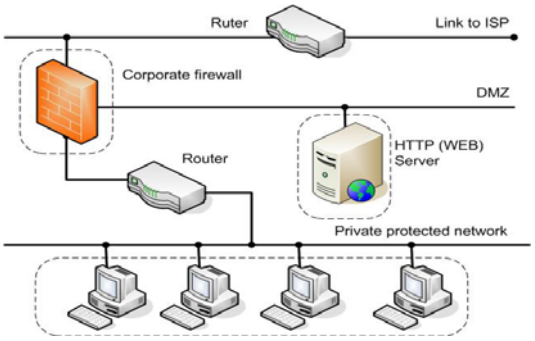


Figure 7. Typical corporate network and its simple layered protection

In this case, the Web server is under constant attack, and it would also cause the most immediate embarrassment and financial loss if compromised. In order to prevent this, the system needs to process any slight anomaly in an automatic manner because of the high traffic volume. In fact, to achieve the highest sensitivity and specificity here, one might need to combine layers of different WIDPS proposed.

3.5. WIDPS and artificial intelligence

As stated earlier, AI systems (Neural Networks, Expert Systems, Fuzzy Logic) can help mimic the way humans think. Human thinking, more precisely, a human DM process is what one wants to achieve in our WIDPS model automatically, so it is appropriate to implement this property of an AI system.

Two types of this AI are clearly distinguished in [28]:

- Strong AI - claims that computers can be made in a way to “think” as humans.
- Weak AI - claims that computers are an important tool in modeling simulations of human behavior.

Systems with AI have greater autonomy in data processing than “classical” computer systems, and also the ability to make an automatic response based on an implemented DM algorithm. An idea of a WIDPS based on a Neural Network model is presented next.

The main focus of Neural Networks usage is to achieve a desirable output to a corresponding input (in other words, train the network to “think” properly). Once the network is trained for n cases (with defined input and output), for n+1 input, it will give the correct (but for network unknown) output. This is achieved by varying the weights on the branches that are entering and exiting neural nodes in the network. Startup weights are changed over the training time, until the network output matches the correct (desirable) output. The core of the Neural Network model is presented at Figure 8.

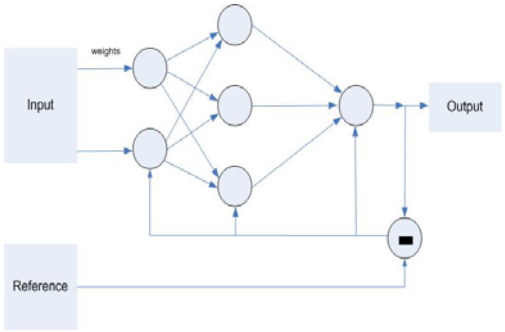


Figure 8. Neural network model

Building WIDPS based on Neural Networks for a particular computer system consists of the following three phases:

- Collecting training data - Data Retrieval in the DM System (i.e. obtain the audit logs for each user for a period of several days. For each day and particular user, form a vector that represents how often the user executed each command).
- Training - Data Modeling and Analysis, Concept Modeling in the DM System (i.e. train the neural network to build the user profile based on these command distribution vectors).
- Performance – Decision in the DM System (i.e., let the network identify the user for each new command distribution vector. If the network's suggestion is different from the actual user, or if the network does not have a clear suggestion, signal an anomaly, and take prevention actions).

These steps, also presented in Figure 9, can be reused over time in future WIDPS systems and the knowledge base can be shared among many servers (used in various networks). As result, one will a have fast growing base and systems with better efficiency and performances.

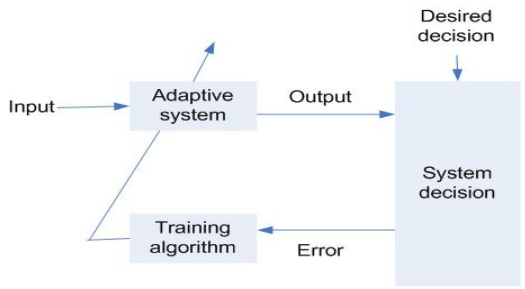


Figure 9. WIDPS solution based on Neural Networks

4. ACHIEVED RESULTS AND FUTURE WORK

Modern systems for intrusion detection and prevention demand greater data processing abilities, because they need to analyze, in a shortest period of time, all of the activities and traffic in the system (and system surroundings) in order to make a proper decision for intrusion detection and prevention. After that, in so called "real time", a system must activate its prevention procedures i.e. its response to attack. The impact of the human intervention in complex computer systems, where one has high data flow and a number of events and activities, is usually much slower (often late, or in worst case impossible) than what the system parameters demand. These facts leads to the conclusion that there is a need for automated DM systems, for which systems like Neural Networks, Fuzzy logic, Expert System, and similar AI systems, can offer automatic reasoning support.

This paper brought together basic approach and criteria for an IDPS (WIDPS specifically), including theory, methodology and measurement to examine and assess the value of Intrusion. Also, it introduces another view that is untraditional and subject to further analysis, changes and improvements. In order to assess the value of a real life network system based on this approach, it is necessary to design all parts of system, develop it, and put it into a real environment. This system environment involves a significant effort which must be taken in next stages of research as follows:

- Analysis of the simulated model in a laboratory environment using tools for simulation.
- Real development of a prototype and work in a controlled environment
- A real system in real environment based on DM System layers on the conceptual level, and artificial intelligence on the operational level.

5. CONCLUSION

In general, the DM mechanism to better differentiate whether something is or is not a legal access/intrusion is a similar problem to many other systems where it is necessary to make a decision based support on a limited set of known details and in limited time. Using DM System layers for the purpose of analyzing and designing efficient automated IDPS offers another opportunity to come up with better solutions. At the beginning, this approach appeared interesting and the first results have demonstrated the basic nature of the problem: introducing DM layers which can also be applied in a human way to bring about a decision. An important possibility is the use of experience with mechanisms that have been introduced in other similar systems where automated DM is also necessary. This is a new way of thinking about IDPS and requires that the work be continued. The next steps are related to building a prototype that will support IDPS concepts in wireless technology and logistic support based on Artificial intelligence.

It can be concluded that IDPS related research, development and usage will be expanded rapidly in the near future. This will occur because, on the one hand, the importance of computers networks and distributed systems (especially including wireless and mobile

networks) grows every day exponentially, and on the other hand, the level of the attacks by malicious people/groups is increasing over time, proportionally to the importance of these systems.

Generally speaking, although theoretical approaches to IDPS have been presented for a long time in scientific circles, their implementation is still not making the necessary progress demanded by companies, industry and everyday computer users. The special emphasis on a WIDPS is because of exponential usage growth of wireless and mobile networks. These types of networks which have unclear borders make it easier for malicious people/groups to perform eavesdropping, jamming, obstructions, and various kinds of new attack types, as well as some new methods of network intrusion.

ACKNOWLEDGMENT

The authors would like to thank to Prof. William Robertson, Dalhousie University, Canada and Prof. Tom Lincoln, University of Southern California, USA, for constructive feedback on this paper.

REFERENCES

[1] Paul Proctor, “The Practical Intrusion Detection Handbook”, Prentice Hall PTR, 2001
[2] Stephen Northcutt, Judy Novak, “Network Intrusion Detection”, New Riders, 2002
[3] Stephen Northcutt, “Inside Network Perimeter Security: The Definitive Guide to Firewalls, Virtual Private Networks, Routers and Network Intrusion Detection”, New Riders, July 2002
[4] Randall K. Nichols, Pannos C. Lekkas, “Wireless Security: Models, Threats, and Solutions”, McGraw-Hill, 2002
[5] Merritt Maxim, David Pollino, “Wireless Security”, Osborne McGraw-Hill, 2002
[6] Dragan Pleskonjic, “Wireless Intrusion Detection Systems (WIDS)”, 19th Annual Computer Security Applications Conference, Las Vegas, Nevada, December 8-12, 2003
[7] Dragan Pleskonjic, Veljko Milutinovic, Nemanja Macek, Borislav Djordjevic, Marko Caric: “Psychological profile of network intruder”, IPSI 2006, Amalfi, Italy, March 23-26, 2006
[8] Dragan Pleskonjic, "Wireless Intrusion Detection and Prevention Systems", Invited speaker at IDC IT Security Roadshow 2006, Belgrade, March 16, 2006.
[9] Dragan Pleskonjic: “Protecting wireless computer networks by using intrusion detection agents”, IPSI 2005, Venice, Italy, November 10-13, 2005
[10] Dragan Pleskonjić, Borislav Đorđević, Nemanja Maček, Marko Carić: “Sigurnost računarskih mreža”, Viša elektrotehnička škola, Beograd, 2006., ISBN 86-85081-16-5
[11] Dragan Pleskonjić, Borislav Đorđević, Nemanja Maček, Marko Carić: “Sigurnost računarskih mreža - priručnik za laboratorijske vežbe”, Viša elektrotehnička škola, Beograd, 2006., ISBN 86-85081-49-1
[12] Dragan Pleskonjić, Borislav Đorđević, Nemanja Maček, Marko Carić: “Sigurnost računarskih mreža – zbirka rešenih zadataka”, Viša elektrotehnička škola, Beograd, 2006., ISBN 86-85081-55-6
[13] Joshua Wright, “Layer 2 Analysis of WLAN Discovery Applications for Intrusion Detection”, [Online document], 2002 Nov 8, Available at: http://home.jwu.edu/jwright/papers/l2-wlan-ids.pdf
[14] Yu-Xi Lim, Tim Schmoyer, John Levine, Henry L. Owen. Wireless Intrusion Detection and Response, Proceedings of the 2003 IEEE Workshop on Information Assurance United States Military Academy, West Point, NY June 2003
[15] IEEE 802.11a/b/g is the set of Institute of Electronics and Electrical Engineers' standards for wireless computer communications. http://standards.ieee.org/getieee802/802.11.html
[16] Berkeley WEP Security Analysis Presentation http://www.drizzle.com/~aboba/IEEE/wep-draft.zip
[17] Yu-Xi Lim, Tim Schmoyer, John Levine, Henry L. Owen, “Wireless Intrusion Detection and Response”, Proceedings of the 2003 IEEE, Workshop on Information Assurance, United States Military Academy, West Point, NY June 2003
[18] Anup K. Ghosh, Aaron Schwartzbard, „A Study in Using Neural Networks for Anomaly and Misuse Detection“, Proceedings of the 8th USENIX Security Symposium, August 23-36, 1999, Washington, D.C.
[19] A.K. Ghosh, A. Schwartzbard, and M. Schatz. “Learning program behavior profiles for intrusion detection”, Proceedings of the 1st USENIX Workshop on Intrusion Detection and Network Monitoring. USENIX Association, April 11-12 1999
[20] Yongguang Zhang, Wenke Lee, “Intrusion detection in wireless ad-hoc networks”, Proceedings of the 6th annual international conference on Mobile computing and networking, p. 275 – 283, Boston, Massachusetts, United States, 2000
[21] http://snort-wireless.org [Online resource]
[22] http://www.loud-fat-bloke.co.uk/ [Online resource]
[23] Sanida Omerovic, Saso Tomazic, Charles Milligan, May 2006 Report on “Concept Modeling,” for Sun Microsystems, USA ,
[24] Tim Berners-Lee, James Hendler and Ora Lassila, “The Semantic Web,” Scientific American. Vol. 284, no. 5, pp. 28-37. May 2001
[25] Cyrus Peikari, Seth Fogie, "Maximum Wireless Security", Sams Publishing, 2002
[26] Sanida Omerovic, Tatjana Filimonova, Saso Tomazic, “Automatic Translation of Natural Languanges with E-speranto,” Proceedings of YU-INFO 2006, Kopanik, Serbia
[27] Matt Bishop, “Computer Security: Art and Science”, Addison Wesley Professional, 2003
[28] Mark Kantrowitz, “The AI Frequently Asked Questions,”maintained by Ric Crabbe and Amit Dubey, available online at www.faqs.org/faqs/ai-faq/general/, January 17, 2003
[29] Thomas Friedman, “Lexus and The Olive Tree,” Anchor Books, USA
[30] Ognjen Scekic, Djordje Popovic, Mina Micanovic, Veljko Milutinovic, IPSI Belgrade team report 2006 on “Concept Modelling,” for Sun Microsystems, USA
[31] Ambareen Siraj, Susan M. Bridges, Rayford B. Vaughn, “Fuzzy Cognitive Maps for Decision Support in an Intelligent Intrusion Detection Systems,”, IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th
[32] Hiren Shah, Jeffrey Undercoffer, and Anupam Joshi, “Fuzzy Clustering for Intrusion Detection,”, Fuzzy Systems, Proceedings of Fuzz-IEEE 2004.
[33] Aly El-Semary, Janica Edmonds, Jesus Gonzalez, and Mauricio Papa, “A Framework for Hybrid Logic Intrusion Detection Systems,” Fuzzy Systems, Proceedings of Fuzz-IEEE 2005
[34] Robert Fuller, “Neural Fuzzy Systems,” In Advances in Soft Computing Series. Springer-Verlag, Berlin/Heildelberg, 2000. ISBN : 3-7908-1256-0.
[35] James P Anderson, “Computer Security Threat Monitoring and Surveillance.”, James P. Anderson Co., Fort Washington, Pa., 1980.
[36] Oleg Kachirski, Ratan Guha, "Intrusion Detection Using Mobile Agents in Wireless Ad Hoc Networks," kmn, p. 153, IEEE Workshop on Knowledge Media Networking (KMN'02), 2002.

Investigating software dependability mechanisms for robotic applications

Laurent, T., Nana

Abstract*—this paper relates our case study of the application of software dependability to the design and implementation of robotics languages and architectures. The failure of a robot or that of its control system may lead to important damage to the environment in which it evolves or on human life. On the other hand, the amount of software used in robotics systems is increasing radically. For these reasons, solutions for software safety are needed in order to meet the dependability requirement inherent to such systems. In this paper, we show that in robotic systems, formal specification and verification methods are only used in a few cases and fault tolerance mechanisms as well as software testing techniques are exploited little. We have experienced static and dynamic testing as well as specification and verification with Petri nets in the context of the robotic language PILOT (Programming and Interpreted Language Of actions for Telerobotics) and its control system software. Among the benefits of the application of those approaches to the control system software, we can quote: detection and elimination of faults, fault tolerance capabilities, algorithms simplification, operator assistance in the design of plans, security of modification in the course of mission execution.

Index Terms—*Robotics languages, robotics architectures*, *software dependability*

1. INTRODUCTION

NOWADAYS, robots, like the tasks they are assigned, are more and more complex. On the other hand, important robotics projects may involve hundreds of engineers and/or researchers and several institutions. In such a context, the availability of proper architecture is necessary in order to structure the different levels of abstraction and to improve reusability and modularity of the hardware and software components of robotics systems.

Most robots are also critical systems. For example, the failure of a mobile robot or that of its control system, may lead to important damage to the environment in which it evolves or on human life. In the same way, in specific applications such as the nuclear field, remote surgery, or search and rescue missions, the failure of the robot may have disastrous consequences on human life.

The amount of software used in robotics systems is increasing rapidly. As a consequence, the use of software engineering techniques becomes necessary and even mandatory from the first steps of the realization of robotic projects in order to reduce re-designing costs and to facilitate software evolution and maintenance. These techniques include dependability means which are of particular interest for critical systems.

In this paper, we focus on software dependability in robotic architecture. Dependability is a justified guarantee that the system will appropriately perform its mission and not cause a catastrophe. Two means of dependability are often used: *fault avoidance* and *fault tolerance* [1]. Fault avoidance aims at producing a system containing as few faults as possible. To achieve this goal, the mechanisms used are formal methods and languages of specification and testing. As for fault tolerance, it aims at making it possible to the system to behave in a satisfactory way even in the presence of faults. *Fault forecasting* is also a common approach used for dependability. It aims at estimating the present number, the future incidence and the likely consequences of faults.

The second section of this paper is dedicated to the problem statement. The work related to software dependability in robotic architecture is presented in the third section. The language PILOT and its software architecture, both proposed by the LISyC Laboratory of Brest in France for safe robotic applications, are presented in the third section. The fourth section addresses the solutions proposed for the dependability of robotics applications in the context of PILOT and its software architecture. The results of the application of these solutions are presented in the fifth section. The seventh section is dedicated to the discussion about the solutions proposed or experienced within the framework of PILOT and those provided by other robotic architectures. This paper ends by conclusions and perspectives in the eighth section.

2. PROBLEM STATEMENT

Although various solutions have been proposed for general purpose software dependability, most of them are currently unapplied or applied in very few cases in the field of robotics. This is probably due to the need of adaptation for a proper use in robotics. Another reason is the low implication of computer science specialists in the early steps of the design of robotics systems. Finally, some aspects of the software engineering process such as testing are not considered as important as they should be.

The following questions are therefore of particular importance. Are the existing software dependability solutions adapted to the field of robotics? How to adapt existing software dependability solutions to the potential designers/developers of robotic systems?

In order to find an answer to the above questions, we have investigated the application of software

Manuscript submitted September 10, 2006.

L. Nana is with the Computer Science Department, University of Brest, CS 93837 – 29238, Brest Cedex 3, France (e-mail: nana@univ-brest.fr).

dependability to the design and implementation of robotic control architecture.

3. RELATED WORK

The modularity inherent to robotic architectures prevents faults by simplifying the development of their components and their testing. It also allows a better error confinement and as a consequence, better fault tolerance capabilities. Apart from this intrinsic feature, the taking into account of software dependability in the design and the implementation of robotic architecture still remains limited nowadays. A certain number of studies however were carried out in this field for fault avoidance and fault tolerance. As far as fault forecasting is concerned, few studies have been done. A short overview of robotics architectures will be given first, and then the state of the art of software dependability in those architectures will be presented.

3.1 Brief overview of robotic architecture

Robotics architectures can be classified in four main categories: *centralized architectures*, *hierarchical architectures*, *behavioral architectures* and *hybrid architectures*.

Centralized architectures were inspired by artificial intelligence [21]. They place planning in the centre of the system and share the axiom whereby the central problem in robotics is cognition, i.e., the handling of symbols to maintain and act on a model of the world, the world being the environment with which the robot interacts. Among centralized architectures, we can mention: the planning system STRIPS [22] and Blackboard architecture [9]. Centralized architectures are well adapted for tasks where reactivity and reflex are not essential criteria.

Hierarchical architectures break up the programming of the applications into increasingly abstract levels. The role of each level is to break up a task recommended to it by the higher level, into simpler tasks which will be ordered at the lower level. The highest level manages the global objectives of the application, whereas the lowest level orders the actuators of the robot. The best known example of this type of architecture is NASREM [15]. In the same family, we can quote the architecture of the LIFIA [8] and architecture SMACH [31]. Hierarchical architectures are adapted for tasks which are carried out in a foreseeable environment and which require a high precision. Hierarchical architectures generally have a rather low reactivity.

Behavioural architectures were born with Brooks's subsumption architecture [5]. They are based on the idea that a complex and evolved behaviour of a robot can emerge from the simultaneous composition of several simple behaviours. The architecture DAMN suggested by Rosenblat [26] is a variation on the work of Brooks. The main asset of behavioural architectures is their speed of reaction vis-à-vis the external events or to specific situations. However, they are not adapted to applications which involve the competition of several behaviours (difficulty to ensure the stability of execution of the law of complex orders) or impromptu changes of the mission (the behaviours are pre-established).

Hybrid architectures combine the reactive capacities of behavioural architectures and the capacities of reasoning specific to hierarchical architectures. Among them, we can cite: the CONTROLSHELL [28], the architecture of the LAAS [24] or architecture ORCCAD [11]. After this overview of robotic software architecture, the next subsection presents the state of the art of dependability in robotic architectures.

3.2 State of the art of dependability in robotics architectures

3.2.1. Fault avoidance

Formal verification

In the subsumption architecture, behaviours are modelled by augmented finite state machines. This modelling allows the application of formal verification methods. The architecture of the LAAS integrates an execution control component which has been synthesized from a model of acceptable and dangerous states using model-checking techniques. Its aim is to ensure that the system will never reach an inconsistent state [24]. In the architecture ORCCAD, the synchronous language ESTEREL and formal verification tools are used for the specification and verification of the control part [33]. In ProCoSA, an environment for implementing advanced mission management in autonomous vehicles, behaviours are formally described with Petri nets [2][3]. Running the system is achieved by an automaton which performs requests to functionality servers in accordance with the specification encoded in the Petri nets. Rutten proposed a toolkit for the safe programming of robotics applications [27]. The latter is based on the synthesis of controllers [25].

Testing

Very few reports exist on the application of testing to robotics software. However, an intensive testing was carried out in the context of the RAX architecture [4][13]. The authors underline the relevance of intensive testing, notably the problem of defining suitable test oracles.

3.2.2. Fault tolerance

Fault tolerance usually involves 4 main steps: *error detection*, *damage evaluation*, *error recovery* and *system repair*. We will only consider error detection and error recovery which are the main steps currently addressed in the context of robotics architectures.

Error detection

In RoboX9 [32], error detection is implemented with *timing checks* by using watchdogs. These checks are used for speed monitoring, obstacle avoidance, bumpers and laser sensors. RoboX9 also implements *reasonableness checks* to monitor robot speed, in order to ensure that this remains within a specified interval. In the R2C component of the LAAS architecture [24], error detection is implemented using *safety-bag-checks* which consist in intercepting and blocking the system commands if they do not respect a set of safety properties specified during development. Error detection is also performed with *Monitoring diagnosis* which consists in checking system behaviour against a mathematical model. It is mainly used to detect hardware errors. In the European project ADVOCATE II, an intelligent diagnosis system has been proposed for a deepened diagnosis of AUV (Autonomous

Underwater Vehicles) in case of failure [23]. This system is based on the following artificial intelligent techniques: Bayesian belief networks, fuzzy logic and neuro-symbolic systems. The architecture proposed in [29] by Seabra Lopes *et al.* for the assembly of tasks also provides at various levels of abstraction, functions for the diagnosis of error. The modelling of the failures of execution made through taxonomies and causality networks plays a central part in the diagnosis.

Error recovery

The main solutions used to implement error recovery in robotics systems are *error confinement*, *positioning in a safe state*, and *reconfiguration*.

In RoboX9, *error confinement* is implemented by using dedicated processors. Critical tasks such as localization and navigation tasks are executed on one processor and the other tasks are launched on another processor.

Positioning in a safe state may be executed in case of critical component failure or while executing a time-consuming recovery action. Such mechanisms are provided for example on the ASTER'x AUV of the IFREMER [16] (parking at the bottom of the sea, procedure of safe failure to release the robot from the ballast, etc.).

Software reconfiguration is not directly addressed in most robotics architectures. Nevertheless, one can quote corrective maintenance through the use of patches, which is a particular form of reconfiguration. It has been applied to recover the Martian rovers Pathfinder and Sojourner which failed respectively because of priority and RAM management failures.

Another implementation of fault tolerance found in robotic architecture is the *fault masking technique* [1]. It consists in taking advantage of the redundancy resulting from the combination and permutations of possible actions to replace a combination of actions which have become unusable, due for example to a failure, by an alternate combination of actions. Such technique is used in architectures such as that of the LAAS, ORCCAD, and CIRCA [7].

Security aspect in the realization of robotics missions was one of the main motivations of the project "Software Architectures for Mobile and teleoperated robotics" initiated by the LISyC Laboratory in the 90's, which led to the creation of PILOT and to its control system. PILOT system has been built as a hierarchical architecture with two main parts: a high level part located on the remote control computer and a low level part embedded on the teleoperated robot. In the two following sections, we present the work completed in this context for the reliability of robotics applications. A short description of the language PILOT and of its control system software is first carried out. The solutions for safety implemented within the framework of PILOT system are then approached.

4. SHORT OVERVIEW OF THE LANGUAGE PILOT AND ITS CONTROL SYSTEM SOFTWARE

4.1. The language PILOT

PILOT is a graphical and interpreted language mainly dedicated to the remote control of robots. It is based on the notion of action. An action encapsulates an order executable by the robot, a precondition and one or more supervision rules to which processing are associated. Two kinds of actions are discriminated: elementary and continuous actions. Unlike a continuous action whose end is triggered by an enclosing primitive of the language, an elementary action generally ends when its predefined goal is reached. Whatever kind it may be, an action is only executed when its precondition rule is true (unless the operator decides to force the execution). In the same way, if during the execution of an action, one of its supervision rules becomes true, then the corresponding processing is performed.

The language PILOT provides control structures for plan building: sequence, conditional, iteration, parallelism and pre-emption. Detailed information on the language PILOT can be found in [6, 14].

4.2. The control system software of PILOT

The control system software of PILOT (FIG. 1) is the interface between the user and the controlled machine (target robot). It comprises six modules: a *Man Machine Interface* (MMI), a *Communications server*, a *Rules Generator*, an *Evaluator*, an *Execution Module* or *Driver* and an *Interpreter*. These modules run in parallel and communicate through sockets and shared memory. The control system can work either in centralized mode, or in distributed mode. The choice of the working mode is done in a static way (before compilation).

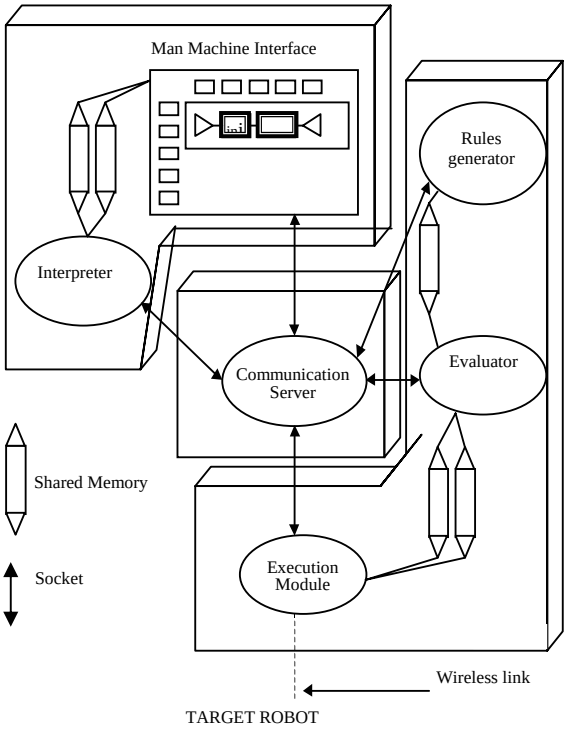


FIG. 1 The control system of PILOT

The MMI provides means for the construction of plans, the dynamic creation of actions (without recompiling of the code), and the modification of the plan before and during the execution of this last. It also integrates means for the supervision of the execution of the plan. The MMI stores the plan in a storage area shared with the interpreter. This reads the plan in shared memory and sends orders (request for evaluation of precondition of an action, order of starting

of the action, etc) to the other modules in order to carry out the execution of the plan. The communications server manages the communications between the modules of the control system. The role of the rules generator is to transform the character strings of preconditions and supervision rules into binary trees. It stores the result in a storage area shared with the evaluator. This evaluates the precondition and supervision rules by using the corresponding binary trees. The execution module is the interface between the robot and the control system. It translates the high level orders of the plan into low level orders comprehensible by the remotely operated machine. The execution module supports various protocols of communication (serial connection, Ethernet, FDDI).

5. SOLUTIONS FOR THE DEPENDABILITY OF ROBOTIC APPLICATIONS WITH PILOT

The solutions for dependability in the context of PILOT can be divided into two categories: features directly integrated in the language or implemented in its control system (internal mechanisms) and solutions related to the software development process (formal specification and verification, testing).

5.1. Internal mechanisms

The actions of PILOT comprise **precondition and supervision rules**. An action can be carried out only if its precondition is true. In the same way, when a supervision rule is satisfied, the associated processing is carried out. PILOT **plans can be modified during their execution**.

The environment of control of PILOT was equipped with a mechanism of **syntax-oriented edition** making it possible to guarantee the syntactic validity of the plan at each phase of its construction (insertion, modification, suppression of primitives) **[17]**. A **controller based on the formalism of "controllers' synthesis"** was also integrated into the control system.

5.2. Mechanisms related to the software development process

Software testing

Static and dynamic testing was applied to the interpreter which is one of the most critical modules of the control system of PILOT [18]. A simple simulator of robot was built for the dynamic testing operations in order to avoid damage to the real robot and its environment. The static test consisted, on the one hand, in reading the source code with the aim of detecting the programming errors and, on the other hand, in analyzing it in comparison with the algorithms of interpretation and the semantics of PILOT. As regards the dynamic testing, it consisted in defining testing data and applying them to the binary code of the interpreter. The testing data were defined by combining a functional approach with the experience feedback of the tests already carried out. For the definition of the representative sample of testing data, we adopted an incremental approach. The empty sequence was initially tested, and then the other primitives of the language were tested individually. Combinations in length, in width and in depth of the primitives of the language were then considered by increasing respectively, the number of elements in the sequences of the plan, the number of branches in parallel and conditional primitives, and the level of overlap. In order to have a limited set of testing data, we emitted stability assumptions: actions of the same type are interchangeable; the set of sequences resulting from the combination of any pairs of primitives is representative of the set of sequences comprising two or more primitives except for the problems of memory, etc.

Modelling, simulation and static verifications

The various modules of the control system software of the language PILOT were modelled using finite state automata and interpretation algorithms were defined for the various primitives of the language [6], but the automata were not checked.

The static and dynamic testing described above do not make it possible to guarantee the conformity of the interpretation of the plans to the operational semantics of the language PILOT. Complementary work was carried out to alleviate this disadvantage [19]. It consisted in modelling the algorithms of interpretation and checking their conformity in comparison with the operational semantics of the language in order to correct the possible dysfunctions and to regenerate the code of the interpreter starting from the validated model. Coloured Petri nets (CPN) were used for modelling and verification. The software used was Design/CPN which is currently replaced by CPN TOOLS (http://www.daimi.aau.dk/DesignCPN). Petri nets and more particularly CPN were selected for various reasons. Their graphic nature offers the user-friendliness necessary for a later use of the model as a means of communication between the various people implied in the development of the control system. They make it possible to represent relatively simply the various concepts of algorithmic and programming. The availability of tools, for the simulation and the checking of the models, was also an important criterion.

Test plans were generated using the approach adopted during the dynamic testing of the interpretation algorithms and their execution was simulated. The simulation making it possible to explore, in practice, only part of the execution paths, complementary work was carried out. From the CPN modelling the algorithms of interpretation and a test plan, the graph of accessible markings corresponding to the possible execution paths was generated using the Design/CPN tool. The graph of markings and the test plan were then transmitted to the verification program which examined, for each execution path, the satisfaction of the operational semantics of the language.

As far as the syntax-oriented edition mechanism mentioned above (subsection 5.1) is concerned, its conformity in comparison with a formal plan synthesizer was checked. It consisted in modelling PILOT plans and the edition operations in Prolog and then checking that all and only all syntactically valid plans were built [17]. This work was done using the SWI-Prolog tool.

In addition to the above verifications, a solution based on real time scheduling has been proposed in order to ensure real time properties of missions [12, 20]. It consists in modelling the whole control system of PILOT in terms of tasks, buffers and communication channels. From this model and from measurements performed on the platform, the maximum bounds on the occupation and the waiting time of its buffers are calculated [30] as well as the worst case response

times of tasks [10]. Once response times of the control system' tasks are computed, one can obtain the response times of the actions and control structures of the language PILOT.

After this presentation of the solutions proposed for the dependability of robotics applications with PILOT, the next section addresses the results emanating from their application to PILOT and to its control system software.

6. RESULTS

Precondition and supervision rules avoid unsafe execution of actions. If we consider for example the action "move forward" for a mobile robot equipped with detectors of obstacles, a rule of precondition could be the test of absence of obstacle. One of the supervision rules would be for example the test of presence of obstacles.

Thanks to the possibility provided by PILOT for runtime modification of plans, the operator can, in the event of dysfunction in the execution of a plan, make **modifications allowing the system to return in a satisfactory operating condition** (continuation of the mission or stop in a safe state). The **syntax-oriented edition mechanism** and the **controller** make it possible to preserve the advantages of the possibility of modifying plans dynamically. Indeed, the interpreted nature of the language and the possibility of modifying plans in the course of execution allow the execution of incomplete plans. One can thus launch the execution of a plan with missing parts such as the end of the main sequence or containing a parallel structure whose execution cannot finish because it is incomplete. The syntax-oriented edition avoids such situations. The controller ensures the semantic validity of the plan at runtime by disallowing modifications such as suppression of running actions, and allowing insertion of primitives only if it makes sense.

The **static and dynamic testing** made it possible to detect and correct errors of various types: error in the management of software interruptions and in the management of continuous actions termination, etc.

The **finite state automata** modelling the modules of the control system software and the **interpreter algorithms** provide a good basis for the prevention of errors (the application of formal verification methods is possible). Moreover, they make it possible to avoid errors due to the distortion of information throughout the design and implementation process of the control system software.

The **modelling of the interpretation algorithms with Petri nets** made it possible to note that simplifications were possible both on the level of the internal representation of a plan, and on the level of the algorithms of interpretation, and to apply these to the control system (suppression of a component in the internal representation of the plan, merging of the two sequence interpretation algorithms, etc.). The **simulation of the Petri net**s made it possible to detect termination problems (deadlock in the execution of an empty preemption primitive, etc.).

The **checking of conformity of the syntax-oriented mechanism** in comparison with a formal plan synthesizer allowed the validation of the proposed mechanism.

The work based on **real time scheduling** made it possible to characterize temporally the control system and to validate temporally PILOT plans before their execution.

7. DISCUSSION

The supervising rules mechanism is equivalent to the exceptions mechanism and constitutes a solution for the implementation of fault tolerance.

As far as formal methods are concerned, the solutions proposed for the LAAS Architecture as well as for the PROCOSA system of the ONERA are mainly oriented towards the generation of controllers whereas the ORCCAD architecture mainly deals with static verification of missions. Both approaches have been experienced in the context of PILOT. These different studies show that although significant effort is needed for the adaptation of formal methods to the field of robotics, it is not only possible but very useful. Controller synthesis is particularly important for architectures such as PILOT which allow runtime modification of missions (the set of authorized missions is restricted to those satisfying some formally defined safety constraints).

Virtually no experience on the use of software engineering techniques for the design of robot control architectures has been reported. Our experimentation of software engineering techniques of modelling, simulation and testing with the PILOT architecture shows that some specificities of robotics need to be taken into account, in particular for the testing (events generated by the robot which are not easily controllable, simulator required for hostile environments or to avoid damage to the robot under test or the environment, …). The characteristics of the mission programming (existence of elementary actions which impacts the structuring of plans) approach can also be exploited for the definition of testing data set.

Operator intervention on the course of the mission is a requirement of robotics applications. It is part of the features needed for fault tolerance of robotics applications. Making it possible to dynamically modify missions is therefore an asset for dependability of such applications. Nevertheless in order to be fully beneficial, such possibility needs to be complemented with additional features such as a syntax and semantic-oriented edition of missions. At first glance, such a mechanism seems time-consuming, but this is not necessarily true. Indeed, in our case study with PILOT, the verification needed in case of modification is limited to the direct enclosing control structure.

8. CONCLUSION AND FUTURE WORK

Formal methods have been used only in a few robotic architectures. In the same way, fault tolerance mechanisms are exploited little. One can also note the small number of experiments reported as regards rigorous testing in the realization of robotic architectures. Structured testing is however very important in the process of development of dependable software. Globally, a significant effort is needed in the application of software dependability techniques to the design and implementation of robotic architectures. The use of distribution in robotic

architectures provides the opportunity of improving fault tolerance through mechanisms such as replication. The work completed within the framework of the language PILOT and its control system brought generic solutions for the dependability of robotics applications: a precondition rule which ensures the safety of action execution, monitoring rules which make it possible to recover from an action failure, a syntax-oriented edition mechanism which assists the operator in the design of plans and reduces the risks of faults, possibility to modify missions during execution which makes it possible to recover from errors, mechanism of security of modifications in the course of execution. The static and dynamic testing as well as the formal modelling and checking applied to the interpreter of plans can also apply to other mission programming environments. Several perspectives can be envisioned in the continuation of the former work. One of them is the automation of the testing process. It is necessary to generate proper traces and to design and implement traces analysis methods for fault detection. Another perspective is the use of theorem proving to extend the validation of the PILOT syntax-oriented edition approach which was limited to plans of a maximum size of 14 (maximum number of elements of the list modelling the plan) due to the combinatory explosion of the model-checking approach formerly used.

REFERENCES

[1] Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C., "Basic concepts and Taxonomy of Dependable and Secure Computing", IEEE Trans. On Dependable and Secure Computing, January-March 2004, 1(1):11-33.

[2] Barbier, M., Gabard, J. F., Vizcaino, D., Bonnet-Torrès, O., "ProCoSA: a software package for autonomous systems supervision", 1st National Workshop on Control Architectures of Robots: software approaches and issues, Montpellier, France, April 2006.

[3] Barrouil, C., Lemaire, J., "Advanced Real-Time Mission Management for an AUV", SCI NATO RESTRICTED Symposium on Advanced Mission Management and System Integration Technologies for improved Tactical Operations, Florence, Italy, Sept. 1999.

[4] Bernard, B. E.,Gamble, E. B., Rouquette, N. F., Smith, B., Tung, Y. W., Muscettola, N., Dorias, G. A., Kanefsky, B., Kurien, J., Millar, W., Nayal, P., Rajan, K., Taylor, W., "Remote agent experiment DS1 Technology Validation Report", Ames Research Center and JPL, 2000.

[5] Brooks, R. A., "A robust layered control system for a mobile robot", IEEE Journal of Robotics and Automation, March 1986, pages 14-23.

[6] Fleureau, J.L., "Vers une méthodologie d'un système de programmation de télérobotique: comparaison des approches PILOT et GRAFCET", PhD thesis, Université de Rennes 1, France, 1998.

[7] Goldman, R.P., Musliner, D.J., Pelican, M.J., "Using Model Checking to Plan Hard Real-Time Controllers", In Proceedings of the AIPS2k Workshop on Model-Theoretic Approaches to Planning, Breckenridge, Colorado, April 14, 2000.

[8] Hassoun, M., Laugier, C., "Reactive motion planning for an intelligent vehicle", In Intelligent Vehicles'92 Symposium, Detroit, July 1992, pp. 259-264.

[9] Hayes-Roth, B., "A blackboard architecture for control, Artificial Intelligence", 1985, 26:pp. 251-321.

[10] Joseph, M., Pandia, P., "Finding Response Time in a Real- Time System", Computer Journal, 1986, 29(5):390-395.

[11] Kapellos, K., Simon, D., Espiau, B., "Control laws, tasks, procedures with ORCCAD; application to the control of an underwater arm", In 6th IARP (International Advanced Robotic Program), La Seyne sur Mer, France, 1996.

[12] Legrand, J., « Contribution à l'ordonnancement de tâches partagant des tampons », PhD thesis, University of Brest, December 2004.

[13] Lussier, B., Lampe, A., Chatila, R., Guiochet, J., Ingrand F., Killijian, M.-O., Powell D., "Fault Tolerance in Autonomous Systems: How and How Much?" 4th IARP/IEEE-RAS Joint Workshop on Technical Challenge for Dependable Robots in Human Environments, Nagoya, Japan, June 16-18, 2005.

[14] Le Rest, E., « PILOT: un langage pour la télérobotique », PhD thesis, Université de Rennes 1, France, 1996.

[15] Lumia, R., Fiala J., Wavering, A., "The NASREM robot control system and testbed", IEEE Journal of Robotics and Automation, 1990, 5(1), pp. 20-26.

[16] Nana, L., Marcé, L., Opderbecke, J., Perrier, M., Rigaud, V., "Investigation of safety mechanisms for oceanographic AUV missions programming", In Proceedings of the IEEE OCEANS'05 Europe Conference, Brest, June 2005.

[17] Nana Tchamnda, L., Marcé, L., "Vers une programmation sûre avec PILOT", MSR'2001, Colloque Francophone sur la Modélisation des Systèmes Réactifs, Toulouse, France, October 2001.

[18] Nana Tchamnda, L., Nicolas, V-A., Marcé, L., "Towards a formal approach for the regeneration of PILOT control system", 6th World Multiconference on Systemics, Cybernetics and Informatics, SCI'2002, IEEE Venezuela, Orlando, Florida, July 2002.

[19] Nana, L., Legrand J., Singhoff, F., Marcé, L., "Modelling and Testing of PILOT Plans Interpretation Algorithms", Multi-conference on Computational Engineering in Systems Applications, CESA'03, IEEE, Lille, July 2003.

[20] Nana, L., Singhoff, F., Legrand, J., Vareille, J., Le Parc, P., Monin, F., Massé, D., Marcé, L., Opderbecke, J., Perrier, M., Rigaud, V., "embedded intelligent supervision and piloting for oceanic AUV", In Proceedings of the IEEE OCEANS'05 Europe Conference, Brest, June 2005.

[21] Nilsson, N., "A mobile automation: an application of artificial intelligence techniques", In Proc. Int. Joint Conf. on Artificial Intelligence, 1969, pp. 509-520.

[22] Nilsson, N., "Shakey the robot", Technical Report 323, SRI, Menlo Park, CA.

[23] Perrier, M., Kalwa, J., "Intelligent Diagnosis for Autonomous Underwater Vehicles using a Neuro-Symbolic System in a Distributed Architecture", In Proceedings of the IEEE OCEANS'05 Europe Conference, Brest, June 2005.

[24] Py, F., Ingrand, F., "Dependable Execution Control for Autonomous Robot", IROS 2004 (IEEE/RSJ International Conference on Intelligent Robots and Systems), Sendai, Japan, September 28 - October 2, 2004.

[25] Ramadge, P. J., Wonham, W. M., "The control of discrete event systems", Proceedings of the IEEE, Special issue on dynamics of discrete event systems, 1989, vol. 77, no. 1, pp. 81-98.

[26] Rosenblatt, J., "DAMN: A distributed architecture for mobile navigation", Journal of Experimental and Theoretical Artificial Intelligence, 9(2/3), 1997, pp. 339-360.

[27] Rutten, E., "A framework for using discrete control synthesis in safe robotic programming", Rapport de recherche, INRIA, 2000.

[28] Schneider, S., Chen, V., Pardo-Castellote, G., Wang, H., "ControlShell: A software architecture for complex electro-mechanical systems", International Journal of Robotics Research, Special issue on Integrated Architectures for Robot Control and Programming, 1998.

[29] Seabra Lopes, L., Camarinha-Matos, L.M., "Learning to diagnose failures of assembly tasks", Annual Review in Automatic Programming, 1994, vol. 19, pp. 97-103.

[30] Singhoff F., Legrand, J., Nana, L., Marcé, L., "Extending Rate Monotonic Analysis when Tasks Share Buffers", In the DAta Systems in Aerospace conference (DASIA 2004), Nice, July 2004.

[31] Tigli, J.Y., « Vers une architecture de contrôle pour robot mobile orientée comportement, SMACH », Thèse de Doctorat, Université de Nice - Sophia Antipolis, Janvier 1996.

[32] Tomatis, N., Terrien, G., Piguet, R., Burnier, D., Bouabdallah, S., Arras, K. O., Siegwart, R., "Designing a secure and Robust Mobile Interacting Robot for the Long Term", In Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, September 14-19, 2003, pp. 4246-4251.

[33] Turro, N., « MaestRo: Une approche formelle pour la programmation d'applications robotiques », Thèse de doctorat, Université de Nice, Sophia Antipolis, Septembre 1999.

A contribution to development of methods for E-business and E-commerce

Milutinovic, Dusan V.

Abstract—*This paper represents an annotated bibliography of interest for development on new internet-based strategies for efficient company management, effective customer-relationships, development, outsourcing, and negotiation infrastructure in mobile semantic web, with a special emphasis on transitional economies. Each bibliographical unit (a paper) to be presented belongs to the one of the following categories:*

TE (Transition economies)
EB (Electronic business)
MB (Mobile business)
SW (Semantic web)
IS (Internet strategies)
CRM (Customer relationship management)
OS (Outsourcing)
NG (Negotiation)
MM (Marketing methods)
SC (Supply chain) and
MG (Management).

The approaches described in presented papers can serve as justification for the application of a novel approach introduced by this author: iterative gradual refinement based on the back tracking and feedback (IGRBTF). This paper concludes with the essence of the proposed approach, which is the subject of a follow-up paper.

1. INTRODUCTION

The field of interest for this study includes a vast body of knowledge. However, this study includes only the work which is of interest as a justification for a new approach elaborated later on.

Each paper is presented in a uniform way, with approximately one extended sentence, which points to the essence. For easier reading, all relevant references are classified, and listed at the end of this survey.

2. CLASSIFICATION

The basic classification used in this paper is flat, but it can easily be made hierarchical, if so necessary for future research.

Basic classes are defined in the abstract and more precise descriptions are presented next, from a number of various sources. More elaborated descriptions can be found in the open literature.

2.1 .Definition of the TE class

The term "transition economy" has often been used to describe the process of change in economies (such as in Eastern Europe) which are passing from a socialist economic system to a free market system based on private property. Transition economies undergo economic liberalization, letting market forces set prices and lowering trade barriers, macroeconomic stabilization, where immediate high inflation is brought under control, as well as restructuring and privatization, in order to create a financial sector and move from public to private ownership of resources.

2.2 Definition of the EB class

Electronic business (e-business) is any process that a business organization conducts over a computer-mediated network. It does not include only buying and selling, but also servicing customers and collaborating with business partners. Today, this is mostly done using the Web-based technologies.

2.3 Definition of the MB class

Business combines business processes, electronic business, and digital wireless communications (in other words, doing business using wireless services). It also means delivering real-time action and information anytime, anywhere via Web, e-mail, and mobile devices to employees, managers, partners, and customers. Mobile business technology enables you to achieve high organizational agility and deliver data, services, and products to customers wherever they are.

2.4 Definition of the SW class

The Semantic Web is a technology that intends to create a universal medium for information exchange, by putting documents with computer-processable meaning (semantics) on the World Wide Web. The Semantic Web is about two things: common formats for interchange of data (where on the original Web we only had an interchange of documents) and language for recording how the data relates to real world objects. For the semantic web to work in everyday applications computers must have access to structured collections of information

Milutinovic, Dusan V. Purdue University, West Lafayette, USA

and sets of inference rules that they can use to conduct automated reasoning.

2.5 Definition of the IS class

Internet Strategy presents a comprehensive plan to clearly define one party's goals for its own web site, vision to be transferred to the visitors of the web site, and how the website will achieve the purpose one party has for it. In other words, it is the plan for using the World Wide Web to add long-term value to ones company. Globally, it is a "game" plan that one will use to expand either e-business (enhancing business to business capabilities and transactions over the web) or eCommerce (providing goods and services over the web to one's customers).

2.6 Definition of the CRM class

Customer Relationship Management is a business strategy to identify, cultivate, and maintain long-term profitable customer relationships. It requires developing a method to select your most profitable customer relationships (or those with the most potential) and working to provide those customers with a service that exceeds their expectations. CRM entails all aspects of interaction a company has with its customer, whether it is sales or service related. For example, an enterprise might build a database about its customers that described relationships in sufficient detail so that management, salespeople, people providing service, and perhaps a selected customer can directly access information, match customer needs with product plans and offerings, remind customers of service requirements, know what other products a customer had purchased, and so forth.

2.7 Definition of the OS class

Outsourcing (or contracting out) is often defined as the delegation of non-core operations or jobs from internal production within a business to an external entity (such as a subcontractor) that specializes in that operation. Outsourcing is a business decision that is often made to lower costs or focus on competencies. Basically, it is contracting with organizations outside your country for work that could otherwise be done by employees within your company. Business process outsourcing is the act of giving a third-party the responsibility of running what would otherwise be an internal system or service. Typically, companies that are looking at business process outsourcing are hoping to achieve cost savings by handing the work to a third-party that can take advantage of economies of scale by doing the same work for many companies.

2.8 Definition of the NG class

Negotiation is a business process by which parties involved try to decide upon the issue of each party's contribution to the entire deal. It begins with an overview of the concepts of negotiation in businesses and goes on to different types of negotiation methods and technology's impact upon them. A business negotiation policy is a general guiding principle for achieving a business negotiation goal under some specified conditions. Negotiation process must be moral or ethical, it must respect the inherent worth and dignity of those involved or affected.

2.9 Definition of the MM class

Marketing is the act or process of buying and selling in a market. It is an organizational function and a set of processes for creating, communicating and delivering value to customers, and also for managing customer relationships in ways that bring benefit to the organization and its stakeholders. Marketing can be observed as a tool to help promote economic growth and development in a community.

2.10 Definition of the SC class

A supply chain, logistics network, or supply network is a coordinated system of organizations, people, activities, information and resources involved in moving a product or service in physical or virtual manner from supplier to customer. The primary objective of supply chain management is to fulfil customer demands through the most efficient use of resources, including distribution capacity, inventory, and labour. The supply chain is typically comprised of multiple companies who are increasingly coordinating activities via an extranet.

2.11 Definition of the MG class

Management represents the group of individuals who make decisions about how a business is run, the person or persons who control or direct a business, or other enterprise. Also, it can be the initiation and maintenance of an investment portfolio. Project management is the discipline of organizing and managing resources in such a way that these resources deliver all the work required to complete a project within defined scope, time, and cost constraints. Knowledge Management caters to the critical issues of organizational adoption, survival and competence in face of increasingly discontinuous environmental change.

* * * * *

In principle many other classifications can be generated. This one includes only the issues of importance for the specific research of the author, related to a newly proposed approach.

3. *PRESENTATION*

Each one of the above defined classes includes examples, which are presented next (minimum there per class).

3.1. The TE class

The TE class includes 4 papers, and their essence is presented next.

Paper [Guru 2002] investigates and analyses the specific conditions of an economy in transition (Romania), and the appropriate strategies for the implementation and use of e-banking.

Paper [Hadidi 2006] is dealing with the issue of implementing e-finance related initiatives, in order to achieve the United Nations Millennium Goal.

Paper [Vij 2006] focuses on the on the issues and challenges in e-banking, with a special focus on the regulatory and taxation issues.

Paper [Alter 2001] investigates the fundamental concepts in information systems, in the light of different meaning and implications of e-business and questions about the nature of the user in e-business.

3.2. The EB class

The EB class includes 12 papers, and their essence is presented next.

Paper [Tan 2006] analyses and specifies agent security services for as Agentcities banking services.

Paper [Yibin 2003] instigates new challenges in E-banking related to policy implications from the prospective of society, banks, and regulatory authority as well as government.

Paper [Kerem 2006] presents the development of a small and successful example of fast introduction of information and communication technologies in Estonia, in spite of the fact that little more that ten years ago it was a technologically relatively backward Soviet country.

Paper [AL-Taitoon 2004] discusses how a global help desk unit accomplishes help to mobile professionals in a global banking organization. This is achieved through analysis of the call tickets from the global helpdesk tracking system.

Paper [Malhotra 2002] examines the key assumptions of any informatics strategy and demonstrates why they should be considered afresh.

Paper [Ahearne 2006] performs a study within one company and mixes multiple data sources in order to analyse sales person performance.

Paper [Hawkins 2006] deals with e-business from both philosophy and technology aspects.

Paper [Galbraith 2006] examines the valuations of B2B Internet firms, and the issue of post-IP short-term under pricing (not good for profit) and long-term underperformance (also not good for profit).

Paper [Lee 2006] observes e-business like a tool for effective Internet-based computing and communications to execute both front-end and back-end business processes.

Paper [Aissi 2006] proposes a Process Coordination Framework for Web services and outlines the building blocks required for e-business automation.

Paper [Papazogly 2001] promotes a use of new global networking possibilities like sharing sales data, customer buying patterns, and future plans in order to achieve better revenue.

Paper [Geoffrion 2006b] presents the idea to connect operations research with related analytic technologies and information technologies as a vital engine of further development for the digital economy.

3.3. The MB class

The MB class includes 4 papers, and their essence is presented next.

Paper [Ropers 2006] investigates mobile business opportunities to create new revenue streams and enhance customer profitability and retention without breaking a successful e-business strategy.

Paper [Kim 2004] analyses e-business implication of the mobile Internet.

Paper [Camponovo 2006] focuses on the strategic uncertainties in a mobile business, where a large number of actors are trying a number of strategic approaches to position them in the most favourable position in the value system.

Paper [Giaglis 2006] extends the definition of mobile business to include a complementary set of ‘micro’ applications in indoor environments (such as museums, exhibitions, hypermarkets, and others), where location awareness can become a crucial parameter of value-added service provision.

3.4. The SW class

The SW class includes 4 papers, and their essence is presented next.

Paper [Losada 2005] contains the WSMO descriptions of the Semantic Web Services that will be deployed in the context of the mortgage comparison application.

Paper [Osterwadler 2002] introduced new e-business model ontology in the light that using

the concepts of business models can measure, simulate and learn more about the different aspects of e-business.

Paper [Gordijn 2006] presents a conceptual modelling approach to b-business called e3-value that is designed to help define how economic value is created and exchanged within a network of actors.

Paper [Gisolfi 2001] explores the business impact of web service, how to identify a relevant solution opportunity, and how to evaluate the many vendor strategies building around this technology.

3.5. The IS class

The IS class includes 8 papers and their essence is presented next.

Paper [Porter 2006] discusses the strategy used in Internet technology to shift the basis of competition away from quality, features, and service and toward price, making it harder for anyone in their industries to turn a profit.

Paper [Damanpour 2006] analyses cooperation integration dealing with the issues like customer loyalty, profitability, and competition advantage in the domain of e-business.

Paper [Bradshaw 2006] presents extensive research conducted over several years by Ovum researchers posing as customers into customer relationship management, call centre, and e-commerce strategies, and the software required to support them.

Paper [U.S Department of Commerce 2000] analyses issues of dramatic cost reductions in computers, computer components, and communication equipment as an important influence of U.S. economic expansion.

Paper [Gupta 2001] discusses issues in adapting and creating technologies used to design, build and deliver e-business applications.

Paper [Akkermans 2006] analyses information technology as a key strategic factor in industry, not just a tool to increase business-as-usual efficiency.

Paper [Kauffman 2003] presents various phenomena in the field; just as the Internet has changed the way many businesses conduct business, the Internet can also change the way academic researchers gather data.

Paper [Baake 2000] focuses on e-business applications (i.e., IT-applications) based on Internet technologies that support all types of business processes and products of an enterprise.

3.6. The CRM class

The CRM class includes 4 papers and their essence is presented next.

Paper [Rygielski 2006] observes (through data mining) the extraction of hidden predictive information from large databases - organizations can identify valuable customers, predict future behaviours, and enable firms to make proactive, knowledge-driven decisions.

Paper [Massey 2006] describes a 4-year initiative undertaken by IBM to re-engineer its customer relationship management process and capitalize on knowledge-based resources.

Paper [Romano 2002] presents the status and maturity of electronic commerce customer relationship management, an emerging subfield of management information systems. It is investigated through an exhaustive literature review of 369 articles from the first published article in 1984 through conference papers given in 2001 and 2002.

Paper [Grönroos 2006] discusses the nature and sometimes negative consequences of the dominating marketing paradigm of today, marketing mix management, and furthermore discusses how modern research into, for example, industrial marketing and services marketing, as well as customer relationship economics, shows that another approach to marketing is required.

3.7. The OS class

The OS class includes 3 papers and their essence is presented next.

Paper [Zeidler 2001] discusses the value of outsourcing in the light that it delivers the best class people, processes and technology under a pricing arrangement those benefits from economies of scale.

Paper [Schubert 2001] analyses the issues in "virtual world," where e-government and e-business are joined and working together.

Paper [Ono 2001] examines outsourcing to test whether productivity-enhancing specialization is facilitated in bigger cities.

3.8. The NG class

The NG class includes 3 papers and their essence is presented next.

Paper [Hammer 2000] presents architecture and framework, including negotiation protocol for automated negotiations applying multiple IDEAL (Internet-based Dealmaker for E-business) servers.

Paper [Kumar 2006] explains the commonality in the structure of different price negotiation mechanisms such as fixed price sales, various forms of auctions, and brokerages.

Paper [Beam 2006] provides a definition of automated negotiation within electronic commerce. It outlines two barriers to automated negotiation: the ontology issue and the strategy problem.

3.9. The MM class

The MM class includes 3 papers and their

essence is presented next.

Paper [Luo 2001] presents trust production and privacy concerns on the Internet as a framework based on relationship marketing and social exchange theory.

Paper [Barakat 2006] approaches marketing as the business function that identifies customer needs and wants, determines which target markets the organization can serve best, and designs appropriate products, services, and programs to serve these markets (basically it is a tool that guides the entire organization).

Paper [Grönroos 2006] discusses the nature and sometimes negative consequences of the dominating marketing paradigm of today, marketing mix management, and furthermore discusses how modern research into, for example, industrial marketing and services marketing, as well as customer relationship economics, shows that another approach to marketing is required.

3.10. The SC class

The SC class includes 3 papers and their essence is presented next.

Paper [Tatsiopoulos 2006] presents research results in the area of Extended Production Data Management System development, which supports (by the business processes) order management, subcontractor selection, and multi-site/multi-firm production orders release.

Paper [Johnson 2002] examines how the Web is changing supply chain management.

Paper [Chopra 2006] deals with the question of Internet value in a firm, in the light that such a value strongly depends on the firm's industry and firm's strategy that it pursues.

3.11. The MG class

The MG class includes 3 papers and their essence is presented next.

Paper [Phan 2002] presents a study of e-business competitive advantage strategies using the success of Intel as an example.

Paper [Barua 2006] identifies the critical links between e-business drivers, financial indicators, and operational excellence measures.

Paper [Geoffrion 2006 a] analyses the impact of the emerging digital economy on management science research.

* * * * *

This survey covers a relatively broad list of topics; however, for all the covered topics, one can say that the existence of the Internet and present-day high technology, enables one to create an iterative feedback- based approach to treatment of selected issues in e-business and e-commerce, and wider.

The presented literature shows a drawback of all solutions elaborated in the here before specified literature. This is the fact that the first step solution, based on the Internet data, is never optimal. One has to do iterative step-by-step refinement, cautiously taking into account the effect of the previous iterations. This is done by connecting feedback and facts. This is especially the case in the countries in transition. This means the solutions from strong economies have to be applied cautiously and taking into considerations feedback facts (culturally and economically specifics make a difference).

In other words, some of the presented approaches include elements of look-ahead and are proactive, but none of them include a solid algorithmic base for iterative refinement based on feed-back. Therefore, there is obviously a room in this field for innovations based on iterative refinement and feedback.

4. Essence of the Proposed Approach

Essence of the proposed new approach can be understood from the title of the approach: Iterative gradual refinement based on the back tracking and feedback (IGRBTF).

One can make a business decision which is believed to be correct (approximately correct), and then one can employ high-tech monitoring tools (including the Internet), to see the effects. If effects (according to the selected performance measure) are showing a positive trend (no matter what is the chosen criterion of positive trend), one can continue with a more detailed model of decision-making. One can proceed (with a constant awareness about feed-back) through the whole business process. A pseudo-code for the IGRBTF may look as follows (subject to improvement in the follow up research):

While the business process is in progress, Do:
- a. Make an appropriate decision
- b. Establish performance monitors
- c. Track the effects of the decision
- d. Create the feed-back analysis, based on output from performance monitors.
- e. Do not stop if results satisfactory, since the environment may change in the near future; do keep informing upstream the decision-making process about the current state-of-the art.

End Do!

This approach ensures that, especially in rapidly changing economies, the decision-making process will become adaptive to the changing's in the environment. This issue is of importance both for established markets, and for markets that are developing.

The follow-up paper develops this pseudo-code into more details and applies the approach to the specific environments of financial auditing, e-government, and digital economy in Serbia.

5. CONCLUSION

This paper first presents a survey of a number of related fields, and tries to extract the essence from a number of relevant references. Then, it presents the findings and proposes a business decision making methodology, applicable to general e-business and e-commerce based on telecommunications, and wider.

Various groups of researchers and practitioners can benefit from this paper. It can serve as a starting point for reading, for those who like to enter the field, and it can also serve as the starting point for research efforts to create new methods based on high-tech tools or the Internet.

The future work implies a more rigorous survey of selected papers, as well as the development of a detailed methodology, based on the pseudo-code presented here. Of course, theoretical and empirical evaluation of the proposed methodology has to be completed in the future, too.

ACKNOWLEDGMENT

The author is thankful to Professors Bozidar Radenkovic and Jovan Filipovic of Belgrade University, Serbia, for their input during the most critical stages of this research. Also to John O'Connor and Cathleen Sullivan of Purdue University for their initial feedback related to the methodology used for this study.

REFERENCES

[1] [Guru 2002] C., Guru, "E-banking in transition economies: The case of Romania," Journal of Financial Services Marketing, Volume 6, Number 4, June 2002, pp. 362-378.
[2] [Hadidi 2006] Hadidi, Rasule, "The Status of E-finance in Developing Countries"
[3] [Vij 2006] Vij, M., "E-Banking: An Emerging Perspective of the Regulatory and Taxation Issues"
[4] [Alter 2001] Alter, S., Ein-dor, P., Markus, M. L., Scott, J., Vessey, I., "Does the trend toward e-business call for changes in the fundamental concepts of informational systems? A debate", Communications of the Association for Information Systems, Volume 5, Article 10, April 2001
[5] [Tan 2006] Tan, J. J., Poslad, S, Titkov, L., "Securing agent-based e-banking services"
[6] [Yibin 2003] Yibin, M., "E-Banking: Status, Trends, Challenges and Policy Issues", CBRC Seminar "The Development and Supervision of E-banking", Shanghai, Nov. 24-26, 2003
[7] [Kerem 2006] Kerem, K., "Internet Banking in Estonia"
[8] [Al-Taitoon 2004] Al-Taitoon, A., Sorensen, C., "Supporting Mobile Professionals in Global Banking: The Role of Global ICT-Support Call-Centres," Journal of Computing and Information Technology, vol. 12, no. 4, pp. 297-308.
[9] [Malhotra 2002] Malhotra, Y., "Knowledge Management for E-business performance: Advancing Information strategy TI "Internet Time", Information Strategy, The Executives Journal, vol (16) 4, Summer 2002, pp.5-16
[10] [Ahearne 2006] Ahearne, M., Schillewaert, N., "The Effect of Information Technology on Salesperson"
[11] [Hawkins 2006] Hawkins, J. L., "What's E-Business? Buzzword du jour, or mandatory for the Millennium?"
[12] [Galbraith 2006] Galbraith, C. S., "IPO Performance in Business to Business "B2B" E-Commerce Firms: Effects of Strategy and Industry
[13] [Lee 2006] Lee, H. L., Whang, S., "E-Business and Supply Chain Integration"
[14] [Aissi 2006] Aissi, S., Malu, P., Srinivasan, K., "E-Business Process Modeling: The Next Big Step," Intel Labs
[15] [Papazoglou 2001] Papazoglou, M. P., "Agent-oriented technology in support of e-business enabling development of "intelligent " business agents for adaptive reusable software"
[16] [Geoffrion 2006 a] Geoffrion, A. M., Anderson, J. E., Krishnan, R., "Prospects for Operations Research in the E-
[17] Business Era"
[18] [Ropers 2006] Ropers, S., "New Business Models for the Mobile Revolution"
[19] [Kim 2004] Kim, J., Lee, I., Lee, Y., Choi, B., "Exploring e-business implications of the mobile internet: a cross-national survey in Hong Kong, Japan and Korea," Int. J. Mobile Communication, Vol. 2, No. 1, 2004
[20] [Camponovo 2006] Camponovo, G., Pigneur, Y., "Business model análisis applied to mobile business"
[21] [Giaglis 2006] George M. Giaglis, G. M., Pateli, A., Fouskas, K., Kourouthanassis, P., Tsamakos, A., "On the Potential Use of Mobile Positioning Technologies in Indoor Environments"
[22] [Losada 2005] Losada, S., Corcho, O., Contreras, J., Montes, M. M., Luis, J., Sergio, B., Richard, B., Ribas, J., DIP Data, Information and Process Integration with Semantic Web Services FP6 – 507483, WP 10: Case study eBanking D10.4 WSMO descriptions of Application, June 29th, 2005
[23] [Osterwalder 2002] Osterwalder, A., "An e-Business Model Ontology for Modeling e-Business," 15th Bled Electronic Commerce Conference e-Reality: Constructing the e-Economy, Bled, Slovenia, June pp 17 - 19, 2002
[24] [Gordijn 2006] Gordijn, J., Akkermans, H., "Designing and Evaluating E-Business Models"
[25] [Gisolfi 2001] Gisolfi, D., "Web services architect, Part 1: An introduction to dynamic e-business Search Advanced Help"
[26] [Porter 2006] Porter, M. E., "Strategy and the Internet"
[27] [Damanpour 2006] Damanpour, F., "E-business E-commerce Evolution: Perspective and Strategy"
[28] [Bradshaw 2006] Bradshaw, D., Brash, C., "Managing customer relationships in the e-business world: how to Personalise computer relationships for increased profitability"
[29] [Gupta 2001] Gupta, A., "Living applications," Enterprise Systems Journal, Feb 2001, Vol. 16, pp 22-27
[30] [Akkermans 2006] Akkermans, H., "Intelligent E-Business: From Technology to Value"
[31] [Kauffman 2003] Kauffman, R. J., Wood, A., "Revolutionary research strategies for e-business management: A philosophy of science perspective for research design and data collection in the age of the internet"
[32] [Bayer 2000] Bayer, F., Junginger, S., Kühn, H., "A Business Process-Oriented Methodology for Developing E-Business Applications," Proceedings of the 7th European Concurrent Engineering Conference (ECEC'2000), April 17-19, 2000, Leicester. Society for Computer Simulation (SCS). pp. 32-40
[33] [Rygielski 2006] Rygielski, C., Wang, J., Yen, D. C., "Data mining techniques for customer relationship management"
[34] [Massey 2006] Massey, A. P., Montoya-Weiss, M. M., Holcom, K., "Re-engineering the customer relationship: leveraging knowledge assets at IBM"
[35] [Romano 2002] Romano, N. C., Fjermestad, J., "Electronic Commerce Customer Relationship Management: An Assessment of Research"
[36] [Grönroos 2006] Grönroos, C., "From Marketing Mix to Relationship Marketing Towards a Paradigm Shift in Marketing"
[37] [Zeidler 2001] Zeidler, W. C., "Employing eBusiness practices through outsourcing," Ohio CPA Journal, Columbus, Jan-Mar 2001

[38] [Schubert 2001] Schubert, P., Häusler, U., "E-Government meets E-Business: A Portal Site for Startup Companies in Switzerland," Proceedings of the Hawai'I International Conference On System Sciences, January 3-6, 2001, Maui, Hawaii.
[39] [Ono 2001] Ono, Y., "Outsourcing Business Service and the Scope of Local Markets"
[40] [Hammer 2000] Hammer, J., Huang, C. B., Huang, Y. H., "The IDEAL Approach to Internet-Based Negotiation for E-Business"
[41] [Kumar 2006] Kumar, M., Feldman, S. I., "Business negotiations on the Internet"
[42] [Beam 2006] Beam, C., Segev, A., Haas, W. A., "Automated Negotiations: A Survey of the State of the Art"
[43] [Luo 2001] Luo, X., "Trust production and privacy concerns on the Internet A framework based on relationship marketing and social exchange theory"
[44] [Barakat 2006] Barakat, P. H., Garris, S., "Principles of Marketing An Independent Learning Course"
[45] [Tatsiopoulos 2006] Tatsiopoulos, I. P., Ponis S. T., Hadzilias E. A., Panayiotou N. A., "Realisation of the Virtual Enterprise Paradigm in the Clothing Industry through E-Business Technology"
[46] [Johnson 2002] Johnson, M. E, Whang, S., "E-business and supply chain management:An overview and framework", Production and operations management, Vol. 11, No.4, Winter 2002, USA
[47] [Chopra 2006] Chopra, S., Mieghem, J. A. V., "Which E-businessis right for suplí Caín?"
[48] [Phan 2002] Phan, D. D., "E-business development for competitive advantages: a case study," Information & Management 40 (2003) 581–590
[49] [Barua 2006] Barua, A., Konana, P., Whinston, A., Yin, F., "Making E-Business Pay: Eight Key Drivers for Operational Success"
[50] [Geoffrion 2006 b] Geoffrion, A. M., Krishnan, R., "E-Business and Management Science: Mutual Impacts (Part 1 of 2)"

Reviewers by Countries

Argentina
Ovando, Gabriela P.; Universidad Nacional de Rosario
Rossi, Gustavo; Universidad Nacional de La Plata

Canada
Feng, Jing; University of Ottawa
Ollivier-Gooch, Carl; The University of British Columbia
Shewchenko, Nicholas; Biokinetics and Associates
Steffan, Gregory; University of Toronto

Czech Republic
Kala, Zdenek; Brno University of Technology

Finland
Lahdelma, Risto; University of Turku
Salminen, Pekka; University of Jyvaskyla

Germany
Accorsi, Rafael; University of Freiburg
Glatzer, Wolfgang; Goethe-University
Gradmann, Stefan; Universitat Hamburg
Klamma, Ralf; RWTH Aachen University
Wurtz, Rolf P.; Ruhr-Universitat Bochum

Greece
Katzourakis, Nikolaos; Technical University of Athens
Bouras, Christos J.; University of Patras and RACTI

Italy
Badia, Leonardo; IMT Institute for Advanced Studies
Carpaneto, Enrico; Politecnico di Torino

Japan
Hattori, Yasunao; Shimane University

Netherlands
Mills, Melinda C.; University of Groningen
Pires, Luís Ferreira; University of Twente

New Zealand
Anderson, Tim; Van Der Veer Institute

Portugal
Cardoso, Jorge; University of Madeira
Natividade, Eduardo; Polytechnic Institute of Coimbra
Oliveira, Eugenio; University of Porto

Republic of Korea
Ahn, Sung-Hoon; Seoul National University

Singapore
Tan, Fock-Lai; Nanyang Technological University

Spain
Barrera, Juan Pablo Soto; University of Castilla
Gonzalez, Evelio J.; University of La Laguna
Perez, Juan Mendez; Universidad de La Laguna
Royuela, Vicente; Universidad de Barcelona
Vizcaino, Aurora; University of Castilla-La Mancha

Sweden
Johansson, Mats; Royal Institute of Technology

Switzerland
Pletka, Roman; AdNovum Informatik AG
Rizzotti, Sven; University of Basel
Specht, Matthias; University of Zurich

Taiwan
Lin, Hsiung Cheng; Chienkuo Technology University

Turkey
Ozalp, A. Alper; Uludag University

United Kingdom
Ariwa, Ezendu; London Metropolitan University
Biggam, John; Glasgow Caledonian University
Dorfler, Viktor; Strathclyde University
Kolovos, Dimitrios S.; The University of York
Vetta, Atam; Oxford Brookes University

USA
Bach, Eric; University of Wisconsin
Bazarian, Jeffrey J.; University of Rochester School
Bolzendahl, Catherine; University of California
Bussler, Christoph; Cisco Systems, Inc.
Charpentier, Michel; University of New Hampshire
Chester, Daniel; Computer and Information Sciences
DeWeaver, Eric; University of Wisconsin - Madison
Ellard, Daniel; Network Appliance, Inc
Gaede, Steve; Lone Eagle Systems Inc.
Gill, Sam; San Francisco State University
Gustafson, John L..; ClearSpeed Technology
Hunter, Lynette; University of California Davis
Iceland, John; University of Maryland
Kaplan, Samantha W.; University of Wisconsin
Koua, Etien L.; The Pennsylvania State University
Lagunas-Solar, Manuel; University of California Davis
Langou, Julien; The University of Tennessee
Liu, Yuliang; Southern Illinois University Edwardsville
Lok, Benjamin; University of Florida
Minh, Chi Cao; Stanford University
Morrissey, Robert; The University of Chicago
Mui, Lik; Google, Inc
Rizzo, Albert ; University of Southern California
Rosenberg, Jonathan M. ; University of Maryland
Shaffer, Cliff ; Virginia Tech
Sherman, Elaine; Hofstra University
Snyder, David F.; Texas State University
Song, Zhe; University of Iowa
Yu, Zhiyi; University of California

Venezuela
Candal, Maria Virginia; Universidad Simon Bolívar

IPSI Team

Advisor for Mathematics-Related Problems:
Marija Radovic and Visnja Tarbuk

Advisors for IPSI Tutorials:
Tanja Kovacevic and Tanja Petrovic

Advisors for IPSI Research:
Milos Kovacevic and Milos Milovanovic

Advisors for IPSI Developments:
Zoran Babovic and Darko Jovic

Welcome to IPSI BgD Conferences and Journals!

http://www.internetconferences.net
http://www.internetjournals.net

VIPSI-2007 VENICE SPRING
Venice, Italy
March 19 to 22, 2007

VIPSI-2007 AMALFI
Hotel Santa Caterina,
Amalfi, Italy
March 22 to 25, 2007

VIPSI-2007 TOKYO
M.I.T., Tokyo, Japan
May 31 to June 3, 2007

VIPSI-2007 CROATIA - OPATIJA
Opatija/Abbazia
Villa Ariston
June 7 to 10, 2007

VIPSI-2007 CROATIA - ROVINJ
Rovinj/Rovigo
Villa Angelo d'Oro
June 10 to 13, 2007

VIPSI-2007 MONTREAL
Campus of UQAM,
Montreal, Quebec, Canada
June 29 to July 02, 2007

VIPSI-2007 ITALY
Rome and Hotel Castello Chiola,
Loreto Aprutino in Abruzzo,
relatively near Rome, Italy
August 20 to 23, 2007

VIPSI-2007 FLORENCE
Grand Hotel & La Pace,
in Montecatini Terme,
near Florence, Piza, Siena, and Lucca, Italy
Aug 23 to Aug 26, 2007

VIPSI-2007 BELGRADE
Belgrade, Serbia
September 3 to 6, 2007

VIPSI-2007 MONTENEGRO MOUNTAIN SAFARI
Villa Bianca,
Kolasin, Montenegro
September 6 to 9, 2007

VIPSI-2007 MONTENEGRO SVETI STEFAN
Hotel Sveti Stefan
September 9 to 15, 2007

VIPSI-2007 SLOVENIA
Lake Bled in Alps,
Slovenia,
October 8 to 11, 2007

VIPSI-2007 VENICE FALL
Venice, Italy
October 11 to 14, 2007

VIPSI-2007 PORTOFINO
Grand Hotel Miramare,
Santa Margherita Ligure,
Italy (near Genova),
October 14 to 17, 2007

The publication of this journal
is supported in part by
the Ministry of Science of Serbia

CIP – Katalogizacija u publikaciji
Narodna biblioteka Srbije, Beograd

ISSN 1820 – 4503 =
The IPSI BGD Transactions on Internet Research
COBISS.SR - ID 119127052