ADVANCES IN INTELLIGENT SYSTEMS AND COMPUTING 224

Wojciech Zamojski
Jacek Mazurkiewicz
Jarosław Sugier
Tomasz Walkowiak
Janusz Kacprzyk (Eds.)

New Results in Dependability and Computer Systems

Proceedings of the 8th International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, September 9–13, 2013, Brunów, Poland

Springer

Advances in Intelligent Systems and Computing

Volume 224

Series Editor

J. Kacprzyk, Warsaw, Poland

For further volumes:
http://www.springer.com/series/11156

Wojciech Zamojski · Jacek Mazurkiewicz
Jarosław Sugier · Tomasz Walkowiak
Janusz Kacprzyk
Editors

New Results in Dependability and Computer Systems

Proceedings of the 8th International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, September 9–13, 2013, Brunów, Poland

Springer

Editors
Wojciech Zamojski
Institute of Computer Engineering,
Control and Robotics
Wrocław University of Technology
Wrocław
Poland

Tomasz Walkowiak
Institute of Computer Engineering,
Control and Robotics
Wrocław University of Technology
Wrocław
Poland

Jacek Mazurkiewicz
Institute of Computer Engineering,
Control and Robotics
Wrocław University of Technology
Wrocław Poland

Janusz Kacprzyk
Polish Academy of Sciences,
Systems Research Institute
Warszawa
Poland

Jarosław Sugier
Institute of Computer Engineering,
Control and Robotics
Wrocław University of Technology
Wrocław
Poland

ISSN 2194-5357 ISSN 2194-5365 (electronic)
ISBN 978-3-319-00944-5 ISBN 978-3-319-00945-2 (eBook)
DOI 10.1007/978-3-319-00945-2
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013940156

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

We are pleased and honoured to present the proceedings of the Eight International Conference on Dependability and Complex Systems DepCoS-RELCOMEX which was held in a beautiful Brunów Palace, Poland, from 9^{th} to 13^{th} September, 2013.

DepCoS – RELCOMEX is an annual conference series organized by the Institute of Computer Engineering, Control and Robotics (CECR), Wrocław University of Technology, since 2006. Its idea came from the heritage of the other two cycles of events: RELCOMEX (1977–89) and Microcomputer Schools (1985–95) which were organized by the Institute of Engineering Cybernetics (the previous name of CECR) under the leadership of prof. Wojciech Zamojski, now also the DepCoS chairman. In contrast to those previous events focused on the classical reliability analysis, the DepCoS mission is to promote a more comprehensive approach which in the new century has earned the name *dependability*.

Contemporary technical systems are integrated unities of technical, information, organization, software and human (users, administrators and management) resources. Their complexity stems not only from involved technical and organization structures (comprising both hardware and software resources) but also from complexity of information processes (processing, monitoring, management, etc.) realized in their operational environment. With system resources being dynamically allocated to the on-going tasks, a flow of system events (incoming and/or on-going tasks, decisions of a management system, system faults, defensive system reactions, etc.) may be modelled as a deterministic or/and probabilistic event stream. Complexity and multiplicity of processes, their concurrency and their reliance on the in-system intelligence (human and artificial) significantly impedes the construction of strict mathematical models and limits evaluation of adequate system measures. In many cases, analysis of modern complex systems is confined to quantitative studies (e.g. Monte Carlo simulations) which prevents development of appropriate methods of system design and selection of policies for system exploitation. Security and confidentiality of information processing

introduce further complications into the system models and the evaluation methods.

Dependability tries to deal with all those challenges by employing a multidisciplinary approach to theory, technology and maintenance of systems working in a real (and very often unfriendly) environment. As opposed to "classic" reliability which focuses mainly on technical system resources (components and structures built form them), dependability studies investigate the system as a multifaceted and sophisticated amalgamation of technical, information and also human resources concentrating on efficient realization of tasks, services and jobs in such an environment. Traditional methods of reliability evaluation focused mainly on technical levels are usually insufficient and more innovative methods of dependability analysis, often based on intelligent and soft computing algorithms, need to be applied. The 50 submissions selected for this volume illustrate the wide diversity of problems that need to be explored, for example methodologies and practical tools for modelling, design and simulation of the systems, security and confidentiality in information processing, specific issues of heterogeneous, today often wireless, computer networks, management of transportation networks, etc.

In the closing words of this introduction we would like to emphasize the role of all reviewers whose support helped to refine the contents of this volume. The following people took active part in the evaluation process of the conference submissions: Salem Abdel-Badeeh, Andrzej Białas, Frank Coolen, Manuel Gil Perez, Zbigniew Huzar, Vyacheslav Kharchenko, Alexey Lastovetsky, Marek Litwin, Jan Magott, István Majzik, Jacek Mazurkiewicz, Yiannis Papadopoulos, Oksana Pomorova, Krzysztof Sacha, Miroslaw Siergiejczyk, Ruslan Smeliansky, Janusz Sosnowski, Jarosław Sugier, Victor Toporkov, Tomasz Walkowiak, Max Walter, Bernd E. Wolfinger, Marina Yashina, Wojciech Zamojski, and Włodzimierz Zuberek.

Finally, we would like to express our sincere gratitude to the authors of all the works selected for publication – we hope that their submissions will be interesting to scientists, researchers, practitioners and students who investigate dependability problems in computer systems and networks, and in their diverse applications.

The Editors

Eight International Conference on Dependability and Complex Systems DepCoS-RELCOMEX

organized by

Institute of Computer Engineering, Control and Robotics,
Wrocław University of Technology
under the auspices of prof. Tadeusz Więckowski, Rector

Brunów Palace, Poland, 9–13 September 2013

Program Committee

Wojciech Zamojski (Chairman)
Wrocław University of Technology, Poland

Ali Al-Dahoud
Al-Zaytoonah University, Amman, Jordan

Salem Abdel-Badeeh
Ain Shams University Abbasia, Cairo, Egypt

George Anders
University of Toronto, Canada

Włodzimierz M. Barański
Wrocław University of Technology, Poland

Andrzej Białas
Institute of Innovative Technologies EMAG, Katowice, Poland

Dariusz Caban
Wrocław University of Technology, Poland

Krzysztof Cios
Virginia Commonwealth University, Richmond, USA

Frank Coolen
Durham University, UK

Antonio Ferrari
University of Aveiro, Portugal

Francesco Flammini
University of Naples "Federico II", Italy

Manuel Gill Perez
University of Murcia, Spain

Janusz Górski
Gdańsk University of Technology, Poland

Franciszek Grabski
Naval University, Gdynia, Poland

Zbigniew Huzar
Wrocław University of Technology, Poland

Igor Kabashkin
Transport and Telecommunication Institute, Riga, Latvia

Janusz Kacprzyk
Polish Academy of Sciences, Warsaw, Poland

Organizing Committee

Contents

Application Level Execution Model for Transparent Distributed Computing

Razvan-Mihai Aciu and Horia Ciocarlie

Department of Computer and Software Engineering, "Politehnica" University of Timisoara
Blvd Vasile Parvan, Nr. 2, Postcode 300223, Timisoara, Romania
razvanaciu@yahoo.com, horia@cs.upt.ro

Abstract. Writing a distributed application involves using a number of different protocols and libraries such as CORBA, MPI, OpenMP or portable virtual machines like JVM or .NET. These are independent pieces of software and gluing them together adds complexity which can be error prone. Still, some issues such as transparent creation and synchronization of the parallel distributed threads, code replication, data communication and hardware and software platform abstraction are not yet fully addressed. For these reasons a programmer must still manually handle tasks that should be automatically and transparently done by the system. In this work we propose a novel computing model especially designed to abstract and automate the distributed computing requirements ensuring at the same time the dependability and scalability of the system. Our model is designed for a portable virtual machine suitable to be implemented both on hardware native instruction set as well as in other virtual machines like JVM or .NET to ensure its portability across hardware and software frameworks.

1 Introduction

Distributed computing is a domain with intense research and applicability in many areas like biology [1-2], physics [3-4], agriculture [5], computing [6]. As the computer networks are increasingly popular, there are more and more possibilities to access data storage and computation power which allows solving of problem types earlier accessible only to supercomputers or dedicated data centers. The main parallelization opportunities today are represented by CPUs with multiple cores, distributed computing in a network and specialized computing using GPUs.

We focus mainly on showing a reliable model for transparent distributed computing, capable to handle by itself almost all the low level details needed by network code replication, data communication and distributed thread synchronization. We demonstrate that our model insures hardware and software abstraction, it scales well with the available computing resources and it is sufficiently general to handle other aspects involved in heterogeneous distributed computing such as different operating systems or abstracting the execution on CPU cores or in a network.

W. Zamojski et al. (Eds.): *New Results in Dependability & Comput. Syst.*, AISC 224, pp. 1–10.
DOI: 10.1007/978-3-319-00945-2_1

For applications that run on a single machine, the task of writing a parallel algorithm is simplified by the fact that many aspects involved are the same, for example the memory layout and access, threads creation and synchronization, computing units binary code. There are some models and libraries like OpenMP that address this situation using special preprocessor instructions or annotations to instruct the compiler to automatically parallelize a loop [7-8]. At the same time many modern additions to standard libraries provide high level concepts for parallel computing, for example thread pools.

When the same algorithm is implemented for distributed computing new problems arise. We mention different data layout and instruction, different operating systems and possible network failures. To address the above issues different standards and libraries were proposed: CORBA, MPI, Java RMI, DCOM and Ibis [8]. Most of these are low level protocols that try to hide the distributed platform differences, but they are not sufficiently high level to hide from the programmer details like code replication or data synchronization [8-9].

Another fundamental issue for any distributed system is its scalability, due to the fact that the system should make an optimal use of all its computing resources [10]. Heterogeneous resources are available in many different versions, speeds or instruction sets. This makes hard for the programmer to make an optimal use of these resources, in order that their combined workload to lead to a minimal application processing time or to another desired target [11]. At the same time some resources can have a dynamic behavior, because they can be added or removed from the system by request or due to network errors.

When analyzing these aspects, it can be seen that most of the work involved in writing distributed software is in fact necessary to address repetitive and standard tasks. All these tasks can be automatically addressed by using a proper infrastructure and model. Such an application level model greatly improves the software dependability. We present such a model, motivate our design decisions and show the results of one of its possible practical implementations.

2 Application Level Distributed Computing Model

Ideally speaking, a transparent application level model should hide from the programmer any low level task. At the same time, it should fit with only minimal additions to the existing programming languages and frameworks, so it can be easily implemented. Our model involves only three concepts, close to OOP programming style and it should seem familiar to any programmer with an OOP background. We will present it by using an example.

We chose to implement the well known Mandelbrot set on a given interval. The example is written in a C++-like language and shows the relevant code for our model:

```
  unit MandelbrotLine{
   double  xMin,xMax;
   int     width,maxIterations;
   MandelbrotLine(double xMin,double xMax,int width,int maxItertions)
   {
      this->xMin=xMin;
      his->xMax=xMax;
      this->width=width;
      this->maxIterations=maxIterations;
   }
   string     run(double y)
   {
      int             pixel,iteration;
      stringstream    r;
      double          xi,yi,xb,xtemp;
      for(pixel=0;pixel<width;pixel++){
         xb=xMin+pixel*(xMax-xMin)/width;
         xi=yi=0;
         iteration=0;
         while(xi*xi+yi*yi<2*2&& iteration<maxIterations){
            xtemp=xi*xi-yi*yi+xb;
            yi=2*xi*yi+y;
            xi=xtemp;
            iteration++;
            }
         r<<iteration%256<<" ";    //to gray tones
         }
      return r.str();
   }
}
#define WIDTH          1000
#define HEIGHT         1000
#define MAX_ITER       10000
int     main()
{
int          lineIdx;
double       yLine;
string       img[HEIGHT];
double       xMin=0.33072017,xMax=0.33925741;
double       yMin=0.04369091,yMax=0.0522281593;
with(img;MandelbrotLine(xMin,xMax,WIDTH,MAX_ITER)){
   for(lineIdx=0;lineIdx<HEIGHT;lineIdx++){
      yLine=yMin+lineIdx*(yMax-yMin)/HEIGHT;
      run[lineIdx](yLine);
      }
   }
ofstream     file("mandelbrot.pgm");
file<<"P2"<<endl<<WIDTH<<","<<HEIGHT<<endl<<"255"<<endl;
for(lineIdx=0;lineIdx<HEIGHT;lineIdx=lineIdx+1)
   file<<img[lineIdx]<<endl;
}
```

Fig. 1 Model example for a distributed algorithm

For each image line a new invocation is created. An invocation encapsulates all data needed for a single computation and it is asynchronously run on a separate distributed thread when a computing resource becomes available.

In our model, a thread is always associated with a resource (the existent cores in network) and the maximum number of threads is at most equal to the number of resources. In this way we eliminate the unnecessary task switching and at the same time we use all the resource at their maximum capacity. If the number of invocations exceeds the available resources, the invocations are put in a queue, waiting for resources to become available. We use the term "invocation" for a scheduled computation and the term "thread" for running invocation.

The example program computes distributedly all the invocations, waits for all of them to complete and writes the results into a file. We used three special concepts to make this program distributed. These concepts are detailed below, each one with its own semantics and requirements.

1) The "unit" concept: is the main encapsulation block for an invocation. Like a regular class from the OOP languages it can contain attributes and methods. When used, a "unit" can be run locally, on the same machine, or it can be transparently sent to another computer from the network. There are some significant differences between a "unit" and a regular class. These differences and their rationale are as follows:

The "unit" constructors are used to set the initial state for all the created threads. The constructors are called in the "with" statement (explained in II.*B*) and the same data is used to initialize all the threads. In our example all the threads have the same *x*-axis interval [*xMin, xMax*], the horizontal resolution *width* and the maximum number of iterations, *maxIterations.*

The "unit" "run" methods are used to specify the code for threads. Their parameters are initialized from the scheduled invocations and their result is returned to the caller after the thread ends. In our example every thread needs only its *y* position on the y-axis to compute a particular line.

The main difference between the constructor data and the invocation data comes from the fact that the constructor data is the same for all invocations, so it can be sent only once for every machine, no matter how many invocations will run on that specific machine. Instead, an invocation data needs to be specifically sent for each thread.

A "unit" can use extern functions or other types such as classes, but it cannot access, directly or indirectly, extern data. If it accesses functions or classes, these are automatically packed with it and sent to the remote machine. The rationale for not allowing extern data access is that it would require a lot of slow and unreliable network traffic, including data serialization and synchronization with other threads. If the programmer will try to use from a remote "unit" some external data in the same way as a regular local data, this will greatly slowdown the entire thread and will also create a bottleneck for all the other (possibly remote) threads waiting to access that data. Because of this, it was chosen for every thread to be able to access only its own data and this data is sent along with it.

2) The "with" concept: is an encapsulation block for creating and synchronizing invocations. "with" accepts two or three parameters, separated by semicolons. The first one is a destination for the threads results, the second is a "unit" constructor and the third one, which is optional, is a set of flags to control different aspects of the statement.

The destination can be an n-dimensional array, an object with a special interface or a function/closure. In case of an array, it will hold after the execution of the "with" block at the positions given by the invocations ordinals (explained at the "for" concept) the results of the invocations. If objects are used as destination, they must implement a specific interface with a handler method. In case of functions/closures, they will receive the invocations results and ordinals as parameters.

The "unit" constructor is used to specify what "unit" will be used (only one for a "with" statement) and its initial data. This data will be the same for all threads creation. The optional "flags" parameter is a set of flags used to specify different aspects, for example the restriction to use only local cores (no network traffic), to enable/disable the GPU processing, etc.

"with" ensures on its end the computation of all invocations and the synchronization of all the running threads, waiting for them to complete, similar with the "join" functions used in multithreading programming. The underlying framework is also responsible with the "unit" code replication in network, data serialization for invocations parameters and results and automatic rerun for computations lost due to network errors. The "with" statement also acts as a threads pool and it monitors the status of the processing units (CPU, GPU, network computers), assigning new invocations to them when there are free resources.

3) The "run" concept: can be used only inside "with" and consists of two lexical parts. The first part is represented like an n-dimensional array access and it is used to specify the ordinals of the invocation. The second part is represented like a function call and it is used to specify invocation specific parameters to be passed to the unit "run" methods.

The invocation ordinals are used to specify unique ids for every invocation. Our Mandelbrot example uses only one ordinal, which is the index of the line returned by the invocation. The parameters specified in the second part of the "run" statement will be used on that specific invocation run.

The "run" statement creates and stores an invocation for the threads pool provided by the encapsulating "with". In this way the system can optimally choose how to run the invocations, one by one or in batches (for GPUs).

3 Model Performance

To assess the theoretical performance of our model, we consider N_C computers in the distributed system with a total of N_P processing units (cores) which need to execute a number of N_I invocations ($N_P \leq N_I$), each invocation requiring a T_I time to complete and T_S is the time to setup one computer (send the "unit" to it):

$$T_T = T_S + \lceil N_I / N_P \rceil * T_I, \tag{1}$$

where T_T is the total computation time and $\lceil x \rceil = \min\{n \in \mathbb{Z} \mid n \geq x0\}$ (2)

$$\text{if } N_P \to N_I: \quad \lim_{N_P \to N_I} T_T = T_S + T_I \tag{3}$$

In that case the total execution time for all distributed threads is the execution time of the longest thread (if there are heterogeneous resources) and the setup time for

the remote computers, which depends on factors like the size of the "unit" data and network usage.

As T_I can also be seen as $T_I=T_{IN}+T_{IC}$ (4)

where T_{IN} is the time needed for network operation (sending the "run" parameters to the processing unit and receiving the results from it) and T_{IC} is the effective computation time on the processing unit, in the best case

$$(N_P=N_I): T_T=T_S+T_{IN}+T_{IC} \quad (5)$$

Because T_S+T_{IN} depends only on the network performance, for optimal results it is best that it's weight with respect to the overall computing time to be as small as possible. The optimal case is when the T_{IC} of the distributed threads is much larger than T_S+T_{IN}, so the distributed system spends most of its time in doing the effective computation than on network traffic. In this case, the performance of the distributed computation becomes close to the performance of executing all the invocations locally on a machine with N_I cores.

4 Implementation and Study

We implemented our model by developing a special virtual machine (VM) and associated runtime, capable to run the VM on CPU cores. For every machine all the computing resources are abstracted using a server which can receive a "unit" and invocations to be run on it. An application runs as a client and it makes requests to the available servers when it needs to run multiple threads. The whole process takes place according to the following steps:

1) The client checks for the available servers: A list of network hosts is used and every machine is queried about its server version, protocol and the number of available cores.

2) The client runs the application: We designed a register based, strongly typed VM with high level abstractions like functions and classes and automatic memory management. Having a portable VM which acts like an abstraction layer between the application and the host available capabilities (native execution using the CPU instruction set or execution on a particular VM like JVM or .NET) allows us to use almost any computer, ensuring both hardware and software framework independence.

A register based VM is also very important for running threads on GPU cores. In order to do this in a portable way, the GPU driver (CUDA or OpenCL) must receive a kernel function written in a C/C++ like language. In our case, the functions from inside a "unit" must be recompiled from the VM opcodes to the required language and using a register based VM makes this job easy.

3) When the application enters a "with" statement, the runtime is invoked to create a scheduler: The scheduler receives the unit constructor parameters, the receiver and possible options. The constructor parameters are serialized only once, in the beginning, as they remain constant.

A scheduler creates when needed a number of worker threads, each one responsible with the connection to a computing resource on a server. The worker communicates with the server using a socket which is maintained open during all the worker life. On the server side a new connection object is created in a separate thread for each worker, so every connection will run in a CPU core.

Before handling invocations, a worker makes sure that the needed code (the "unit" code and all its dependencies) are available on server. The server is able to cache all the code sent to it, so a "unit" must be sent only once to each server. The arguments for the "unit" constructor are also sent for each worker.

4) On a "run" statement, a new invocation is added: The scheduler keeps a list of all the invocations. The invocations are added asynchronously. When an invocation is added, the scheduler checks if there are available workers for that invocation and if not tries to create a new one using the list of the available servers. At most, the total number of workers can be equal with the total number of cores in all servers from the list.

5) When a worker is free, it processes an invocation from the invocations list: On server a new VM instance is created on the worker's connection. The "unit" constructor is called with the initialization data already on server so a new "unit" instance will be available.

The worker sends the specific "run" method signature (to allow method overloading) and its parameters. The "run" method is run on server, isolated on the connection's specific VM instance.

The server serializes the "run" results and returns them to the worker. If an error occurs during the processing (for example network errors), the worker puts back the invocation in the invocations list for reprocessing. If there is no error, the results are put/sent to the "with" receiver.

In this way, the workers will take invocations from list, run them remotely and put the results where needed. The process continues until there are no invocations left for processing.

6) At the end of the "with" statement, the scheduler waits for the completion of all invocations: For this the invocations list must be empty and all workers must have ended their current jobs. After all the invocations are processed, all workers are ended and the scheduler is disposed. On the server side, the resources allocated to the connections are also disposed.

5 Experimental Results

We tested the implementation performance and scalability both on a processor's local cores and in a network of 10 computers. Our test program is the one from Fig. 1 with HEIGHT=2000, so we have a total of 2000 invocations. The program implementation in our VM resulted in a code set of about 1.1KB and every set were run with a clean server, so on each run the servers invocation setup needed to be complete, by sending each time the required application code to them.

For each test we measured the speedup from the case with 1 core or 1 computer to test the scalability of the distributed computing system and the workload on each core, measured in the percent amount of distributed invocations computed on

that core, to evaluate the implementation capacity to distribute evenly the invocations on all available processing units.

The best scalability would be if the speedup is equal with the number of computers/cores involved in computation from the case of performing the computation on only one computer/core. The best workload distribution would be if all the invocations would be distributed completely equal on all the available processing units, assuming that all the processing units have equal capabilities.

5.1 Tests on a Computer Network

We used a Wi-Fi network of 10+1 computers with 2 cores each and all cores were used. The invocations were allowed to run only on remote computers and one computer was used only to run the main application, so all the created threads can run in equal conditions. We started with one computer and on each test added another computer to the available servers. The results for speedup can be seen in Fig. 2 and for the workload on every core from all computers in Fig. 3.

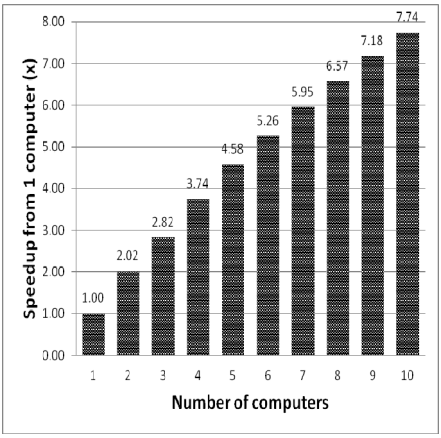


Fig. 2 Speedup results on network

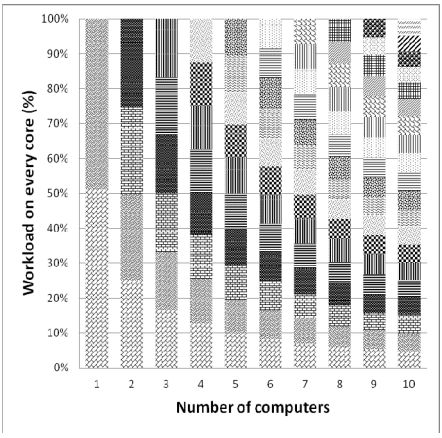


Fig. 3 Workload results on a computer network

When the number of computers is small, so the amount of total computation on each computer is high, the speedup when adding a new one is very close to the optimal case. When the number of computers is higher, the network setup and traffic time, which are constant, starts to have a significant proportion, so the speedup is smaller. This is consistent with our theoretical model performance presented in Section III-D.

Even if we used a Wi-Fi network with lower reliability, the system did a good job in allocating an equal number of invocations on every core. From our test results, the maximal percent difference from the optimum was 13.6%, with an average percent difference of maximum 5% for all test runs.

5.2 Tests on a Computer Cores

We used a computer with 4 cores. We started with one core and on each test we added another core. The results can be seen in Fig. 4 for speedup and in Fig. 5 for the workload on every core.

Because on running invocations only in the local processor cores there is no network traffic involved, the T_S+T_{IN} term from our theoretical model is 0 and only a smaller overhead of threads synchronization is involved. In this case, the speedup has a linear grow, close to the optimal case, both in the beginning and at the ending of the graphic.

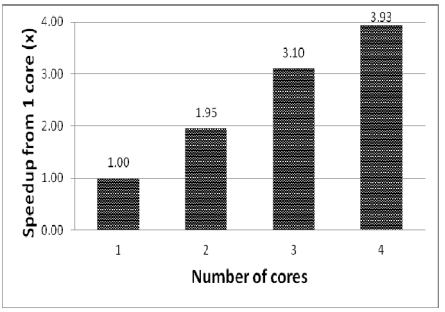


Fig. 4 Speedup results on a computer cores

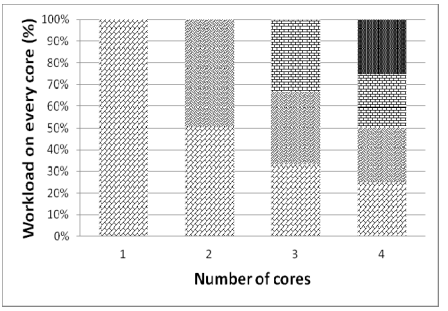


Fig. 5 Workload results on a computer cores

Even if one core was also used to run the main application, the implementation succeeded to distribute the invocations on every core close to the optimal case. From our test results, the maximal percent difference from the optimum was 0.6%, with an average percent difference of maximum 0.6% for all test runs. The small difference from the optimal workload even if one core also runs the "with" scheduler, proves that it mainly waits for the threads to complete, so it takes only very few computing resources.

6 Conclusion

We proposed an application level model for dependable transparent distributed computing. We showed that using only 3 concepts it is possible to model a large domain of distributed algorithms. The model semantics is close to the OOP, which makes the concepts easily to be implemented in most of the actual programming languages.

Our implementation using a portable VM shown that a scheduling system based on distributed thread pools can distribute the invocations computations across all of the available computing resources in a close to optimal manner, obtaining a well balanced workload. It has shown that different computing resources (CPU cores or network computers) can be abstracted using server/client and threads semantics.

The experimental results indicate that the model is scalable, because it succeeded to use well the available resources and the speedups achieved were close to performing the computations as by using multiple programs in parallel.

Even on network failures the application was able to run again the dropped invocations, in order to complete them, so the model is dependable.

We further consider developing our work on directions that include using GPUs as transparent computing resources, ensuring computation reliability and recovery from more possible errors and optimal scheduling algorithms for heterogeneous resources.

References

[1] Afek, Y., Alon, N., Barad, O., Hornstein, E., Barkai, N., Bar-Joseph, Z.: A Biological Solution to a Fundamental Distributed Computing Problem. Proc. Natl. Acad. Sci. USA 108(14), 5488–5491 (2011)

[2] Macía, J., Posas, F., Solé, R.V.: Distributed computation: the new wave of synthetic biology devices. Trends in Biotechnology 30(6), 342–349 (2012)

[3] Lawrenz, M., Baron, R., Wang, Y., Andrew McCammon, J.: Independent-Trajectory Thermodynamic Integration: A Practical Guide to Protein-Drug Binding Free Energy Calculations Using Distributed Computing. In: Computational Drug Discovery and Design Methods in Molecular Biology, vol. 819, pp. 469–486 (2012)

[4] Charbonneau, A., Agarwal, A., Anderson, M., Armstrong, P., Fransham, K., Gable, I., Harris, D., Impey, R., Leavett-Brown, C., Paterson, M., Podaima, W., Sobie, R.J., Vlie, M.: Data intensive high energy physics analysis in adistributed cloud. In: Journal of Physics: Conference Series, vol. 341 (2012)

[5] Polojärvi, K., Luimula, M., Verronen, P., Pahkasalo, M., Koistinen, M., Tervonen, J.: Distributed System Architectures, Standardization, and Web-Service Solutions in Precision Agriculture. In: GEOProcessing: The Fourth International Conference on Advanced Geographic Information Systems, Applications, and Services (2012)

[6] Jakovits, P., Srirama, S.N., Kromonov, I.: Stratus: A Distributed Computing Framework for Scientific Simulations on the Cloud. In: IEEE 14th International Conference High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems, HPCC-ICESS (2012)

[7] Larsen, P., Ladelsky, R., Karlsson, S., Zaks, A.: Compiler Driven Code Comments and Refactoring. In: MULTIPROG (2011)

[8] Henrio, L., Huet, F., Zsolt, I., Sebestyen, G.: Multi-active Objects. sop.inria.fr (2011)

[9] Lee, H.J., Brown, K.J., Sujeeth, A.K., Chafi, H., Olukotun, K., Rompf, T., Odersky, M.: Implementing Domain-Specific Languages for Heterogeneous Parallel Computing. IEEE Micro 31(5) (2011)

[10] Coulouris, G., Dollimore, J., Kindberg, T., Blair, G.: Distributed Systems - Concepts and Design, 5th edn. Addison-Wesley (2011)

[11] Korkhov, V.V., Moscicki, J.T., Krzhizhanovskaya, V.V.: The User-Level Scheduling of Divisible Load Parallel Applications With Resource Selection and Adaptive Workload Balancing on the Grid. IEEE Systems Journal 3(1) (March 2009)

Software Support of the Risk Reduction Assessment in the ValueSec Project Flood Use Case

Jacek Bagiński

Institute of Innovative Technologies EMAG,
40-189 Katowice, Leopolda 31, Poland
jbaginski@emag.pl

Abstract. The chapter presents information about the first stage of validation of the OSCAD tool for the risk reduction assessment within the decision support process. First, general information about risk management and risk assessment is given, and relations of the risk assessment with the flood issue are described. Basic information about the ValueSec project and its relations with risk assessment is presented. Next, the results of first experiments heading for OSCAD usage as one of the possible elements supporting the Risk Reduction Assessment (RRA) software pillar in the ValueSec project are described. The possibility of OSCAD usage for the RRA pillar was validated on the example of the so-called "flood use case" of the ValueSec project. This use case relates to the assessment and selection of flood countermeasures. The main objective of the validation is to find out if the risk assessment method implemented in OSCAD can be used for the flood issue.

1 Introduction

The risk management process and risk analysis related issues are nowadays quite well defined and established in standards. The ISO 31000 [1] and ISO 31010 [2] standards can be mentioned here which distinguish, among others, the risk assessment and risk treatment activities within the whole risk management process.

The same approach is proposed by ISO 27005 [3] which contains recommendations for the information security risk management.

The risk analysis and risk assessment aspects, as well as the risk level reduction are also elements of many other normative documents, standards and recommendations, for example standards for occupational health and safety management systems (OHSAS 18001 [4]) which relate also to such issues as threats identification, risk assessment and control.

W. Zamojski et al. (Eds.): *New Results in Dependability & Comput. Syst.*, AISC 224, pp. 11–24.
DOI: 10.1007/978-3-319-00945-2_2

Regardless of the application domain of the risk management and risk assessment, the basic definition of risk level is mostly described in the same way, as a combination of incident (undesirable event) consequences and the likelihood (probability) of its appearance:

$$R = C * L \tag{1}$$

This formula (1) can be extended with additional elements, depending on the analysis object and the results purposes. It can be supplemented for example with financial aspects or other parameters influencing the risk value, like security measures which reduce the risk level, their effectiveness, quality, etc.

The same approach is also used in the floodplain management programs, which must consider the risk management aspect.

The requirements on the risk assessment in the flood domain come out of the EU directive 2007/60/EC [5]. Due to the Directive requirements, the EU member countries started during the last few years to launch flood management programs with risk assessment as one of key elements. For example, the Scottish Environment Protection Agency (SEPA) [6] began The National Flood Risk Assessment program. Based on this program results, the document *Flood Risk Management Planning in Scotland: Arrangements for 2012 -2016* [7] was prepared.

Other countries started to implement the risk management process and risk assessment activities within the flood management programs. For example, on the 30th of July 2010, in Poland, the ISOK project began ("Informatyczny system osłony kraju przed nadzwyczajnymi zagrożeniami") [8], whose result was a report [9] with the preliminary assessment of flood risk in Poland.

An important element of the risk management process is the security measures (in case of flood – flood countermeasures) selection. The selection of appropriate security measures is a vital problem for the decision makers, who have to take efforts to keep their activities and decision making process transparent. Their decisions must be backed by evidences of relevance selected solutions. These evidences, in the case of security measures selection, may contain also risk analysis results about the estimated efficiency, effectiveness, reliability, economic factors, and other costs and benefits (like social, environmental, etc.).

The usage of software support for the security measures selection provides such transparency and possibility to define clearly risk assessment rules and to document the whole process. It also confirms that all necessary activities for the best variant selection were performed.

The implementation of a software tool for such support is one of the main goals of the ValueSec project. Its assumptions were described in more detail in [10] and on the project website [11]. Within the project there will be a tool developed which will support the decision making process. One of the test cases is the above mentioned process of security measures selection in case of a flood.

2 RRA in the Three Pillars Context

According to the assumptions worked out in the course of the project, while selecting the security measures it is necessary to take into account such factors as the risk reduction level offered by a security measure or a group of security measures, the relation between costs and benefits gained from the measure implementation, and the so called "qualitative criteria", i.e. the assessment of "unmeasurable" parameters and impacts, such as social satisfaction, cultural, political, environmental aspects, etc. [11].Therefore it was assumed the software supporting the security measures selection, developed within the project, should be based on three pillars:

- Risk reduction analysis (RRA),
- Cost-Benefit Analysis (CBA),
- Qualitative Criteria Analysis (QCA).

Within the project a tool will be developed that will support the operations of these three pillars. The validation of this tool will be performed in the five preliminarily chosen areas (contexts): public mass events, mass transportation, airport security, communal security (with a flood protection application scenario), cyber security

Due to the fact that the risk reduction analysis can be made using a number of risk analysis tools, it was assumed that for the implementation of the first pillar the existing tools will be used, while the CBA and QCA pillars will be fully implemented within the project. For the implementation of the RRA pillar, the following software tools were considered: Riger from ATOS, Lancelot from WCK, RAS from Fraunhofer Institute, OSCAD from EMAG.

The tools were assigned to particular application scenarios in each context with a view to check the correctness of the adopted solution about using a ready-to-use risk analysis tool. OSCAD was selected to fulfill the RRA pillar in the "Flood use case". "Use case" can be defined as the set of security measures, which contains the measures for evaluation in the ValueSec toolset.

3 Possibility of OSCAD Usage in RRA Pillar

The OSCAD system was developed at the Institute of Innovative Technologies EMAG within a project co-financed by the National Centre for Research and Development (NCBiR) [12]. The OSCAD software was developed to support the integrated system, consisting of a business continuity management (BCM) and information security management (ISM) system. Risk analysis is an important element of both these management systems.

The risk analyzer module implemented in OSCAD was worked out mainly based on the requirements and recommendations of such standards as BS 25999 [13], ISO 27001 [14] and ISO 27005 [3]. They describe requirements and recommendations for BCMS and ISMS. The possibilities of the OSCAD system and the

risk analysis methods have already been described in [15], [16], [17]. The adopted solution is partly based on the approach described in [18]. The adopted method of threats and vulnerabilities assessment corresponds to earlier works conducted in this domain in the EMAG Institute and described in [19], [20]. However, the method was modified and simplified.

The security attributes (confidentiality, integrity and availability) assessment was distinguished in the form of the Business Impact Analysis – BIA. The low level analysis in turn [18], called detailed analysis in OSCAD, does not take into account a part of parameters proposed in [18] and related to risk assessment. Thanks to that, the method used in OSCAD is simpler while the analysis can be performed quickly and without any program support (if there is no such possibility). Additionally, there is no analysis of some economic parameters (e.g. cost efficiency of security measures or return on investment). With OSCAD used for RRA in ValueSec, the lack of the economic parameters analysis is not an obstacle since this issue will be taken care of by the CBA pillar.

For the OSCAD tool a number of experiments and validations were performed, e.g. for the pharmaceutical business (delivery of medications, medical supplies) and the mining sector [21], but these experiments were not related to the flood domain and were oriented on a wider scope of security management than just risk management.

According to the EU directive [5] on the assessment and management of flood risks, for this type of risk management the Member States shall establish objectives, which will focus on the reduction of flood consequences (for human health, the environment, cultural heritage, economic activity) and/or reduction of flood likelihood. According to another directive requirement, during the risk analysis the existing security measures should be considered. This means that, the way of calculation used for the risk assessment in the OSCAD tool can be easily adapted to the risk assessment in the flood domain.

The OSCAD method uses the following formula for the risk calculation:

$$R = \frac{C * L}{SMi * SMta} * Pc, \tag{2}$$

where R means risk value, C and L mean Consequences and Likelihood of incident (flood) appearance. *SMi* and *SMta* parameters relate to existing or planned security measures. *Pc* parameter means value of process criticality, which can be assessed during the business impact analysis.

SMi and *SMta* were initially intended as security measures implementation level and technical advancement level values. But thanks to the possibility of free definition of values and descriptions of these parameters, they can be adapted to the flood domain needs. Such a possibility is provided by the mechanism of dictionaries implemented in OSCAD (Fig. 1). Based on the [9] *SMta* and *SMi* parameters can be defined for example as a class of the flood countermeasure structure, and the safety status of the flood countermeasure structure.

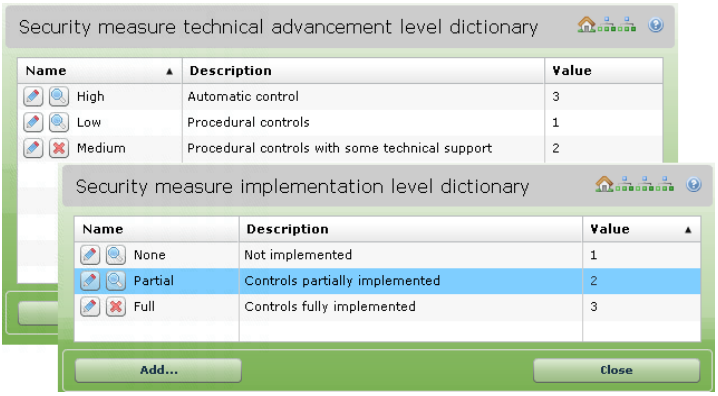


Fig. 1 Example of security measures assessment scales definition in OSCAD dictionaries

The same mechanism of dictionaries can be used to define and describe possible values of consequences (impacts) and likelihood (probability) of an adverse event. Information about required probability levels is also presented in the EU directive [5], which distinguishes three possible levels:

- floods with a low probability, or extreme event scenarios;
- floods with a medium probability (likely return period ≥ 100 years);
- floods with a high probability, where appropriate.

Assigning values to these levels in the dictionary will allow to use them in the formula (2) for the risk level calculation in the OSCAD tool. For the validation of OSCAD usage in the flood domain, more precise values of probability (likelihood) were defined (as presented in Fig. 2), but these values can be easily changed and adjusted to the requirements.

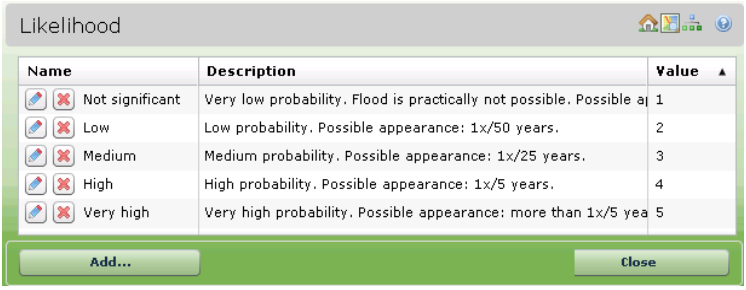


Fig. 2 Example of likelihood assessment scale definition in OSCAD dictionary

In the case of OSCAD usage in the ValueSec frames, it was necessary to analyze the possibility to adapt and configure OSCAD for the needs of the "Flood use case". For this purpose the documents describing the use case had to be analyzed first. The data were prepared and provided by the project partners (Fraunhofer Institute and The Centre for European Security Studies) with the participation of

the employees of the Saxony-Anhalt Ministry of the Interior and Sports and Saxony-Anhalt Ministry of Agriculture and the Environment.

The initial validation of OSCAD for the analyzed case was conducted based on the data about the flood on the Elbe and Mulde rivers in the Magdeburg area in Saxony-Anhalt. Information about the impact and size of previous floods came from reports about floods in this area. Based on these materials main threats were identified first, along with security measures which were assessed during RRA within the "Flood use case". Sample values of other parameters required for the analysis in OSCAD were defined in the dictionaries (e.g. assessment scales used to assess the impact, frequency of occurrence, security measures efficiency, etc.).

The dictionaries allow to freely assign values in a defined scale. These can be successive natural numbers (1, 2, 3, …) or the values can change with different steps (e.g. 1, 5, 25, 100, …). This way it is possible, to some extent, to control the range of possible output values of the risk level, yet the general shape of the diagram remains the same for the adopted formula (2), as presented in Fig. 3.

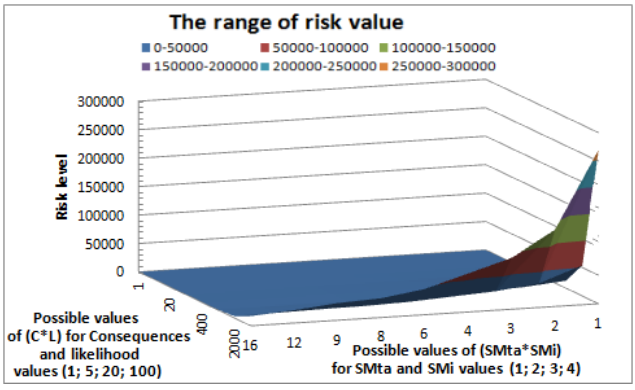


Fig. 3 The example of risk values range depending on the parameters value

After entering the data to the OSCAD data base there was a trial to conduct a risk analysis. The EU directive on floods lists possible aspects (operations) of the flood management process. These operations (processes) are: prevention, protection and preparedness (Fig. 4). That is why there was an attempt to conduct an analysis of processes comprising these operations.

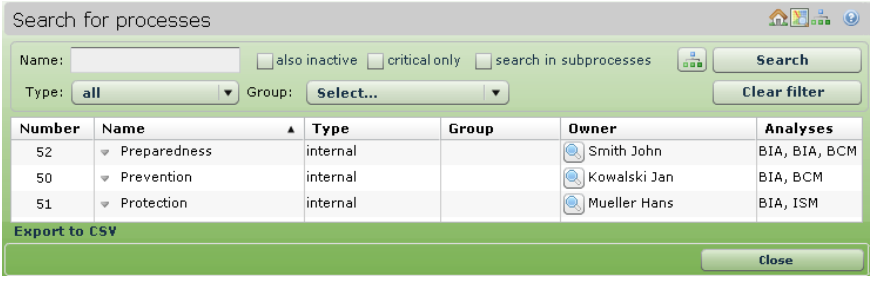


Fig. 4 Example of possible processes for the flood domain registered in OSCAD

The Business Impact Analysis (BIA) allows to assess the criticality (weight) of each performed process, which, as the *Pc* parameter, is put into the risk formula (2) and affects its value. The more important the process is, the higher is the risk level related to the materialization of the threat in this process. The assessment is conducted based on the definable business loss matrix which defines the considered loss categories, the number of loss levels and the description of each level (Fig. 5). In the course of the BIA analysis it is necessary to assign to each assessed process the estimated level of losses which can occur after losing the security attribute.

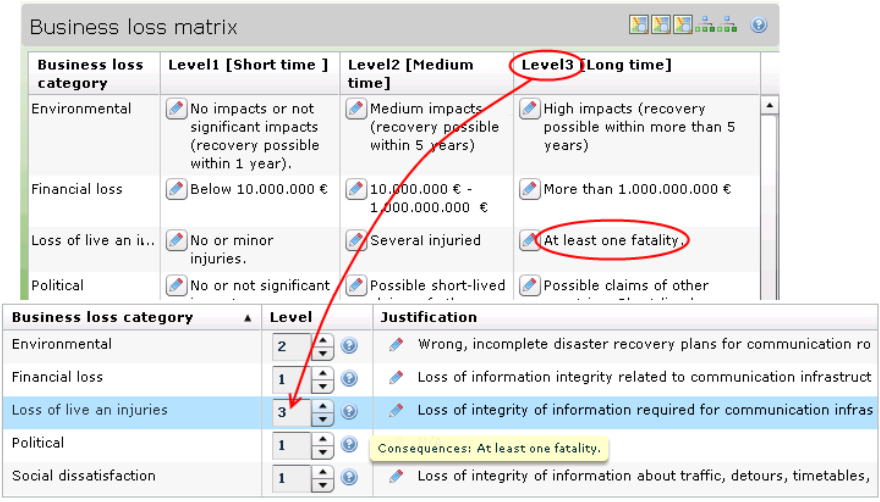


Fig. 5 Example of business loss matrix used for the assessment of possible losses in BIA

If there are no processes distinguished for the "Flood use case", and the flood risk management is treated as one process, it is possible to have value '1' for the assessment of the impact of security attributes loss. Then the value of the process will be equal to '1' and it will be possible to proceed to the detailed analysis.

Within this analysis threats and vulnerabilities (i.e. weak points which can contribute to the threat materialization) are identified (Fig. 6). The existing security measures, which reduce vulnerabilities and counter threats, are also identified.

Fig. 6 The example of threats and vulnerabilities selected in the flood use case

The main threat considered in the case of a flood are the following:

- Heavy rainfall results in the rising water level and flooding of the given area with direct impact (damaged communication and telecommunications infrastructure, loss of lives and health, damages to the natural environment, …) and indirect impact (social dissatisfaction, disturbances in the functioning of enterprises and public institutions, epidemic threat, …).
- Breakdown of the early warning system.
- Damage of a dam or dikes (failure to detect a high water level, inappropriate maintenance of flood infrastructure).

OSCAD enables to enter the list of threats, vulnerabilities and security measures to the data base and then to make connections between these values. Threats can have typical vulnerabilities assigned, and the vulnerabilities – typical security measures that reduce them (Fig. 7). The connections make it easier to search for these elements during the risk analysis process.

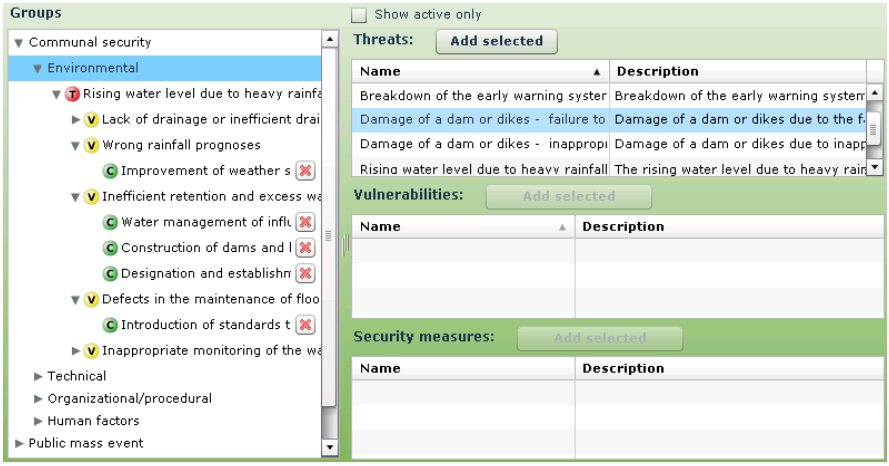


Fig. 7 View of dictionary for Threats-Vulnerabilities-Controls linkage

Next, for these parameters there is an assessment carried out to determine potential impacts of the threat materialization and probability of the event occurrence. Additionally, the functioning security measures are assessed.

Based on the assessments the current risk level is calculated (Fig. 8) which will be a point of reference to the risk values estimated after the security measure implementation. This way it is possible to determine the risk reduction level achieved for security measures considered in a given decision making process. With the "Flood use case" the following security measures are considered:

- 'Non-measure', which means leaving the current situation as it.
- Building and/or extension of dikes in the particular area.
- Introduction of standardized crisis management support software for communal crisis management task forces.

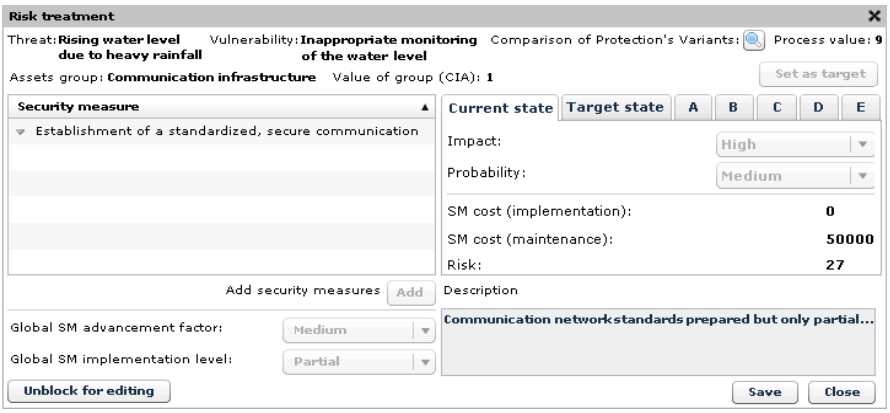


Fig. 8 Risk treatment window – assessment of current state of the risk level

It is possible to compare a few security measures (or sets of measures) in the OSCAD system thanks to the function which allows to assess several variants (up to five, described on tabs marked from A to E – as presented in Fig. 9). For each variant it is necessary to conduct a process of impacts estimation, event probability occurrence and security measures parameters assessment. On this basis, just as for the current risk, the estimated risk level will be calculated.

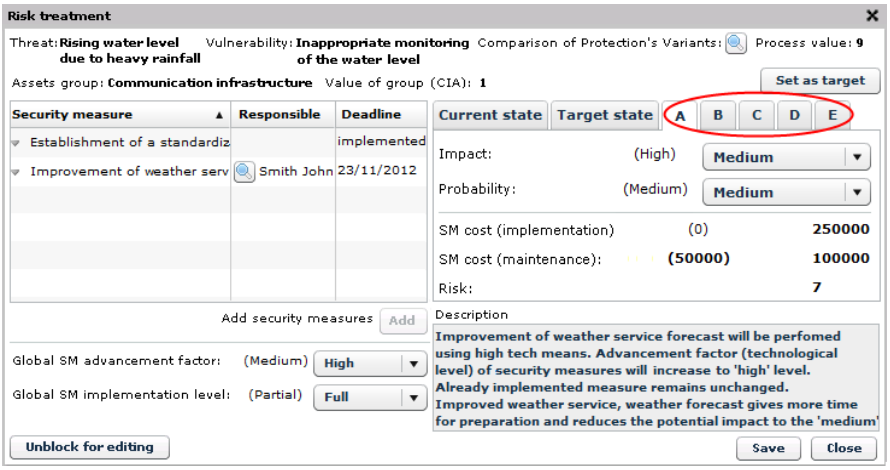


Fig. 9 Assessment of risk parameters for security measures ‘A’ variant

As it can be seen in Fig. 8, the current risk level for the analyzed case was equal to 27. While for the variant A (Fig. 9) the risk value was calculated at the level 6,75. Results for three example variants are presented in the Table 1. While comparing the analyzed variants it can be seen that the best risk reduction can be achieved with variant C, but the cost is very high. A similar risk reduction level can be achieved by implementing the A variant with moderate costs.

Table 1 Comparison of risk assessment parameters for analyzed variants

Risk assessment iteration	Risk level	Overall cost
Current risk	27	50000
A variant	6,75	350000
B variant	13,5	250000
C variant	6	1580000

As it was assumed for the ValueSec project tool architecture, the result of this stage of analysis (RRA pillar) should provide simple, clear information about the risk reduction level for each considered security measures set to support the decision maker in selecting the most appropriate variant of measures. The OSCAD tool gives at the end of the detailed risk analysis several different data sets related to costs, the risk level, and the level of impacts and probability estimated after the security measures implementation. Using the data gathered in the OSCAD tool, information can be processed manually and presented, for example, in the form of tables (as presented in Table 1). But the tool offers also a quick view window with predefined charts (Fig. 10) for presenting values of each pair threat-vulnerability, existing security measures and variants of planned security measures.

These charts present information related to the level of reduction of each risk parameter, such as expected risk level for each variant, or estimated value of such parameters (as impact, likelihood, levels of security measures parameters).

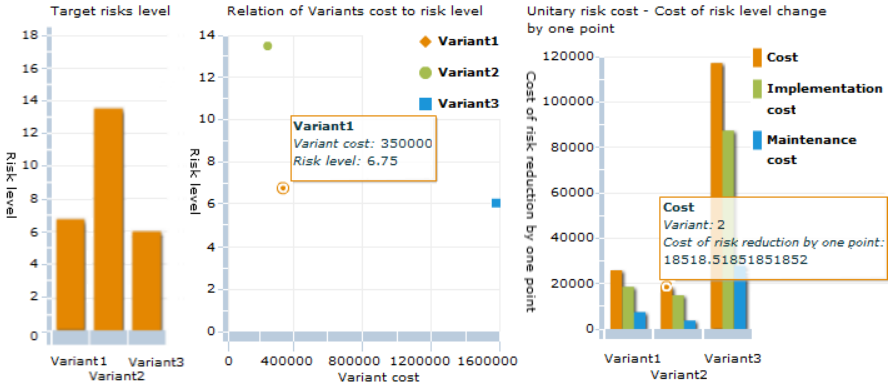


Fig. 10 The example of charts presenting results for each variant

By comparing these values with those calculated for the current situation, one can get information about the level of reduction in the case of a particular variant implementation. Other charts present information from the financial point of view. This is information about the risk level that can be achieved by the implementation of each set of measures with relation to their cost, and information about unitary cost of risk which means the cost of changing the risk by one-point.

The latter value (unitary cost of risk) is calculated in OSCAD with the use of the following formula (based on the solution proposed in [18], [20]):

$$UC = \frac{C_{i+1}}{RV_i - RV_{i+1}}, \tag{3}$$

where UC is the Unitary cost of risk, C_{i+1} means the cost of new security measures, RV_i means Risk Value for the i-th analysis and RV_{i+1} is the estimated value of risk after the new security measures implementation.

The information presented on the charts, received from the OSCAD database, can be next sent to the main ValueSec tool as the result of the RRA pillar, where together with Cost-Benefit Analysis and Qualitative Risk Assessment results will be presented to the decision makers as the result of the decision problem analysis.

4 Conclusions

The results of the first attempts to use the OSCAD software for the risk reduction analysis within the decision-support process were presented to the project partners. They seem to confirm the assumption that the general method adopted in the software, taking into account extra parameters (such as applied security measures and their efficiency or the implementation level), can be applied also in the selection of anti-flood security measures.

The results of the analysis conducted in the RRA pillar can be then considered during the analysis in the successive pillars (CBA and QCA). This way, during the whole cycle of the analysis and support of the decision about the security measures selection there will be certain aspects considered, such as the risk reduction level of particular variants, economic efficiency analyzed in CBA. Possible impacts of the security measures on the factors which are difficult to measure, such as social, political, etc. are also taken into consideration during the QRA.

One of the crucial issues to solve in the ValueSec project is the scope and form of input data and results, as well as the way of exchanging these data between particular tools selected for the risk assessment (Riger, Lancelot, OSCAD, RAS) within this RRA pillar. Data exchange between pillar is also one of the implementation challenges in this project.

One of the tested methods of data exchange was the usage of a broker which communicates with the database of each tool. The broker gets information about threats, vulnerabilities, security measures, or any other required element saved in the database. The user sends queries in the SPARQL query language to the broker which returns results prepared based on the mapping file. This file serves the mapping between objects in databases on classes and properties defined in the ontology (more information about the ontology usage in the security domain can be found in [22], [23], [24]). Such approach requires information about the part of the database structure which stores data for the exchange. While there are some issues regarding the access to the database and presenting the database structure, this approach was tested only with the different instances of OSCAD systems. Access to the database of each tool (Fig. 11) was simulated by connection with different instances of the OSCAD tool.

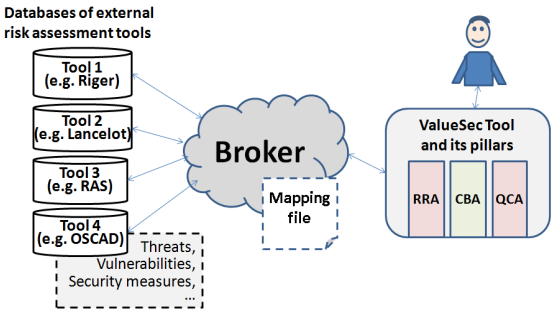


Fig. 11 A simple diagram of tested method of data exchange between tools in RRA

Due to the fact that EMAG is the owner of OSCAD and has access to source codes, it is possible to make some changes in the software for the needs of ValueSec if it is necessary to make extra lists or give access to extra output data. The software itself, in the majority of its modules, offers a function of data export to the CVS format which can be then read by any calculation sheet (e.g. Microsoft Excel) while the exported data can be used to work out a list or diagram.

Some parts of the OSCAD system were also verified against the Common Criteria standard. As a result the security architecture of the tool was assessed according to possible vulnerabilities which could be exploited by threat agents. Consequently, the tool becomes more reliable to users. The OSCAD system can be placed at remote sites (as it was mentioned in the description of the data exchange test). In that case the assurance of the system can be strengthen by applying another Common Criteria approach called Site Certification, as it was described in [25].

Currently, when the article is being written, further tests are underway. It is planned to have meetings of the project partners and representatives of Saxony-Anhalt ministries to specify input data and configure the software for the case of flood risk assessment of the Elbe river in Saxony-Anhalt.

References

[1] ISO 31000:2009 – Risk management - Principles and guidelines
[2] ISO/IEC 31010:2009 – Risk management - Risk assessment techniques
[3] ISO/IEC 27005:2008 – Information technology - Security techniques - Information security risk management
[4] BS OHSAS 18001:2007 – British Standard for occupational health and safety management systems – Requirements
[5] Directive 2007/60/EC of the European Parliament and of the Council of 23 October 2007 on the assessment and management of flood risks
[6] http://www.sepa.org.uk
[7] Flood Risk Management Planning in Scotland: Arrangements for 2012 - 2016, http://www.sepa.org.uk/flooding/flood_risk_management/national_flood_risk_assessment.aspx

[8] isok.imgw.pl (accessed January 18, 2013)
[9] Raport z wykonania wstępnej oceny ryzyka powodziowego. IMGW PIB. W konsultacji z Krajowym Zarządem Gospodarki Wodnej (2011), http://www.kzgw.gov.pl/files/file/Materialy_i_Informacje/WORP/Raport.pdf (accessed January 18, 2013)
[10] Białas, A.: Risk assessment aspects in mastering the value function of security measures. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 25–39. Springer, Heidelberg (2013)
[11] ValueSec Project, http://www.valuesec.eu (accessed January 10, 2012)
[12] Institute EMAG, Reports of a specific-targeted project "Computer-supported business continuity management system – OSCAD" (2010-2012)
[13] BS 25999-2:2007 Business Continuity Management – Specification for Business Continuity Management
[14] ISO/IEC 27001:2005 – Information technology – Security techniques – Information security management systems – Requirements
[15] Bagiński, J., Rostański, M.: The modeling of Business Impact Analysis for the loss of integrity, confidentiality and availability in business processes and data. Theoretical and Applied Informatics 23(1), 73–82 (2011) ISSN 1896-5334
[16] Baginski, J., Białas, A.: Validation of the software supporting information security and business continuity management processes. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) Complex Systems and Dependability. AISC, vol. 170, pp. 1–17. Springer, Heidelberg (2012)
[17] Bialas, A.: Computer support in business continuity and information security management. In: Kapczyński, A., Tkacz, E., Rostanski, M. (eds.) Internet - Technical Developments and Applications 2. AISC, vol. 118, pp. 155–169. Springer, Heidelberg (2012)
[18] Białas, A.: Bezpieczeństwo informacji i usług w nowoczesnej instytucji i firmie. WNT Publishing House, Warsaw (2006)
[19] Białas, A.: Development of an Integrated, Risk-Based Platform for Information and E-Services Security. In: Górski, J. (ed.) SAFECOMP 2006. LNCS, vol. 4166, pp. 316–329. Springer, Heidelberg (2006)
[20] Białas, A., Lisek, K.: Integrated, business-oriented, two stage risk analysis. Journal of Information Assurance and Security 2(3) (September 2007) ISSN 1554-10
[21] Białas, A., Cała, D., Napierała, J.: Wspomaganie zarządzania ciągłością działania zakładu górniczego za pomocą system OSCAD. Mechanizacja i Automatyzacja Górnictwa, Czasopismo Naukowo – Techniczne 7(497), 11–25 (2012)
[22] Białas, A.: Security Trade-off – Ontological Approach. In: Akbar Hussain, D.M. (ed.) Advances in Computer Science and IT, pp. 39–64. In-Tech, Vienna-Austria (2009) ISBN 978-953-7619-51-0, http://sciyo.com/articles/show/title/security-trade-off-ontological-approach?PHPSESSID=kkl5c72nt1g3qc4t98de5shhc2
[23] Bialas, A.: Ontological Approach to the Business Continuity Management System Development. In: Arabnia, H., Daimi, K., Grimaila, M.R., Markowsky, G. (eds.) Proceedings of the 2010 International Conference on Security and Management, The World Congress In Applied Computing – SAM 2010, Las Vegas, USA, July 12-15, vol. II, pp. 386–392. CSREA Press (2010) ISBN: 1-60132-159-7, 1-60132-162-7 (1-60132-163-5)

[24] Bialas, A.: Common Criteria Related Security Design Patterns for Intelligent Sensors—Knowledge Engineering-Based Implementation. Sensors 11, 8085–8114 (2011), http://www.mdpi.com/1424-8220/11/8/8085/

[25] Rogowski, D., Nowak, P.: Pattern based support for site certification. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) Complex Systems and Dependability. AISC, vol. 170, pp. 179–193. Springer, Heidelberg (2012)

Risk Assessment Aspects in Mastering the Value Function of Security Measures

Andrzej Białas

Institute of Innovative Technologies EMAG,
40-189 Katowice, Leopolda 31, Poland
a.bialas@emag.pl

Abstract. The chapter presents the risk management approach applied in the EC FP7 ValueSec project. The security measures selection process is based on three pillars: Risk Reduction Assessment (RRA), Cost-Benefit-Analysis (CBA) and Qualitative Criteria Assessment (QCA). The ValueSec tool set, which is elaborated in the project, should be equipped with components corresponding to these pillars. The chapter overviews the researches of the project focused on the decision model elaboration and selection of existing method to be implemented, or existing tools to be integrated in the ValueSec framework. Risk management is a broad issue, especially in five of the project assumed contexts. For this reason more specialized components are allowed for the RRA pillar. Currently the project passes to the implementation and use case experimentation phase. The chapter shows the general architecture, currently implemented and the RRA component example.

1 Introduction

The chapter concerns the selected aspects of risk management with respect to the ValueSec project [1] financed by the EC 7th Framework Programme.

The objective of the project is to improve the decision process related to security measures selection so that the proposed measures could take into consideration the stakeholders' needs and interests to the highest possible extent. Along with the developed decision support methodology, a software tool is prepared for policy level stakeholders in the field of security. The project results should allow to support policy decision makers in making better informed decisions.

The scientific problem to be solved in the interdisciplinary ValueSec project can be defined in the following way: developing a computer-aided decision support methodology in security concerning the selection of security measures, so that the measures not only properly affected the risk but also were cost-effective and took into account social, political and legal restrictions which are related to the decision making process. Taking into account these restrictions, here called qualitative factors (criteria), is the basic added value of the project.

W. Zamojski et al. (Eds.): *New Results in Dependability & Comput. Syst.*, AISC 224, pp. 25–39.
DOI: 10.1007/978-3-319-00945-2_3

The project passed its half-way point. The key analytical works were completed, the assumptions for the decision support methodology were made [2], [3], [4], [5], methods or tools for implementation were selected [6], [7], [8], [9], the functional design and architecture of the software were developed, and the validation process of the developed solution is being prepared.

The chapter will focus on one of the project pillars which is risk assessment, therefore in the next section the ValueSec approach to the security-related decision support will be presented.

Section 3 will feature the analysis of the existing methods and tools with a view to select some of them for implementation in the ValueSec project. Sections 4 and 5 will feature the general architecture and an example of a risk assessment tool implementation. The final summary will present operations that are planned in the course of the project and related to risk assessment in the project contexts.

2 ValueSec Approach to Security Related Decisions

The decision making process related to security measures is important to many organizations, projects, social groups, and individuals as it affects security, business efficiency and social acceptance for these security measures. At the same time it is an extremely complex process since it has to take into account a number of factors of different, complicated and still unexplored nature.

Generally speaking, security measures are applied to reduce risk. Among many security measures that are able to reduce risk in a particular case, a part may be economically inefficient or impossible to apply due to objective restrictions. The decision about the measure selection is a multi-dimensional issue.

The objective of the ValueSec project is to master the value function of security measures [10]. This value function is researched, all its arguments (factors) and their multidimensional impact to the security problem are identified.

The decision about selecting a security measure in a given situation should be worked out on the basis of the following three factor groups, each including a number of detailed issues:

- the security measure should be able to affect the risk volume sufficiently (based on the risk appetite) in order to provide security on a suitable level,
- the security should be cost-effective in order not to reduce the efficiency of operations and not to incur unnecessary costs,
- the security should take into account a number of restrictions: social, psychological, political, legal, ethical, economical, technical, environmental, etc. – in order to use the security measures in practice; in the project terminology these factors are called qualitative criteria.

Each group of issues in the project is called a pillar, thus one can distinguish three pillars in ValueSec:

- Risk Reduction Assessment (RRA) pillar,
- Cost-Benefit-Analysis (CBA) pillar,
- Qualitative Criteria Assessment (QCA) pillar.

The first pillar concerns a vast domain of risk assessment, including risk analysis and risk evaluation. The risk analysis is conducted in many domains, including widely understood security. Different strategies, methods and tools are used, models of different degrees of detail are applied. They provide a set of factors which are taken into account in the decision making process concerning the security measures selection [11]. This pillar is responsible for calculating risk reduction resulting from the application of the given security measure. Due to the existence of many theories, methods and tools for conducting and supporting risk assessment, an exhaustive two-stage analysis of these TMTs had to be conducted to select those that could be implemented or integrated in the ValueSec framework. This issue will be discussed in detail in section 3.

Out of the security measures that affect properly the risk level it is necessary to choose cost-effective variants which comply with different restrictions characteristic of a concrete situation. The monetary approach is used. The key issue is an economic analysis about the cost-efficiency of the applied measures with respect to their costs and benefits. The applied economic models provide the second set of factors considered in the decision making process. This pillar is responsible for calculating, in monetary units, negative and positive effects of applying a certain measure [12].

The objective of the CBA analysis is to assess from the economic point of view the impact of security-related decisions. This analysis encompasses the following categories:

- costs of provision and investment,
- direct and indirect maintenance costs,
- immaterial costs,
- direct and indirect benefits.

Each of these categories has its subcategories. For example, the category of direct and indirect costs has the following subcategories:

- operational costs, comprising, e.g. costs of equipment and its modifications, costs of personnel training,
- maintenance costs, comprising the costs of planned and unexpected reviews and repairs of equipment, costs of spare parts, costs of IT services and maintenance,
- costs of utilization, comprising, among others, costs of a system closure, its disassembly, costs of recycling.

The category of direct and indirect benefits, in turn, includes the following subcategories:

- direct economic benefits, such as the estimated sales volume or incomes from the sale of patents and licenses,
- benefits resulting from the reduction of risk and the degree of vulnerabilities to threats,
- social, legal and political benefits, e.g. better image of the organization, higher consumption, acquiring new clients, better contacts with the organization's business and legal environment.

The third pillar comprises the analysis of restrictions with the use of varied factors which are difficult to determine [8]. This is a new research issue. About 120 issues in several groups (social, political, legal, etc.) were identified and the character of their relations to security measures was determined. This way the third set of factors was created – qualitative criteria, taken into consideration in the decision making process. This pillar is responsible for the evaluation of other important security-related factors.

The Qualitative Criteria Analysis (QCA) is meant to assess those criteria of the decision making process which cannot be assessed by means of quantitative methods. In this case the assessment process has to take into account a number of immaterial parameters of security-related decision making. These parameters can be assigned to the following groups:

- social parameters (social group level),
- individual parameters (individual level),
- legal regulations,
- social laws and ethics,
- politics,
- economy,
- technologies and science,
- environment.

For example, in the economic group of parameters the following issues are included:

- Does the applied security measure affect the consumption behaviour of the society?
- Does it affect the general investment climate (of the country, region, city)?
- Do the applied security measures affect production processes?
- Can the applied security measures cause economic losses for an organization, city, region?
- Are the costs of applied security measures proportional to the achieved effects?
- Can the applied security measures increase or reduce the market value of real estate (in a city, region)?

Though the above issues have an economic background, they cannot be expressed in monetary units as it was the case with categories used in the CBA analysis.

In the ValueSec project the above three groups of issues are considered in five application domains, called contexts [13]: public mass event, public mass transportation, air transportation/airport security, communal security planning, cyber threat.

In the course of the ValueSec project fulfillment the ValueSec framework is developed. It integrates a set of tools corresponding to the three pillars. The tools are to provide a set of analytical data to be used in the security-related decision-making process. Security-related decisions are made based on the conceptual decision model [3], prepared for ValueSec, presented in Fig. 1.

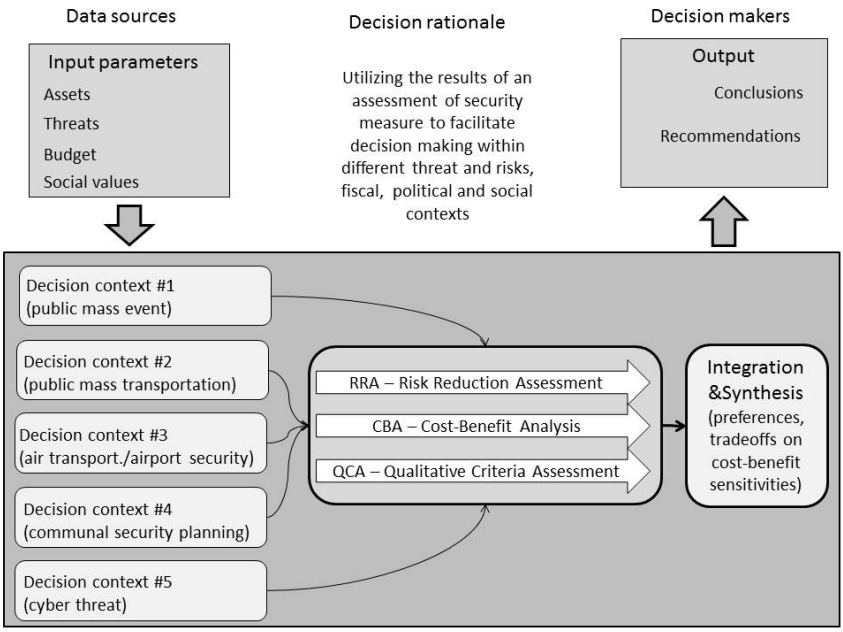


Fig. 1 Conceptual decision model

For the given decision context the decisions have three phases:

- acquisition of input data and definition of data sources that will be the base for the analysis of variants; the data encompass assets, threats, vulnerabilities, budget and time restrictions, soft factors;
- evaluation of different measures – as decisions variants; tools for three pillars are used (RRA, CBA and QCA); different experiments are provided with variants to produce information for decision makers,
- integration and synthesis; information obtained from the previous step are analyzed, integrated and concluded by decision makers, to produce transparent decisions (considering preferences, restrictions and trade-offs).

3 Components of the ValueSec Framework

The problem was to identify methods which can be implemented as the ValueSec framework components, or tools which can be integrated in this framework – all of them should meet the project needs and restrictions, and indirectly should satisfy the stakeholders' expectations. Before defining the ValueSec framework architecture, 2-stage researches on the current state of technology were performed to identify theories, methods or tools (TMTs) which can be implemented in the ValueSec framework.

3.1 General Review of the Theories, Methods or Tools

In the first stage the following categories of TMTs were considered:

- Information handling,
- Risk analysis, risk assessment,
- Cost structuring/analysis and evaluation and analysis of societal impacts,
- Structuring and analysis of values,
- Analysis of decision alternatives,
- Other supporting methods, theories and tools,

The following individual assessment criteria (characteristics) were defined for the TMTs assessment framework: name, theory/method/tool, category, summary, objective, functionalities, qualitative/quantitative attributes of interest, weighting of attributes, phase of decision making, type of decision support, type of damage, inclusion of incident probability and uncertainty, assessment of effects, scientific experience, decision-makers' experience, age, complexity, required resources, required competencies, maturity, timeframe, data-related questions, questions about application and implementations, references [6].

The consortium members evaluated 29 TMTs [7]. They were organized in a matrix structure to match the ValueSec requirements:

- expected functionality (3 above mentioned pillars),
- decision-making context (5 above mentioned contexts).

The following 10 methods and tools were transferred to the next stage for further assessment:

- Quantitative Risk Assessment (QRA),
- Risk Measurement /Risk Analysis (RM/RA),
- Expert Choice (EC),
- Lancelot,
- OSCAD,
- Riger,
- Bayesian Network Analysis (BNA),
- Strategic Approaches (SA),

- TableTop Exercise (TTE),
- Magerit.

The theories were excluded due to the restricted project resources for their implementation. For the CBA and QCA pillars no proper solutions were found. For this reason it was decided that they will be implemented by the consortium partners. All methods and tools transferred to the next stage are related to the risk assessment process (first RRA pillar).

3.2 Usability Assessment Criteria and Usability Analysis

The second stage of the methods and tools assessment was focused on the usability and feasibility aspects.

Recommending the right methodology to implement it in the ValueSec tool set, or recommending the given tool to integrate it (as a component) with this tool set, requires the identification of their most favorable features and the elaboration of the usability criteria to assess them. Different types of issues have been considered during the usability assessment criteria elaboration:

- Risk-related, security economics and soft factors (further quality criteria of decisions) methodology; they should cover the requirements and expectations of stakeholders;
- Related to the capability of decision support for policy makers, the urgency of the users' needs, decision context;
- Dealing with the expected efficient use of the required ValueSec project resources and the feasibility within the given time frame and development risks;
- Related to the implementation requirements, methodical innovations, project challenges issues, and constraints of the selected methods.

The usability assessment criteria, elaborated with the use of Microsoft Excel, were specified in [9]. These criteria have a three-level hierarchical structure (Fig. 2). On each level weights were assigned. The assessment results are presented in a tabular and graphical form. There are 8 groups of criteria dealing with:

- The compliance of the considered method/tool with the ValueSec assumptions, objectives – general parameters related to the method/tool compliance, adequacy, and usability with respect to the area determined within the project;
- The parameters regarding the possibility of situation description (framing conditions specification, problem identification);
- The data characteristics parameters concerning the method of description of input/output data, type, source and other data related issues;
- The functional parameters comprising the desirable functions, possible analyses which are performed and supported by the method/tool;
- The recommendations, reports, final decision support related issues, the types of provided reports, the way of results presentation for decision-makers;

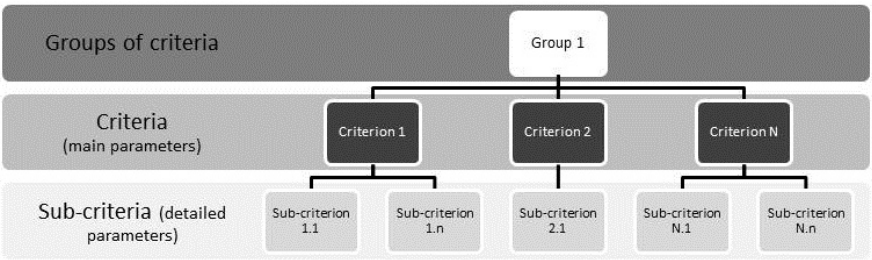


Fig. 2 Hierarchical structure of the usability criteria

- The development/implementation risk parameters concerning the level of risk implied by the use of the given method/tool, its implementation, like source code availability, time, budget and resources required for implementation; such parameters like method/tool age, maturity, general experience, which were identified within WP3, were also taken into consideration; they may have significant influence on the development risk;
- The general challenges specified for the ValueSec project concerning such topics as: scientific challenge, implementation challenge, testing challenge;
- The tool characteristics parameters characterizing mostly the technical aspects of the tool.

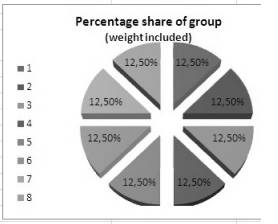
Each group contains one or more main parameters (criteria). Additionally, for some main parameters (criteria) there were detailed parameters (sub-criteria) defined. An example of the three-layer structure:

- The "Tool characteristic (technical) attributes" group incorporates;
- The "Software processing (How are the data processed?)" criterion, which includes;
- Several sub-criteria (detailed parameters), e.g.: "online", "real time", "batch", which can be assessed by the given method/tool evaluator.

On each hierarchy level weights can be assigned, allowing to express the importance of the given group in comparison with others, the importance of the given criteria in comparison with others and the importance of the given sub-criteria with respect to others. The initial weight value for each element was set to "1" and it is the lowest possible value. The highest value was not specified a priori, but during the adjustment of criteria weights the maximum value was determined as "3". The use of the weights will enable to distinguish parameters that are essential for the project and to give better score to particularly desirable characteristics of the methods and tools.

Fig. 3 presents weights for groups of criteria, initially set to "1". It means all groups have the same importance. For the group "Compliance …" its four criteria with the same weights are shown.

Group No.		Group names	Group weight	Percentage share (weight included)
I		Compliance of MT with the VS assumptions, objectives	1	12,50%
II		Situation description	1	12,50%
III		Data characteristic parameters	1	12,50%
IV		Functional parameters (available functions, analyses, attributes de	1	12,50%
V		Recommendations, reports, final decision support (parameters, a	1	12,50%
VI		Development/Implementation risk parameters	1	12,50%
VII		General challenges specified in the project DoW	1	12,50%
VIII		Tool characteristic attributes	1	12,50%
			8	100,00%



	Criterion No.		Criterion weight	Percentage share (weight included)
I. Compliance				
	1	Context	1	25,00%
	2	ValueSec project use case	1	25,00%
	3	Potential end-users	1	25,00%
	4	Decision-making process	1	25,00%
II. Situation description				

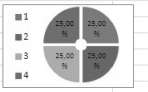


Fig. 3 Weights for groups of criteria and for criteria of the group "Compliance"

Fig. 4 presents a small part of the usability criteria with some data related to one of the evaluated methods/tools called "OSCAD". Please note: the group "Compliance of considered method/tool with the ValueSec assumptions", the criterion "context", and its six sub-criteria: "1a" through "1f".

					Name		The name of the method/tool (owner)	OSCAD (EMAG)		
					Acronym		The acronym of the method/tool	OSCAD		
					Identifier from document WP3 T3.1/D3.1		Link do detailed description placed in WP3 T3.1/D3.1			
					Categorization of method/tool			Method/tool categorization		
	Source of the parameter	Type of the parameter (General/Tool/Method specific)	Main parameter	Weight factor for main parameter	Detailed parameter (if needed)	Weight factors of parameters	Set of possible values	Assessed value	Value with weight factor included	Description/justification of assessment
Compliance of MT with the VS assumptions, c						1	Total (for compliance parameters		#ADR!	
1a	D3.1_v3.0 - ch.3 p. 25c	G	Context To which of the ValueSec contexts does the method relate?	1	public mass event	1	1 - suitable 0 - not suitable	1	1	Oscad can be useful for the risk analysis (threat/vulnerability assessment) for all areas. It is possible to assess possible impacts of the loss of confidentiality/integrity/availability (C/I/A) of a data/process/operation. Matrix with categories of expected losses and description of each possible level of loss must be prepered first to quantify the assessment).
1b	D3.1_v3.0 - ch.3 p. 25c	G			public mass transportation	1	1 - suitable 0 - not suitable	1	1	
1c	D3.1_v3.0 - ch.3 p. 25c	G			air transportation/airport security	1	1 - suitable 0 - not suitable	1	1	
1d	D3.1_v3.0 - ch.3 p. 25c	G			communal security planning	1	1 - suitable 0 - not suitable	1	1	
1e	D3.1_v3.0 - ch.3 p. 25c	G			cyber threat	1	1 - suitable 0 - not suitable	1	1	
1f	D3.1_v3.0 - ch.3 p. 25c	G			other outside the ValueSec contexts - specify in description	1	1 - suitable 0 - not suitable	1	1	

Fig. 4 Hierarchical structure of the usability criteria

These issues express whether "OSCAD" is suitable or not for particular contexts of the ValueSec project.

All sub-criteria have global meaning, which is expressed by "G". Please note weights assigned on each of the three levels of the usability criteria. During the method/tool assessment two columns "Assessed value" and "Description/justification of assessment" are filled in with data. For each assessed method/tool about 200 such detailed items should be filled in by scores.

The assessment against the methods/tools feasibility and the rationale of further implementation were based on the elaborated usability criteria. The functions and properties offered by the particular method or tool were analyzed. Particular attention was paid to how the method or tool fulfills the needs and requirements of the ValueSec project.

The project partners made an overview of the evaluation process of the 10 preselected, above mentioned methods/tools with the use of usability criteria. The assessment process encompasses two steps:

- Reviewing the particular methods/tools with respect to the usability criteria and their groups (horizontally, by methods/tools) – example Fig. 5;
- Identifying strengths and weaknesses of the preselected methods/tools with respect to the usability criteria (vertically, each method/tool against criteria) – example Fig. 6.

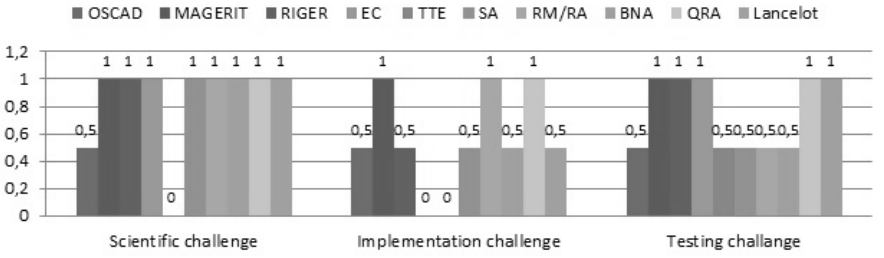


Fig. 5 General challenges specified in the project description of work – results

The results obtained during the assessment process are considered as "sensitive data". For this reason the deliverable D4.1 Part 2 has a document status RE (restricted), and below only partial results are presented as examples.

Fig. 5 presents the results related to the criteria group "General challenges specified in the project DoW (Description of Work)" as the example. For all methods/tools selected for the implementation of the ValueSec tool set, it will be a challenge to combine them together, to get as a result one comprehensive, common tool supporting the decision making process. It will be a challenge to integrate different risk analysis tools, and furthermore supplementing this tool set with other analyses (qualitative criteria as well as cost-benefit analysis).

Some of the considered tools have already been implemented (Lancelot, Riger, OSCAD) in an IT environment. They are used with respect to threats and security measures in the IT domain. Other selected and assessed tools come from other business sectors. Testing the risk analysis process in different environments (for different threats, vulnerabilities, and security measures) can be a challenge.

Fig. 6 presents another example – the results of the OSCAD assessment. They can be shown because OSCAD is developed by the author's organization.

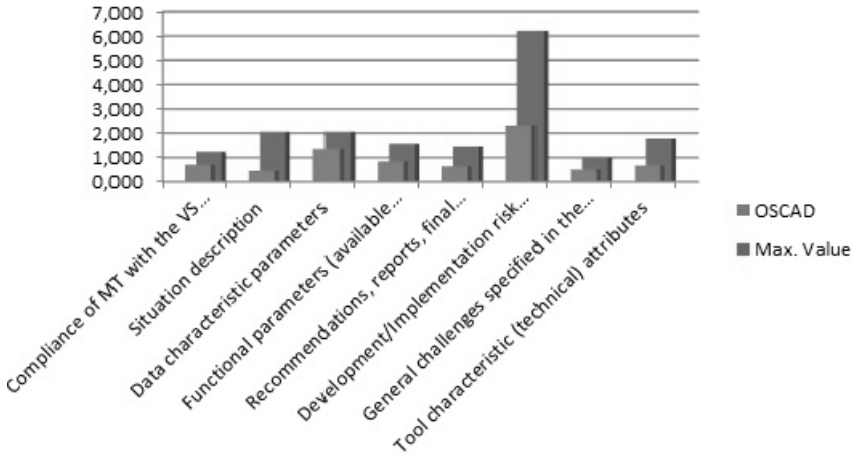


Fig. 6 OSCAD assessment – results for each group of criteria

The OSCAD tool "Data characteristic parameters" received a high score. The entered data are quantified which enables their computer-based analysis (risk level calculation). Similarly to Lancelot, "Situation description" covers, first of all, information about existing threats and vulnerabilities. There is no description of other aspects, such as time for conducting analyses, decision making dimension and decision making domain, assumed budget and time for security measures implementation, or expected benefits.

The tool has a very wide range of functions supporting information security and business continuity management. However, in terms of the ValueSec project requirements it enables, first of all, to conduct a risk analysis for the identified threats and vulnerabilities. Still, it does not offer other analyses, such as the "Cost-Benefit analysis". "Soft criteria" (now called" Qualitative criteria") are considered with respect to losses incurred due to the loss of confidentiality, integrity or availability of business processes. Their use in the implementation of the ValueSec tool set would require changes in the software. Hence the overall score of functional parameters is on a medium level.

Low score was given to the ability to generate reports. The tool does not offer too many possibilities of reports adaptation. The only way to generate reports other than the predefined ones is to export data from the selected views to the csv format and then process the data in a calculation sheet, e.g. in MS Excel.

Low score in the range of risk-implementation parameters results from the fact that OSCAD is a new tool. The general knowledge about the tool is not widely available. It is necessary to make some changes in the application and the owner's resources are limited. Nevertheless, it is possible to make changes and adaptations.

The source code belongs to a member of the ValueSec consortium, which is certainly a great advantage.

Technical parameters were scored on a medium level. Technical documentation and user's documentation need to be supplemented. There are limited possibilities as far as the configuration changes for the ValueSec tool set implementation, scalability and interoperability are concerned.

During the assessment process the most favorable methods/tools from the technical point of view and software implementation possibility point of view were identified. The results were presented in the form of tables and diagrams on which the assessment value of the given tool is presented against the maximal number of points that can be achieved during the assessment (due to sensitive information related to the method/tools comparison, the D4.1/part 2 deliverable has dissemination status "restricted").

As a result of the conducted assessment of methods and tools, there were four candidates selected for the final implementation of the RRA pillar:

- Riger, OSCAD – methods focused on assets,
- Lancelot, RAS (QRA) – methods focused on processes.

They will be assigned to concrete application domains, i.e. contexts.

4 ValueSec Framework Architecture

The ValueSec tool set encompasses components of three pillars (Fig. 7).

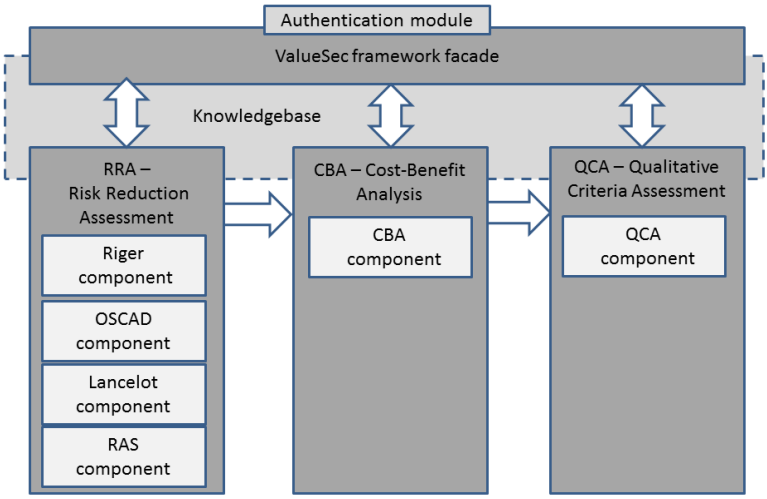


Fig. 7 ValueSec general architecture

For the CBA and QCA pillars the specific components have been developed by the consortium members. For the RRA pillar four different components will be integrated. They have been assigned to given contexts with respect to the context requirements.

Pillars are integrated by a common façade controlling work flow inside the ValueSec tool set. Additionally, some common components (not discussed) exist, such as the knowledge base, authentication module, etc.

5 OSCAD as the RRA Component Example

The OSCAD tool developed by the author's organization is assigned to the "flood protection use case" of the "communal security planning" decision context. The use case deals with flood prevention measures and will be modeled on the experiences of the German Bundesland Saxony-Anhalt (LSA) during the 2002 flood of the Elbe river.

The OSCAD software system manages business continuity according to the BS25999 standard and information security according to the ISO/IEC 27001 standard [14]. It has several modules, but for the ValueSec RRA pillar the risk management functionality has key importance. For the ValueSec project a dedicated software version was elaborated.

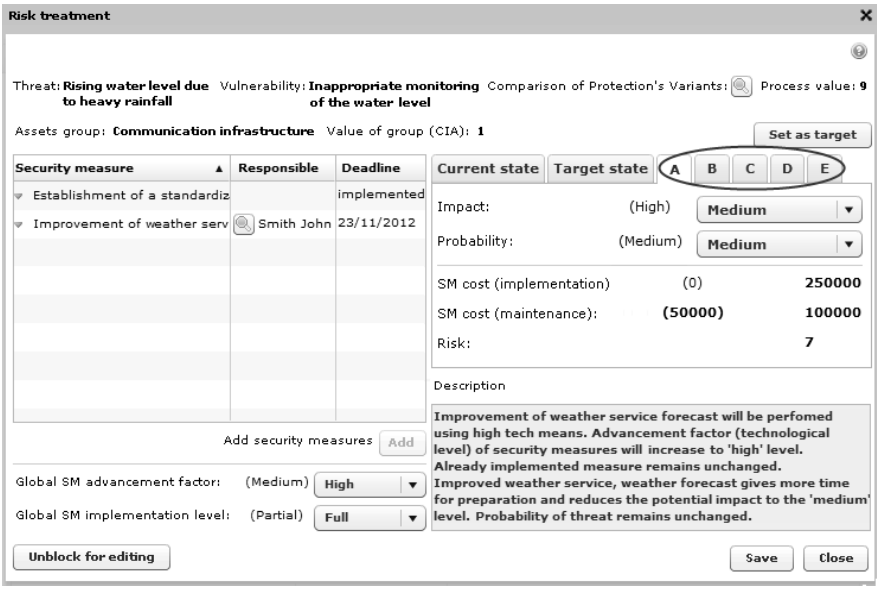


Fig. 8 OSCAD – analyzing variants of security measures (source: EMAG)

Risk management functions are responsible for:

- identification and specification of the business processes, taking into consideration assets related to the particular processes;
- conducting an analysis of harmful influence of losing a continuity attribute on business processes and harmful influence on losing integrity, availability and confidentiality of assets groups related to the given process; this type of analysis is called BIA (Business Impact Analysis) [15] and corresponds to HLRA (High Level Risk Analysis); processes with critical significance for the institution are identified;
- conducting LLRA (Low Level Risk Analysis) which allows to determine the risk value for each triple asset-threat-vulnerability; taking into account the existing security measures, their technical advancement and implementation level;
- selecting security measures which reduce the risk volume; security variants are defined (Fig. 8); the most beneficial variant is considered for implementation, i.e. the one which can reduce the risk and implementation costs the most.

The flood protection use case and the OSCAD tool facilities are discussed in a separate chapter [16].

6 Conclusions

The chapter presents a general approach to risk management applied in the EC FP7 ValueSec project. This approach is focused on mastering the value function of security measures. The ValueSec tool set, which has been elaborated in the course of the project, is based on three pillars: Risk Reduction Assessment (RRA), Cost-Benefit-Analysis (CBA) and Qualitative Criteria Assessment (QCA). The applied measures should be efficient in risk reduction, cost limitation, benefits increase and should be applicable with respect to different restrictions. The project passed its half-way point: analytical works were completed, methods and tools to be implemented as the components were selected, functional design and architecture were defined. Currently the ValueSec tool set is implemented and prepared for use cases experimentations in five decision contexts.

References

[1] ValueSec web page: `http://www.valuesec.eu` (accessed January 10, 2012)
[2] D2.1 Decision domains concepts and trends (2011), `http://www.valuesec.eu/content/d21-decision-domains-concepts-and-trends`
[3] D2.2 Data model and decision model (2011), `http://www.valuesec.eu/content/d22-data-model-and-decision-model`

[4] D2.3 Relational concept between security and politico-economic sphere (2011), http://www.valuesec.eu/content/d23-relational-concept-between-security-and-politico-economic-sphere
[5] D2.5 Report on workshop on user needs and requirements (2011), http://www.valuesec.eu/content/d25-report-workshop-user-needs-and-requirements
[6] D3.1 Framework for the assessment of methods and tools (2011), http://www.valuesec.eu/content/d31-framework-assessment-methods-and-tools
[7] D3.2 Catalogue of evaluated methodologies and tools available (2011), http://www.valuesec.eu/content/d32-catalogue-evaluated-methodologies-and-tools-available
[8] D3.3 Evaluation of methods and tools, and the required improvements (2012), http://www.valuesec.eu/content/d33-evaluation-methods-and-tools-and-required-improvements
[9] D4.1 Part 1 Usability assessment criteria and usability analysis (2012), http://www.valuesec.eu/content/d41-part-1-usability-assessment-criteria-and-usability-analysis
[10] Zuniga, E.B., Blobner, C.: ValueSec – Mastering the Value Function of Security Measures. In: Ender, J., Fiege, J. (eds.) 6th Future Security: Security Research Conference, Future Security, Berlin, September 5-7. Conference Proceedings, pp. 277–281 (2011)
[11] Bjorheim Abrahamsen, E., Aven, T., Pettersen, K., Rosqvist, T.: A framework for selection of strategy for management of security measures. In: PSAM 2011 & Esrel 2012 Int'l Conference Proceedings. Scandic Marina Congress Centre, Helsinki, Finland, June 25-29, pp. 18-Tu2-4. USB memory stick (2012)
[12] Räikkönen, M., Rosqvist, T., Poussa, L., Jähi, M.: A Framework for Integrating Economic Evaluation and Risk Assessment to Support Policymakers' Security-related Decisions. In: PSAM 2011 & Esrel 2012 Int'l Conference Proceedings. Scandic Marina Congress Centre, Helsinki, Finland, June 25-29, pp. 18-Tu3-2. USB memory stick (2012)
[13] Adar, E., Blobner, C., Hutter, R., Pettersen, K.: An extended Cost-Benefit Analysis for evaluating Decisions on Security Measures of Public Decision Makers. Forthcoming CRITIS 2012, 7th International Conference on Critical Information Infrastructures Security, Lillehammer, Norway, September 17-19 (2012)
[14] Białas, A.: Computer support in business continuity and information security management. In: Kapczyński, A., Tkacz, E., Rostanski, M. (eds.) Internet - Technical Developments and Applications 2. AISC, vol. 118, pp. 155–169. Springer, Heidelberg (2012)
[15] Bagiński, J., Rostański, M.: The modeling of Business Impact Analysis for the loss of integrity, confidentiality and availability in business processes and data. Theoretical and Applied Informatics 23(1), 73–82 (2011) ISSN 1896-5334
[16] Bagiński, J.: Software support of the risk reduction assessment in the valueSec project flood use case. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 11–24. Springer, Heidelberg (2013)

Reduction of Computational Cost in Mutation Testing by Sampling Mutants

Ilona Bluemke and Karol Kulesza

Institute of Computer Science,
Warsaw University of Technology, Nowowiejska 15/19,
00-665 Warsaw, Poland
I.Bluemke@ii.pw.edu.pl

Abstract. The objective of this chapter is to explore the reduction of computational costs of mutation testing by randomly sampling mutants. Several experiments were conducted in the Eclipse environment using *MuClipse* and *CodePro* plugins and especially designed and implemented tools: *Mutants Remover* and *Console Output Analyser.* Six types of mutant' subsets were generated and examined. Mutation score and the source code coverage were used to evaluate the effectiveness of mutation testing with subsets of mutants. The ability to detect errors introduced "on purpose" in the source code was also examined.

1 Introduction

Mutation testing is a fault based software testing technique that was introduced more than forty years ago. The general idea is that the faults used in mutation testing represent the mistakes made by a programmer so they are deliberately introduced into the program to create a set of faulty programs called *mutants*. Each mutant program is obtained by applying a mutant operator to a location in the original program. To assess the quality of a given set of tests these mutants are executed against the set of input data to see, if the inserted faults can be detected. A very good survey of mutation techniques was written in 1996 by Jia and Harman [1], they also created a repository [2] containing many interesting papers on mutation testing. Recently Bashir and Nadeem published a survey on object mutation [3].

Mutation testing is effective at measuring the adequacy of a test suite, but it can be computationally expensive to apply all the test cases to each mutant. Previous research has investigated the effect of reducing the number of mutants by selecting certain operators, sampling mutants at random, or combining them to form new higher-order mutants.

The objective of this chapter is to examine what is the impact of randomly sampling mutants for Java programs, on the mutation score, the code coverage and the ability to detect real errors. The main ideas of mutation testing and reducing the number of mutants are briefly described in section 2 while related work is presented in section 3. The results of experiments are presented in section 4 and some conclusions are given in section 5.

W. Zamojski et al. (Eds.): *New Results in Dependability & Comput. Syst.*, AISC 224, pp. 41–51.
DOI: 10.1007/978-3-319-00945-2_4

2 Mutation Testing

The mutation testing is a fault based software testing technique that was introduced in 1971 by Richard Lipton (according to [4]). Surveys on mutation techniques were written e.g. by Jia and Harman [1], Bashir and Nadeem [3]. Many papers on mutation testing can be found in a repository [2].

The general idea of mutation testing is that the faults represent mistakes made by a programmer, so they are deliberately introduced into the program to create a set of faulty programs called *mutants*. Each mutant program is obtained by applying a mutant operator to a location in the original program. Typical mutation operators include replacing one operator e. g. '+' by another e.g. '-' or replacing one variable by another. To assess the quality of a given set of tests the mutants are executed on a set of input data to see, if the inserted faults can be detected. If the test is able to detect the change (i.e. one of the tests fails), then the mutant is said to be *killed*. The input data for test should cause different program states for the mutant and the original program.

A variety of mutation operators were explored by researchers. Some examples of mutation operators for imperative languages: statement deletion, replacement of each Boolean sub expression with *true* and *false*, replacement of each arithmetic operation with another one, e.g.: "*" with " /", replacement of each Boolean relation with another one, e.g.: > with >=, == .

These mutation operators are also called traditional mutation operators. There are also mutation operators for object-oriented languages, for concurrent constructions, complex objects like containers etc., they are called class-level mutation operators. In [3] a survey on the existing object oriented mutation techniques is presented. These techniques are critically reviewed on the basis of evaluation criteria designed by the authors by considering important aspects of mutation testing. These aspects can have their influence on mutation testing process. Another contribution of this work is a survey of available mutation testing tools.

One of the greatest challenges to the validity of mutation testing is the number of mutants that are semantically equivalent to the original program. *Equivalent mutants* produce the same output as the original program for every possible input. For seven large Java programs, 45% of the mutants not detected by the test suite were shown to be equivalent [5]. Equivalent mutants occur when the mutation can never be exercised, its effect is later cancelled out or it is corroded away by other operations in the program [6]. Determining which mutants are equivalent is a tedious activity, usually not implemented in tools. The impact of equivalent mutants is studied in [7]. Techniques have been devised to identify equivalent mutants using program slicing [8], compiler optimization [9], constraint solving [10] and, more recently, impact assessment [7]. Equivalent mutants are still however difficult to remove completely.

Mutation score is a kind of quantitative test quality measurement that examines a test set's effectiveness. It is defined as a ratio of the number of killed mutants to the total number of non-equivalent mutants. The total number of nonequivalent