Master thesis
15 credits

Goran Brestovac and Robi Grgurina

Applying Multi-Criteria Decision Analysis Methods in Embedded Systems Design

Västerås, Sweden
September 2013
ABSTRACT

In several types of embedded systems the applications are deployed both as software and as hardware components. For such systems, the partitioning decision is highly important since the implementation in software or hardware heavily influences the system properties. In the industry, it is rather common practice to take deployment decisions in an early stage of the design phase and based on a limited number of aspects. Often such decisions are taken based on hardware and software designers’ expertise and do not account the requirements of the entire system and the project and business development constraints. This approach leads to several disadvantages such as redesign, interruption, etc. In this scenario, we see the need of approaching the partitioning process from a multiple decision perspective. As a consequence, we start by presenting an analysis of the most important and popular Multiple Criteria Decision Analysis (MCDA) methods and tools. We also identify the key requirements on the partitioning process. Subsequently, we evaluate all of the MCDA methods and tools with respect to the key partitioning requirements. By using the key partitioning requirements the methods and tools that the best suits the partitioning are selected. Finally, we propose two MCDA-based partitioning processes and validate their feasibility thorough an industrial case study.

Date: September 2013
Carried out at: MDH and ABB Corporate Research, Sweden
Advisor at MDH: Gaetana Sapienza
Advisor at ABB: Gaetana Sapienza
Examiner: Ivica Crnkovic
ACKNOWLEDGEMENTS

First of all we would like to thank Tihana Galinac Grbac, professor at Faculty of Engineering at University of Rijeka, for encouragement, advice and help for taking the opportunity to write our master thesis at the School of Innovation, Design and Engineering (IDT) at Mälardalen University (MDH).

Special thanks to Gaetana Sapienza our thesis supervisor at IDT for doing a splendid job at providing a lot of valuable advices, good ideas and a great guidance throughout the master thesis.

We would also like to thank Ivica Crnkovic, the master thesis examiner at MDH for providing a good environment, facilities for completing this master thesis, and all help and support while staying at MDH.

Last but not least we would like to thank to our parents for the support.
PREFACE

In today’s industrial applications based on component design, some of the components may be implemented in software or in hardware based on a number of different and conflicting requirements. The decision impacts performance, power consumption, price, reliability and other system properties.

The system must fulfill all requirements taking into consideration all the constraints. Constraints could be project limitations like development time, budget, project deadline etc. or client strategic decisions like vendor loyalty. There are also a lot of interactions between components in the system which could lead to a multiple dependencies between them.

This leads to an optimization problem with multiple criteria. The criteria could be defined differently for software and hardware components. The criteria values for some components could be defined as ranges, might be missing or estimated. Multiple criteria decision analysis (MCDA) methods could be useful approach for solving the hardware/software partitioning problem.

Västerås, September 2013

Goran Brestovac

Robi Grgurina
# NOMENCLATURE

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHP</td>
<td>Analytic Hierarchy Process</td>
</tr>
<tr>
<td>ANP</td>
<td>Analytic Network Process</td>
</tr>
<tr>
<td>CRSA</td>
<td>Classical Rough Set Approach</td>
</tr>
<tr>
<td>DM</td>
<td>Decision Maker</td>
</tr>
<tr>
<td>DRSA</td>
<td>Dominance-based Rough Set Approach</td>
</tr>
<tr>
<td>ELECTRE</td>
<td>ELimination Et Choix Traduisant la REalité (ELimination and Choice Expressing the REality)</td>
</tr>
<tr>
<td>ER</td>
<td>Evidential Reasoning</td>
</tr>
<tr>
<td>HW</td>
<td>Hardware</td>
</tr>
<tr>
<td>IDS</td>
<td>Intelligent Decision System</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>MCDA</td>
<td>Multi-Criteria Decision Analysis</td>
</tr>
<tr>
<td>MCDM</td>
<td>Multiple-Criteria Decision-Making</td>
</tr>
<tr>
<td>NA</td>
<td>Not Available</td>
</tr>
<tr>
<td>NS</td>
<td>Not Specified</td>
</tr>
<tr>
<td>PROMETHEE</td>
<td>Preference Ranking Organization Method for Enrichment Evaluations</td>
</tr>
<tr>
<td>ROSE</td>
<td>Rough Set Data Explorer</td>
</tr>
<tr>
<td>RSA</td>
<td>Rough Set Approach</td>
</tr>
<tr>
<td>SMAA</td>
<td>Stochastic Multicriteria Acceptability Analysis</td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>TOPSIS</td>
<td>Technique for Order Preference by Similarity to Ideal Solution</td>
</tr>
<tr>
<td>VIKOR</td>
<td>VIskeKriterijumska Optimizacija I Kompromisno Resenje (Multicriteria Optimization and Compromise Solution)</td>
</tr>
</tbody>
</table>
## CONTENTS

### Chapter 1 INTRODUCTION

1.1 Motivation ................................................................. 1  
1.2 Research questions ................................................... 2  
1.3 Contribution .................................................................. 3  
1.4 Structure of the Thesis ................................................ 3  

### Chapter 2 PARTITIONING

2.1 Hardware/software partitioning ...................................... 4  
  Related works ................................................................. 5  

### Chapter 3 MCDA METHODS

3.1 About MCDA .................................................................. 6  
3.2 Ranking ........................................................................ 8  
  AHP .............................................................................. 8  
  ANP ........................................................................... 14  
  TOPSIS ....................................................................... 16  
  VIKOR ....................................................................... 17  
  PROMETHEE I AND II .................................................. 20  
  EVIDENTIAL REASONING ............................................ 22  
3.3 Classification .................................................................. 24  
  RSA ............................................................................ 24  
  DRSA .......................................................................... 27  
3.4 Ranking and Classification ............................................ 29  
  Electre III ..................................................................... 29  
  Electre IV ..................................................................... 31  
  ELECTRE TRI .............................................................. 33  
  SMAA-2 ..................................................................... 34  
  SMAA-TRI .................................................................. 36  

### Chapter 4 MCDA TOOLS

4.1 Ranking ........................................................................ 38  
  Electre III/IV ................................................................. 38  
  MakeItRational .............................................................. 39  
  ANP Solver ................................................................. 42  
  Web ANP Solver .......................................................... 43  
  TransparentChoice ........................................................ 44  
  Triptych .................................................................... 47  
  SANNA ....................................................................... 48  
  Visual PROMETHEE Academic ........................................ 49  
  Intelligent Decision System .......................................... 52  
4.2 Classification .................................................................. 55  
  Electre TRI ................................................................. 55  
  4eMka2....................................................................... 56  
  jMAF ......................................................................... 57  
  ROSE2 ....................................................................... 60  
4.3 Ranking and Classification ............................................ 62  
  JSMAA ....................................................................... 62  

### Chapter 5 MCDA METHODS AND TOOLS FOR ENABLING THE PARTITIONING

vii
Chapter 1
INTRODUCTION

1.1 Motivation

In many industrial embedded systems an application could be implemented as software or as hardware executable unit. The process that determines if the application will be deployed as hardware or software is called partitioning. The partitioning process is of key importance to guarantee the application properties such as reliability, sustainability, efficiency, etc. Widely used approaches for carrying out the partitioning are based on hardware as well as software expertise, however they are not combined in synergic ways and the decisions are taken too early. As a consequence, this approach causes a number of issues such as redesign, interruption etc., which negatively affect the overall development process, the performance of the system and its lifecycle. Indeed, the decisions upon how the partitioning is carried out are very important because they affect many system properties. There are also several requirements that the system must satisfy. Additionally depending on the system size and complexity of the application there might be some dependencies between the components that have to be taken into consideration during the partitioning process.

According to the scientific papers the partitioning process is currently done based on a limited number of criteria. The partitioning is usually looked at as an optimization problem and is limited to few parameters. Parameters are usually hardware/software size, performance/timing constraints, power consumption and cost. These are the most important criteria but in practice there are a lot of other criteria that influence the partitioning process. The criteria are of different types (functional and non-functional) and some of the values could be missing or just estimated. Additionally, there might also be a number of existing components that could be reused instead of the development of the new ones. In this context it is of crucial importance to threat the partitioning problem from a multiple criteria perspective, in order to be able of taking partitioning decision on a large and even conflicting number of properties.

In this thesis we focus on defining the HW/SW partitioning process that could be used with a greater number of criteria utilizing existing MCDA methods and tools to help the decision-maker (DM) make the decision. In the process we take into consideration the reusability aspect, trying to find the best alternative from the set of existing component variants if they are available.
1.2 Research questions

In this section we introduce the research questions which have lead and guided this thesis work.

1. Could MCDA methods and tools be used for HW/SW partitioning?

Numerous MCDA methods and tools used both for ranking and classification have been proposed and studied, but only few of them have been applied to HW/SW partitioning process [1]. By answering two minor research questions (a, b) it will be also possible to answer to this main research question and find out which MCDA methods, tools and processes are able to satisfy the proposed HW/SW partitioning process requirements and what are their limitations.

a. What MCDA methods and tools exist and what are their characteristics?

It is necessary to perform a systematic research and identify a number of HW/SW partitioning process requirements that will be used for evaluating MCDA methods and tools. For example, handling missing and qualitative values could represent the characteristics extracted from the requirements. Subsequently, the most suitable tools and methods with respect to the requirements for the partitioning process will be selected and applied to the wind turbine case study.

b. What kind of MCDA-based process could enable HW/SW partitioning?

We will examine and find out what kind of MCDA-based process could satisfy the HW/SW partitioning process requirements. For example, the scalability of the process (i.e. it must not be restricted to a limited number of units or alternatives) could represent one process requirement.
1.3 Contribution

In this thesis the hardware/software (HW/SW) partitioning processes that are able to handle a large number of criteria are proposed and presented on a practical example. To be able to do that the analysis of the MCDA methods and tools has been carried out. The partitioning process requirements have been identified and the key partitioning process requirements have been used in the evaluation of the MCDA methods and tools. Overall 14 methods and 14 tools have been analysed. The two partitioning processes have been created to enable the partitioning using the existing methods and tools. The first one is called the ranking process in which the existing component alternatives are available and the data is available for new hardware and/or software components. The second one is called the classification process in which the component is newly designed and there are no component alternatives available. In this case the component is defined by its criteria and the decision has to be made whether it is better to implement it as software or hardware. If the system consists of new and existing components, both processes could be used to partition the system, depending on the particular components. The ranking process could be used for partitioning components which have existing component alternatives to choose from, while the classification process could be applied in case of a new component for which there are no existing component alternatives to choose from. Both processes have been applied on the wind turbine case study.

1.4 Structure of the Thesis

This master thesis is organized as follows. The introduction to the thesis is provided in chapter 1. In chapter 2 the concept of partitioning process is introduced and the related works in the field are presented. Chapter 3 provides the background on the MCDA followed by the systematic research in which a number of the MCDA methods both for ranking and classification are identified. Chapter 4 provides the description of tools that implement these methods. The HW/SW partitioning process requirements are proposed in chapter 5. The number of the most important criteria are extracted from those requirements and used for evaluating the MCDA methods and tools. The final result of evaluation is presented in chapter 6 where the most suitable tools and methods for the partitioning process are selected. Two possible scenarios (i.e. ranking and classification scenarios) in the partitioning process are further proposed followed by the wind turbine case study of how processes work. Chapter 7 contains the discussion of the research questions and suggestions for the future work. The final conclusions are presented in chapter 8. Appendix A contains the methods table that consists of 14 MCDA methods and 32 criteria extracted from the HW/SW partitioning process requirements. Appendix B presents the tools table that consists of 14 MCDA tools and 27 criteria that are also extracted from the HW/SW partitioning process requirements. The ids file used for the wind turbine case study with the training data is presented in the Appendix C and the file with the testing data in the Appendix D.
Chapter 2

PARTITIONING

2.1 Hardware/software partitioning

The process for the development of systems that consist of software and hardware components is called hardware-software codesign. It deals with meeting the system requirements and trying to keep the project within time and budget constraints. In embedded systems some components might be implemented both in hardware and in software. Depending on the system and different component requirements, one of the two options has to be chosen. The hardware-software codesign is characterized by the concurrent development of hardware and software components.

For the system to be implemented, first the system specifications are defined. After the specification process the system model is created. When the model is verified the HW/SW partitioning could be done. After the partitioning the software and hardware components are developed separately at the same time followed by the system integration and testing. Verification and validation are done in the end [2], [3].

![Hardware-software codesign process](image)

Figure 1. Hardware-software codesign process

The partitioning decisions are usually based on a limited number of criteria, usually hardware size, performance/timing constraints and power consumption. There are different approaches to the partitioning depending on the most important criterion and the number of criteria that are taken into consideration. In some cases it might be good enough to optimize one parameter, for example minimize the cost or hardware size while providing the acceptable level of performance.
There are many previous works done to HW/SW partitioning and numerous approaches like genetic algorithms [4], greedy heuristics [5], dynamic programming [6], simulated annealing algorithms [7] have been developed since the early 1990s.

One of these works models the Hardware Software partitioning problem as a Constraint Satisfaction Problem (CSP) [8]. The near optimal solution is obtained by solving a CSP using the Genetic algorithm approach where three types of constraints (cost, timing and concurrency) are considered.

Another paper presents two partitioning algorithms used in a HW/SW partitioning process [9]. The first one is an Integer Linear programming (ILP) based approach and the second one is genetic algorithm (GA). The partitioning problem is represented using the graph model in which nodes are the components of the system that have to be partitioned, and the edges represent communication between the components. Empirical tests have shown that the ILP-based solution works efficiently for graphs with several hundreds of nodes and produces the optimal solutions, whereas the genetic algorithm works efficiently for graphs of even thousands of nodes and gives near-optimal solutions.

The integer programming approach for solving the HW/SW partitioning problem and leading to (nearly) optimal results is also presented in another work [10]. This approach is based on the idea of „using the tools“ for cost estimation and nearly optimal results are calculated in short time because the high computation time of solving IP-models is reduced by developing algorithm that splits the partitioning approach in two phases.

A technique for allocating hardware resources for partitioning has been proposed in another work [11]. The estimated HW/SW partition is built during the allocation algorithm. The components are taken one by one and then the most critical building block of the current component is allocated to hardware.
Chapter 3
MCDA METHODS

3.1 About MCDA

MCDA is an acronym that stands for Multiple Criteria Decision Analysis/Aiding, and it is sometimes referred as MCDM (Multiple Criteria Decision Making). It is a subdiscipline of operational research. Operational research is often considered to be a subfield of mathematics that applies advanced analytical methods to get optimal or near-optimal solutions in complex decision-making problems. Operational research is focused on practical problems in marketing, manufacturing, transportation, information technology (IT) and other fields. Therefore operational research overlaps with other disciplines, particularly operations management and engineering science. MCDA is a subdiscipline of operational research that explicitly deals with decision problems that use multiple criteria to determine the best possible solution [12].

The discipline was started in the 1960s and it evolved over time as new type of problems arose and had to be solved.

MCDA is applied in many fields like medicine, forestry, economics, management, transportation, etc. to solve a vast number of different problems. Indeed most decisions that are not easy to be taken, because they involve multiple and conflicting objectives. For instance:

- Procurement: who is the best supplier? [13]
- Key Performance Indicators: how to evaluate performance of business units? [14]
- Portfolio Management: how to compose financial assets portfolio? [15]
- Location: what is the best place to build a new facility (plant, warehouse, hypermarket, etc...)? [16]
- Health Care: which criteria should be taken into account when introducing new technologies into the health care system? [17]
- Sustainable Development: what is the best way to achieve sustainable development? [18]

In each case, one or several persons (DMs or stakeholders) have to compare different solutions (or actions) with several objectives in mind. For instance:

- Procurement: minimize the price paid, maximize the quality of the product purchased, etc.
- Key Performance Indicators: minimize costs, maximize profit, etc.
- Portfolio Management: minimize risk, maximize expected return, etc.
- Location: minimize investment cost, maximize expected return, etc.
Health Care: maximize efficiency, minimize side effects, etc.

Sustainable Development: reduce environmental impacts, reduce social impacts, etc.

The degree of achievement of these objectives can be measured by defining appropriate quantitative or qualitative evaluation criteria and related metrics such as for instance:

- the price of an equipment (in € or in $),
- a qualitative measurement of social impact (on a scale such as: very low, low, moderate, high or very high),

These criteria are often conflicting with each other, for instance:

- Usually higher quality equipment is more expensive.
- Taking care of environmental issues can have a negative impact on profit.
- Building a new hypermarket closer to a big city will cost more money but will bring a higher level of expected return.

That's why most of decisions are difficult to make. So what decision is the best, which account, traded-off/satisfy all of the required objectives? It depends on DM's priorities and preferences.

MCDA methods are designed to assist DMs in such a context.

Today there are a lot of different methods available which are able to choose the best alternative, the best subset of alternatives, rank alternatives or classify/sort them into predefined preference classes [19]. For a lot of these methods there are software tools available to help the DM with the decision making. Most of the tools implement only one method, but there are some tools that implement more than one method. Unfortunately, some tools do not implement the complete method(s), but a simplified version where some steps are omitted or simplified.

Numerous methods have been proposed for dealing with complex problems in decision making. Specifically, in this thesis we are considering the problem of choosing the best alternative from a finite set of alternatives with the respect to criteria and extracting the classification rules from a set of already classified examples which can then be used to do the partitioning of the new data set. The rules are presented as “if...then...” sentences.

We studied a number of different methods in order to find the one that satisfies as many requirements as possible. The list of requirements is defined in chapter 5. The MCDA methods used in this thesis are AHP, ANP, VIKOR, PROMETHEE, TOPSIS, ELECTRE, Evidential Reasoning, SMAA, RSA and DRSA. Not surprisingly, the proposed methods may result in a different indication of ranking of all alternatives or choosing the best alternative with respect to all criteria.

A general multicriteria decision problem can be defined through table 1 where the set of alternatives (i.e. a possible decision or an item to evaluate) is represented by \( A = \{a_1, \ldots, a_n\} \), and the set of criteria (i.e. attributes associated to each alternative that makes it possible to evaluate the alternatives and to find the most suitable ones) by \( F = \{f_1, \ldots, f_m\} \). The evaluation value (i.e. number associated to each alternative with respect to criterion) of alternative \( a_i \), with the respect to the \( j \)-th criterion can be defined by \( f_j(a_i) \). The main goal of DM is to find the alternative which best optimizes/trades-off all the criteria. The solution of multicriteria decision problem depends not only on the alternatives' values, but it is also mostly affected by different criteria weights assigned by different DMs.
Table 1. MCDA table

<table>
<thead>
<tr>
<th>aj</th>
<th>f1(aj)</th>
<th>f2(aj)</th>
<th>...</th>
<th>fj(aj)</th>
<th>...</th>
<th>fk(aj)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>f1(a1)</td>
<td>f2(a1)</td>
<td>...</td>
<td>f1(a1)</td>
<td>...</td>
<td>f1(a1)</td>
</tr>
<tr>
<td>a2</td>
<td>f1(a2)</td>
<td>f2(a2)</td>
<td>...</td>
<td>f1(a2)</td>
<td>...</td>
<td>f1(a2)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>ai</td>
<td>f1(ai)</td>
<td>f2(ai)</td>
<td>...</td>
<td>f1(ai)</td>
<td>...</td>
<td>f1(ai)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>an</td>
<td>f1(an)</td>
<td>f2(an)</td>
<td>...</td>
<td>f1(an)</td>
<td>...</td>
<td>f1(an)</td>
</tr>
</tbody>
</table>

Detailed method histories, descriptions, properties and limitations together with available tools are presented in the following subchapters.

### 3.2 Ranking

Ranking is the process of putting a set of alternatives in order from the most desirable to the least desirable based on values of multiple criteria. Since usually not all criteria are equally important, criteria weights are used to express the criteria importance.

#### AHP

The Analytic Hierarchy Process (AHP) is one of the most popular mathematical method applied for MCDA. It was developed by Thomas L. Saaty in the 1970s and it can be defined as a structured technique for organizing and analysing complex decisions [20]. Initially developed for contingency planning [21], the method can be used for solving decision making problems in almost any type of subject.

The main objective of the AHP method is to get ranking of a finite set of alternatives in terms of a finite number of decision criteria. It is based on establishing preferences between criteria and alternatives using pairwise comparisons. The most suitable alternative for a DM is always the best ranked.

As shown on the Figure 2 it is possible to use as input to AHP method actual measurements like height, weight and price or use subjective opinions like reliability, satisfaction and quality. The output of the AHP method, as shown on the Figure 2, are the ratio scales that describe DMs preference in comparing two elements in decision making process using intensity of importance defined in table 3. The ratio scales are obtained from the Eigen vectors that are used to derive the priorities between compared elements.

Another output is consistency index (CI) obtained from the Eigen values that are used to derive priority vectors. Saaty proposed the consistency ratio to check the consistency of comparing two elements in decision making process. The consistency ratio (CR) is calculated by dividing the consistency index (CI) by the random consistency index (RI).
\[ CR = \frac{CI}{RI} \]  

Table 2. Random consistency index table (n represents the number of items compared in matrix)

<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RI</td>
<td>0</td>
<td>0</td>
<td>0.58</td>
<td>0.9</td>
<td>1.12</td>
<td>1.24</td>
<td>1.32</td>
<td>1.41</td>
<td>1.45</td>
<td>1.49</td>
</tr>
</tbody>
</table>

The random consistency index is obtained from the table 2 with respect to number of elements compared in matrix that is defined in table 4. If the value of consistency ratio is smaller than 0.01, the inconsistency is acceptable. If the consistency ratio is greater than 0.01, DM needs to revise its subjective judgments. If the value equals to 0 then that means that the judgments are consistent.

![Input and output values of AHP](image)

Figure 2. Input and output values of AHP

The hierarchical structure of the AHP method is shown on the Figure 3. It consists of a goal at the top, criteria influencing the goal in the next level down, possibly subcriteria in levels below that and alternatives of choice at the bottom of the model. In the AHP method the independence among criteria, subcriteria or alternatives is assumed. If the decision problem assumes dependence among criteria, subcriteria or alternatives, the ANP method is more appropriate and will be further discussed.
AHP involves ten steps, as follows:

1. Define the main goal of your decision problem.
2. Structure elements of the decision problem in groups of criteria, sub-criteria, alternatives.
3. Construct a pairwise matrix as shown by the example in table 4.
   a. All criteria are compared one-to-one with the other criteria using the Saaty scale as proposed in table 3, e.g. if something is considered twice as important as something else, a weight of 2 is given in the matrix for that relation. Apply the same comparison procedure for comparing sub-criteria and alternatives.
   a. Each weight is divided by the sum of all weights in each matrix column.
5. Deriving a priority vector, i.e. the resulting relative weights
   a. The sum of each row of normalized weights gives a priority vector.
6. Calculating a maximum Eigen value vector.
   a. The product of the pairwise matrix (step 3) and the priority vector (step 5).
7. Calculating the consistency index.
   a. The sum of the values in a maximum Eigen value vector is subtracted by the number that represents the size of the comparison matrix (e.g. the size is equal to 3 if there are 3 alternatives) and divided by the size of the comparison matrix minus 1.
8. Calculating the consistency ratio.
   a. The consistency ratio is calculated by dividing the consistency index (CI) by the random consistency index. A consistency check is made to see if the ratio is smaller than 0.1. If the value of consistency ratio is smaller or equal to 0.1, the inconsistency is acceptable. If the inconsistency ratio is greater than 0.1, there is need to revise the subjective judgment.
9. Evaluate the criteria and alternatives with respect to the weighting.

10. Get ranking.

The pairwise comparison judgments used in the AHP pairwise comparison matrix are defined as shown in the Fundamental Scale of the AHP in the table below.

Table 3. The fundamental scale of AHP

<table>
<thead>
<tr>
<th>Intensity of importance</th>
<th>Definition</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Equal importance</td>
<td>Two elements contribute equally to the objective</td>
</tr>
<tr>
<td>3</td>
<td>Moderate importance</td>
<td>Experience and judgment slightly favor one element over another</td>
</tr>
<tr>
<td>5</td>
<td>Strong importance</td>
<td>Experience and judgment strongly favor one element over another</td>
</tr>
<tr>
<td>7</td>
<td>Very strong importance</td>
<td>An activity is favored very strongly over another</td>
</tr>
<tr>
<td>9</td>
<td>Absolute importance</td>
<td>The evidence favoring one activity over another is of the highest possible order of affirmation</td>
</tr>
<tr>
<td>2,4,6,8</td>
<td>Used to express intermediate values</td>
<td></td>
</tr>
<tr>
<td>Decimals</td>
<td>1.1, 1.2, 1.3, …1.9</td>
<td>For comparing elements that are very close</td>
</tr>
<tr>
<td>Rational numbers</td>
<td>Ratios arising from the scale above that may be greater than 9</td>
<td>Use these ratios to complete the matrix if consistency were to be forced based on an initial set of ( n ) numerical values</td>
</tr>
<tr>
<td>Reciprocals</td>
<td>If element ( i ) has one of the above nonzero numbers assigned to it when compared with element ( j ), then ( j ) has the reciprocal value when compared with ( i )</td>
<td>If the judgment is ( k ) in the ((i, j)) position in matrix ( A ), then the judgment ( 1/k ) must be entered in the inverse position ((j, i)).</td>
</tr>
</tbody>
</table>

To compare \( n \) elements in pairs construct an \( n \times n \) pairwise comparison matrix \( A \) of judgments expressing dominance. For each pair choose the smaller element that serves as the unit and the judgment that expresses how many times more is the dominant element. Reciprocal positions in the matrix are inverses, that is \( a_{ij} = 1/a_{ji} \).
The AHP process is here described in details. The goal represents the parent node of the criteria and they comprise one of the comparison groups. The criteria will be pairwise compared with respect to the goal. The pairwise comparison judgments are made using table 3 and the judgments are arranged in a matrix, called the pairwise comparison matrix.

Considering an example comprising of 4 criteria, the numbers in the cells in an AHP matrix as shown on table 4, by convention, indicate the dominance of the row element over the column element. In the AHP pairwise comparison matrix in table 4 the (Criteria_2, Criteria_3) cell has a judgment of 3 in it, meaning Criteria_2 is moderately important than Criteria_3. So logically this means Criteria_3 has to be 1/3 as important as Criteria_2. Thus, 1/3 needs to be entered in the (Criteria_3, Criteria_2) cell. The diagonal elements are always 1, because an element equals itself in importance.

There will be 6 judgments required for these 4 elements. If the number of elements is \( n \) then the number of judgments is \( n(n-1)/2 \) to do the complete set of judgments.

Table 4. The AHP pairwise comparison matrix

<table>
<thead>
<tr>
<th></th>
<th>Criteria_1</th>
<th>Criteria_2</th>
<th>Criteria_3</th>
<th>Criteria_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteria_1</td>
<td>1</td>
<td>1/4</td>
<td>1/3</td>
<td>1/2</td>
</tr>
<tr>
<td>Criteria_2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>3/2</td>
</tr>
<tr>
<td>Criteria_3</td>
<td>3</td>
<td>1/3</td>
<td>1</td>
<td>1/3</td>
</tr>
<tr>
<td>Criteria_4</td>
<td>2</td>
<td>2/3</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

The priorities of an AHP pairwise comparison matrix are obtained by solving the principal Eigen vector of the matrix. The mathematical equation for the principal Eigen vector \( w \) and principal Eigen value \( \lambda_{\text{max}} \) of a matrix \( A \) is presented by equation (2). It says that if a matrix \( A \) times a vector \( w \) equals a constant (\( \lambda_{\text{max}} \) is a constant) times the same vector, that vector is an Eigen vector of the matrix. Matrices have more than one Eigen vector; the principal Eigen vector which is associated with the principal Eigen value \( \lambda_{\text{max}} \) (that is, the largest Eigen value) of \( A \) is the solution vector used for an AHP pairwise comparison matrix. The Eigen vector of the matrix shown in table 4 is depicted in table 5 and the most important criteria with the respect to goal is Criteria_2.

\[
Aw = \lambda_{\text{max}} w
\]

\( (2) \)
Table 5. Eigen vector for matrix shown in table 4

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Priorities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteria_1</td>
<td>0.0986</td>
</tr>
<tr>
<td>Criteria_2</td>
<td>0.425</td>
</tr>
<tr>
<td>Criteria_3</td>
<td>0.1686</td>
</tr>
<tr>
<td>Criteria_4</td>
<td>0.3077</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1.000</strong></td>
</tr>
</tbody>
</table>

Perron's Theorem [22] shows that for a matrix of positive entries, there exists a largest real Eigen vector and its Eigen vector has positive entries. This is an important theorem that supports the use of the Eigen vector solution in AHP theory to obtain priorities from a pairwise comparison matrix. The book [23] gives more details about the mathematics background of the pairwise comparison matrix.

The AHP uses a data structure called a Super Matrix that contains priorities of the children from a comparison group of children and priorities of the parent from the comparison group of the parent. The name Super Matrix is used for this matrix because it is made up of column vectors of priorities, each of which was obtained from a matrix. So in a way it is a matrix of matrices.

There are three Super Matrices:

1) The unweighted Super Matrix contains the derived priorities obtained from the pairwise comparison results.

2) The weighted Super Matrix, weighted by the importance of clusters, is important only in network models [24], and will be discussed in the next subsection. The weighted Super Matrix is the same as the unweighted Super Matrix for hierarchy model. In the weighted Super Matrix all columns must sum to zero.

3) The limit Super Matrix is the final version of the Super Matrix obtained by raising the weighted Super Matrix to powers until all columns are the same (i.e. until it converges).

The simple way to get the ranking of alternatives for a hierarchy is to multiply the priority of each element in the hierarchy (derived through pairwise comparisons) by the weight of its parent element and sum the bottom level priorities of the alternatives to get the final ranking.

However, the solution may also be obtained using a Super Matrix. The weighted Super Matrix is raised to powers until it converges to the limit Super Matrix which contains the final results, the priorities for the alternatives, as well as the overall priorities for all the other elements in the model.

Since the AHP method is based on establishing preferences between criteria and alternatives using pairwise comparisons, it only supports quantitative values as input values to matrices (i.e. qualitative values, handling missing data, etc. is not supported by this method).
One of the advantages using the AHP method is that it allows decomposing a large problem into its constituent parts using the hierarchical structure. When the problem is presented with the hierarchical structure it becomes very easy to explain to other people. Since AHP is one of the first methods used in multi criteria decision analysis, there are a lot of tools which implement this method. It is also possible to create its own Excel sheet implementing AHP method which can then be used for decision problems with different number of criteria.

On the other side, the disadvantage of using AHP method may be a limited number of criteria. Since it is necessary to perform n(n-1)/2 pairwise comparisons it is recommended to use not more than 10 criteria. DMs are often not used to give a pairwise comparisons between items (i.e. it may be difficult to decide if one alternative is strongly or more strongly preferable than another alternative). They are more used to give either a sorting or ranking based on the importance. If the consistency index is too high, it may be a problem to explain to people to make another pairwise comparisons and reconsider their inputs. This method suffers from the rank reversal problem (i.e. the ranking can be reversed when a new alternative is introduced).

**ANP**

The Analytic Network Process (ANP) method represents a generalization of the Analytic Hierarchy Process (AHP). It was developed in 1996 by Thomas L. Saaty and has been applied in many research papers such as research on evaluation index system of mixed-model assembly line [25], university-industry alliance partner selection method [26], partner selection method for supply chain virtual enterprises [27], etc.

The main objective of the ANP method is to get ranking of a finite set of alternatives in terms of a finite number of decision criteria. It is based on establishing preferences between criteria and alternatives using pairwise comparisons. The most suitable alternative for a DM is always the best ranked.

![Diagram of ANP](image)

Figure 4. The structure of ANP
The structure of ANP is shown on Figure 4. It consists of 4 clusters and 7 nodes. The decision nodes are organized in a network of clusters with links between the nodes going in either direction or both directions. The first cluster consists of Goal, the second cluster of Criteria 1 and 2, the third cluster of Criteria 3 and 4, and the fourth cluster of Alternative 1 and 2. As shown, in ANP nodes might be grouped in cluster whereas in AHP this is not possible. Thus, beside priorities in the comparison of one node to a set of other nodes, it is also possible to define cluster priorities with respect to the goal. Matrix is used to represent the network of ANP. This matrix is composed by listing all nodes horizontally and vertically. The connection and weight between one node (columns-header) to another node (row-header) of the network is presented entering values from the table 3. After populating all rows and columns, this matrix becomes so called Super Matrix consists of values expressing relative importance of one node/cluster to other node/cluster.

The pairwise comparison of nodes or clusters to each other’s and the calculation of local priorities are the same as in AHP method. After calculating the Eigen vector of the pairwise comparison matrix we get the local priorities arranged as column vectors in the super matrix. After all comparisons are completed we get an unweighted Super Matrix of the network model. The matrix then needs to be normalized and we get resulting weighted Super Matrix. The whole model is synthesized by calculating the weighted Super Matrix, taken to the power of \( k + 1 \) until it converges (in this case all columns are the same), where \( k \) is an arbitrary number. The weighted Super Matrix is called the Limit Matrix. The final result is ranking of the alternatives in network model.

Difference in structure between AHP and ANP is shown on Figure 5. A hierarchy in AHP is a network too, but a special kind of network with a goal cluster from which all the arrows lead away, and a sink cluster (the alternatives) that all the arrows lead into. Links go only downward in a hierarchy. A typical network has neither sinks nor sources; and the links can go in any direction. A network can more faithfully represent the relative world we really live in. Let’s consider an example of buying a car. One does not buy a car by determining in the abstract the importance of the criteria before going shopping and looking at a few cars. The available cars determine how important the criteria are. And when new cars are added to those being considered, the importance of the criteria may change. Thus, ANP reflects the way how we make decisions where the importance of criteria can change with the available alternatives.

Figure 5. The AHP structure (left) and ANP structure (right)
Since the ANP method is based on establishing preferences between criteria and alternatives using pairwise comparisons, it only supports quantitative values in range from 1 to 9 as input values (i.e. qualitative values, handling missing data, etc. is not supported by this method).

One of the advantages using ANP method is introduction of clusters and links between the elements going in either one direction or both directions. Thus, it is possible to deal with more complicated problems. Also, we can obtain much deeper understanding of a decision problem and interconnections between elements in the structure.

On the other side, the disadvantage of using ANP method may be a limited number of criteria and alternatives. Since it is necessary to perform \( n(n-1)/2 \) pairwise comparisons it is recommended to use not more than 5 criteria and alternatives in a cluster. People are often not used to give a pairwise comparison between items. They are more used to give either a sorting or ranking based on the importance. If the consistency index is too high, it may be a problem to explain to people to make another pairwise comparison and reconsider their inputs. Due to feedback loops and interconnections it may be very difficult to create own implementation of ANP in Excel spreadsheet, so it is necessary to obtain a specific software. This method suffers from the rank reversal problem (the ranking can be reversed when a new alternative is introduced).

**TOPSIS**

TOPSIS is an acronym for Technique for Order Preference by Similarity to Ideal Solution. It is multiple-criteria decision analysis method proposed by Hwang and Yoon in 1981. It is used for ranking the alternatives with a finite number of criteria.

The method is not limited to a specific field and so far has been applied in supply chain management and logistics [28], [29], business and marketing management [30], energy management [31], chemical engineering [32] etc.

TOPSIS method is getting into account the distance of alternatives from the positive-ideal and negative-ideal solution.

Paper [19] provides a method explanation and it was a source for the following formulas. It has five steps. The first step after creating the decision matrix is the normalization of the input data. The construction of the normalized decision matrix is done using the formula

\[
    r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}} \quad \text{for } i=1,\ldots,m; \ j=1,\ldots,n;
\]

where \( x_{ij} \) are original and \( r_{ij} \) are normalized values.

After that the weighted normalized decision matrix is created in the second step, taking into account the criteria importance with the formula \( v_{ij} = w_j r_{ij} \) where \( w_i \) is the weight of \( j \) criterion.

The positive and negative ideal solutions are determined in the step 3. Positive ideal solution is given by the following formula
The negative ideal solution is

\[ A^* = \{ (\max v_{ij} \mid j \in J), (\min v_{ij} \mid j \in J') \mid i = 1,2,...,m \} \]

\[ A' = \{ v_1', v_2', ..., v_j', ..., v_n' \} \] (4)

In step 4 the separation from the positive and the separation from the negative ideal alternative are calculated. The separation from the positive ideal solution is calculated by formula

\[ S_i^* = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_{ij}^*)^2} \quad i = 1,2,...,m \] (6)

and the separation from the negative ideal solution is given by formula

\[ S_i' = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_{ij}')^2} \quad i = 1,2,...,m \] (7)

In the fifth step the closeness coefficients are calculated.

\[ c_i^* = \frac{S_i'}{S_i^* + S_i'} \quad 0 < c_i^* < 1 \quad i = 1,2,...,m \] (8)

\[ c_i^* = 1 \quad \text{if} \quad A_i = A^* \]

\[ c_i^* = 0 \quad \text{if} \quad A_i = A' \]

The alternatives are ranked according to the descending order of the closeness coefficient values, so the higher the value the better the solution.

TOPSIS method supports quantitative values and it is relatively simple to use and implement. Therefore the method is widely used in numerous real-life problems. There are also multiple tools that support this method.

The negative side of the method is that it does not support uncertain or missing values. Like most of the MCDA methods it could suffer from rank reversal problem.

VIKOR

VIKOR is a method used in multicriteria decision analysis developed in 1998 by Serafim Opricovic. It has been applied in many areas such as neural network [33], multi-criteria decision making problems under intuitionistic environment [34], suppliers selection [35], etc.

The main objective of the VIKOR method is to evaluate several possible alternatives according to multiple conflicting criteria and rank them from the worst to the best one. The most suitable alternative for DM is always the best ranked.

It is based on ranking and selecting from a set of alternatives. The positive and negative ideal solutions are defined. The positive ideal solution is the alternative with the highest ranked value in terms of benefit, and the lowest ranked value in terms of cost. The negative
ideal solution is the alternative with the lowest ranked value in terms of benefit, and the highest ranked value in terms of cost.

The procedure of VIKOR method consists of several steps:

1. First we need to compute the best and the worst value of each criteria.
   - Suppose we have $m$ alternatives, and $n$ criteria. Various $j$ alternatives are denoted as $A^{(i)}$, $A^{(2)}$, ..., $A^{(g)}$. Let $f_{ij}$ presents the value of the $i_{th}$ criterion function for the alternative $A^{(j)}$. The best value for all $n$ criteria functions is denoted as $f_{ij}^*$ and presents the positive ideal solution for the $i_{th}$ criterion. The worst value for all $n$ criteria functions is denoted as $f_{ij}^-$ and presents the negative ideal solution for the $i_{th}$ criterion.

   If $i_{th}$ criterion function represents a benefit then the following equations can be obtained, as defined in [36]:
   
   $$f_{ij}^* = \max_i f_{ij}, \quad f_{ij}^- = \min_i f_{ij}$$  
   (9)

   If $i_{th}$ criterion function represents a cost then the following equations can be obtained, as defined in [36]:
   
   $$f_{ij}^- = \min_i f_{ij}, \quad f_{ij}^* = \max_i f_{ij}$$  
   (10)

2. For each alternative compute the ideal value $S_j$ (or utility measure) and negative value $R_j$ (or regret measure) using the following equations, as defined in [36]:

   $$S_j = \sum_{i=1}^{n} \omega_i \left( \frac{f_{ij}^* - f_{ij}}{f_{ij}^* - f_{ij}^-} \right)$$  

   $$R_j = \max_i \left[ \omega_i \left( \frac{f_{ij}^* - f_{ij}}{f_{ij}^* - f_{ij}^-} \right) \right]$$

   where $j = 1, ..., m$ and $i = 1, ..., n$  

   - $W_i$ are the weights of criteria expressing DM’s preferences and may be assigned using the AHP method.

3. For each alternative compute the synergy value $Q_j$ using the following equations, as defined in [36]:

   $$Q_j = v \left( \frac{S_j - S^*}{S^* - S^*} \right) + (1 - v) \left( \frac{R_j - R^*}{R^* - R^*} \right)$$

   $$S^* = \min_j S_j \quad S^- = \max_j S_j$$

   $$R^* = \max_j R_j \quad R^- = \min_j R_j$$

   where $j = 1, ..., m$  

   and $i = 1, ..., n$

   - $v$ represents the weight for the strategy of maximum group utility and can take any value from 0 to 1. It is usually set to value 0.5.

   - $(1 - v)$ represents the weight of the individual regret.
4. The final step consists of ranking the alternatives by measured values $S$, $R$ and $Q$ in decreasing order and then proposing a compromise solution consisted of the alternative $A^{(i)}$ which represents the best ranked solution by the measure $Q$ (minimum) [36].

- When the compromised solution is proposed, two conditions must be satisfied:
  
a) Acceptable advantage – ($A^{(2)}$ denotes the second best ranked alternative)

\[
Q(A^{(2)}) - Q(A^{(1)}) \leq DQ \quad \text{where} \quad DQ = \frac{1}{m - 1}
\]  

b) Acceptable stability in decision making - the proposed alternative $A^{(1)}$ must also be best ranked by $S$ and/or $R$.

- If one of the conditions mentioned above is not satisfied, alternative solutions are:
  
a) Alternatives $A^{(1)}$ and $A^{(2)}$ if the condition b) is not satisfied.
  
b) Alternatives $A^{(1)}$, $A^{(2)}$, ..., $A^{(m)}$ if the condition a) is not satisfied. $A^{(m)}$ is determined by the relation $Q(A^{(M)}) - Q(A^{(1)}) < DQ$ for maximum $M$.

Compared to AHP method, in VIKOR method it is not necessary to perform consistency test, as discussed in AHP chapter. This method supports only quantitative values and it is relatively simple to use and implement.

The VIKOR method has a significant advantage over the other ideal point method, such as TOPSIS method, because the “VIKOR algorithm can order directly without considering that the best solution is closer to the ideal point or more farther to the worst ideal point” [37].

On the other side, there is no available tool developed that implement this method. VIKOR is not able to handle incomplete and uncertain information. This method suffers from the rank reversal problem (i.e. the ranking can be reversed when a new alternative is introduced).
PROMETHEE I and II

PROMETHEE is an acronym for Preference Ranking Organization Method for Enrichment Evaluations. It was developed in the 1980s and applied in many areas such as automotive sector [38], web service selection [39], exploration strategies for rescue robots [40], suppliers evaluation [41], etc.

It is based on building an outranking on the set of alternatives in which the main objective is to find the alternative that is the most suitable for DM. Outranking methods are based on a more familiar way of thinking. Instead of trying to define what is good and what is bad, it is usually much easier to compare one solution to another. That is the underlying principle of outranking methods.

There are different types of PROMETHEE methods (and their extensions). The family of PROMETHEE methods is presented in Table 6.

Table 6. Family of PROMETHEE methods

<table>
<thead>
<tr>
<th>Name of method</th>
<th>Short description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROMETHEE I</td>
<td>Used for a partial ranking – considers only dominant characteristics of one alternative over other. Based on the positive and negative flows.</td>
</tr>
<tr>
<td>PROMETHEE II</td>
<td>Used for a complete ranking – considers both dominant and outranked characteristics of one alternative over other. Based on the net flow.</td>
</tr>
<tr>
<td>PROMETHEE III extension</td>
<td>Extension to PROMETHEE II which is based on interval ranking of alternatives [44].</td>
</tr>
<tr>
<td>PROMETHEE IV extension</td>
<td>Extension to PROMETHEE II which includes the infinite set of actions (continuous set of possible alternatives), but was never further developed nor applied [42].</td>
</tr>
<tr>
<td>PROMETHEE V extension</td>
<td>Extension to PROMETHEE II that supports the optimization under constraints [43].</td>
</tr>
<tr>
<td>PROMETHEE VI extension</td>
<td>Extension to PROMETHEE II which includes the sensitivity analysis procedure into decision making process [44].</td>
</tr>
</tbody>
</table>

The PROMETHEE I and II methods consist of several steps that are subsequently described. In the first step a preference function has to be associated with each criterion in
order to reflect the perception of the criterion scale by the DM. Usually the preference function \( P_j(a_i, a_k) \) is a non-decreasing function of the difference \( f_j(a_i) - f_j(a_k) \) between the evaluations of two alternatives \( a_i \) and \( a_k \). Several typical shapes of preference functions are proposed in the literature \([45]\), and indications are given on the way to select appropriate functions for different types of criteria. The value of \( P_j(a_i, a_k) \) is a number between 0 and 1. It corresponds to the degree of preference that the DM express for \( a_i \) over \( a_k \) according to criterion \( f_j \). Value 0 corresponds to no preference at all while 1 corresponds to a full preference.

In a second step the DM assesses numerical weights to the criteria to reflect the priorities: more important criteria receive larger weights. We note \( w_j \) the weight of criterion \( f_j \) and we assume that the weights are normalized (sum of all weights are equal to one). A multicriteria pairwise preference index is then computed as a weighted average of the preference functions:

\[
\Pi(a_i, a_k) = \sum_{j=1}^{k} \omega_j P_j(a_i, a_k)
\]  

(14)

Three preference flows are then computed in order to globally evaluate each alternative with respect to all other ones. The positive flow is a measure of the strength of an alternative \( a_i \) with respect to the other ones, the negative flow is a measure of the weakness of an alternative \( a_i \) with respect to the other ones. Finally the net flow is the balance between the two first ones. Obviously the best alternative should have a high positive value (close to 1) and a low negative value (close to 0), and thus a high positive balance value.

PROMETHEE I ranking is obtained by looking at the positive flow value and negative flow value. It is a partial ranking as the two flows usually give a different ranking of the alternatives because they synthesize the pairwise comparisons of the alternatives in two different ways. A traditional way to present PROMETHEE I is via a network representation where arrows indicate preferences. However that representation doesn’t give any visual information about the differences between the flow values. It is for instance difficult to appreciate how the ranking would be influenced by small changes in the weighting of the criteria.

PROMETHEE II final ranking is obtained by net flow values. All alternatives are compared but the differences between the positive and entering flows are lost, leading to a possibly less robust ranking.

These methods are able to handle only quantitative and missing values in information table. When the missing value is encountered, the PROMETHEE I and II methods perform all the pairwise comparisons and the resulting preference degree is set to zero as if both alternatives had equal evaluations on that criterion.

The advantages of using PROMETHEE II methods are listed as follows. The PROMETHEE II method classifies alternatives which are difficult to be compared because of a trade-off relation of evaluation standards as non-comparable alternatives. It is not necessary to perform a pairwise comparisons again when comparative alternatives are added or deleted. \([46]\)

On the other side, these methods suffer from the rank reversal problem (the ranking can be reversed when a new alternative is introduced).

21
EVIDENTIAL REASONING

Evidential reasoning is an MCDA method that is applied in many areas such as object recognition system [47], remote sensing images [48], evaluating students learning ability [49], fault diagnosis [50], etc.

The main objective of the method is ranking the alternatives with a finite number of criteria.

This method is also able to address uncertainties such as incomplete information using belief structure based on Dempster-Shafer theory of evidence [51]. The missing data in incomplete decision matrix are denoted by “*”. Dempster-Shafer theory provides methodology for solving incomplete decision matrix. It is necessary to ensure that the values in decision matrix are qualitative. If they are quantitative, the direct application of this approach may lead to large errors.

For example, when assessing the quality of specific hardware-software component for our system, we can have five evaluation grades:

\[
H = \{H_1, H_2, H_3, H_4, H_5\} = \{\text{Worst, Poor, Average, Good, Excellent}\}
\]

(15)

Suppose we have \(G\) alternatives \(A_j (j = 1,...,G)\) and we need to evaluate them with respect to \(T\) criteria \(C_i (i = 1,...,T)\). The assessment of a criterion \(C_i\) on an alternative \(A_j\) where \(\beta_{n}^{i}\) presents the degree of belief that the criterion \(C_i\) is assessed to the evaluation grade \(H_n\) is presented using the belief structure:

\[
S\left(C_i \left( A_j \right) \right) = \left( \beta_{1}^{i}, H_{1} \right), \left( \beta_{2}^{i}, H_{2} \right), \left( \beta_{3}^{i}, H_{3} \right), \left( \beta_{4}^{i}, H_{4} \right), \left( \beta_{5}^{i}, H_{5} \right) \right]
\]

(16)

The value of \(\beta_{n}^{i}\) is always between 0 and 1 and the sum of all degrees of belief \((\beta_{1}, \beta_{2}, \beta_{3}, \beta_{4}, \beta_{5}\) in our case) must not be larger than 1. If the sum of all degrees of belief is equal to 1 we have a complete distributed assessment, otherwise we have an incomplete assessment.

For example, the distributed assessment result of the quality of specific hardware-software component for criterion \(C_i\) with the respect to alternative \(A_j\) for our system could be:

\[
S\left(C_i \left( A_j \right) \right) = \left( \text{Excellent, 40\%} \right), \left( \text{Good, 20\%} \right), \left( \text{Average, 20\%} \right), \left( \text{Poor, 0\%} \right), \left( \text{Worst, 0\%} \right)
\]

(17)

The above can be read as follows: We have 40\% of belief degree that the quality of specific hardware-software component for our system is excellent, 20\% that the quality is good or average and 0\% of belief degree that the quality is poor or worst.
For each criterion we can define an extended decision matrix $S(C_i(A_j))$ which consists of $T$ criteria $C_i (i = 1, ..., T)$, $G$ alternatives $A_j (j = 1, ..., G)$ and $N$ evaluation grades $H_n (n = 1, ..., N)$:

$$S(C_i(A_j)) = \left( H_n, \beta_{n,i}(A_j) \right) \quad n = 1, ..., N \quad i = 1, ..., M \quad j = 1, ..., K$$  \hspace{1cm} (18)

The main goal of Evidential reasoning algorithm is to aggregate belief degrees based on the Dempster-Shafer theory. Thus, in Evidential reasoning approach it is not necessary to aggregate average scores and in this way this method is different from other MCDA methods.

Suppose we have the following two complete criterion assessments and we need to aggregate them to generate a combined assessment.

$$S(C_1(A_i)) = \left\{ \left( \beta_{1,1}, H_1 \right), \left( \beta_{2,1}, H_2 \right), \left( \beta_{3,1}, H_3 \right), \left( \beta_{4,1}, H_4 \right), \left( \beta_{5,1}, H_5 \right) \right\}$$

$$S(C_2(A_i)) = \left\{ \left( \beta_{1,2}, H_1 \right), \left( \beta_{2,2}, H_2 \right), \left( \beta_{3,2}, H_3 \right), \left( \beta_{4,2}, H_4 \right), \left( \beta_{5,2}, H_5 \right) \right\}$$  \hspace{1cm} (19)

The next step is to define a basic probability mass $m_{n,j} (j = 1, 2)$ and the remaining belief $m_{H,j} (j = 1, 2)$ for criterion $j$ (with weight $\omega_i$) unassigned to any of the $H_n (n = 1, ..., N)$ as follows (equations are obtained from [52]):

$$m_{n,1} = \omega_1 \beta_{n,1} (n = 1, ..., 5) \quad m_{H,1} = 1 - \omega_1 \sum_{n=1}^{5} \beta_{n,1} = 1 - \omega_1$$

$$m_{n,2} = \omega_2 \beta_{n,2} (n = 1, ..., 5) \quad m_{H,2} = 1 - \omega_2 \sum_{n=1}^{5} \beta_{n,2} = 1 - \omega_2$$  \hspace{1cm} (20)

To generate a combined probability masses $m_n (n = 1, ..., 5)$ and $m_H$ we need to use the Evidential Reasoning algorithm to aggregate the basic probability masses as follows (equations are obtained from [52]):

$$m_n = k \left( m_{n,1} m_{n,2} + m_{H,1} m_{n,2} + m_{n,1} m_{H,2} \right) \quad (n = 1, ..., 5)$$

$$m_H = k \left( m_{H,1} m_{H,2} \right)$$

$$k = \left( 1 - \sum_{n=1}^{5} \sum_{n=1}^{5} m_{n,1} m_{n,2} \right)^{-1}$$  \hspace{1cm} (21)

We can aggregate the combined probability masses for all other criteria at the same way. Because we have only two assessments, the combined degrees of belief $\beta_n (n = 1, ..., 5)$, as defined in [52], are generated by:

$$\beta_n = \frac{m_n}{1 - m_H} \quad (n = 1, ..., 5)$$  \hspace{1cm} (22)
The final assessment for the alternative $A_i$ can be defined as follows:

\[ S(A_i) = \{(\beta_1, H_1), (\beta_2, H_2), (\beta_3, H_3), (\beta_4, H_4), (\beta_5, H_5)\} \]  

An average score for alternative $A_i$ can be defined as follows:

\[ u(A_i) = \sum_{i=1}^{5} u(H_i) \beta_i \]  

Where $u(H_i)$ is the utility of the $i$-th evaluation grade $H_i$. By applying the equation (24) it is possible to obtain the utility interval for all decision alternatives of the problem with incomplete decision matrix and get the final ranking of alternatives.

The biggest advantage is the possibility of handling qualitative and missing data using belief matrix structure. The evidential reasoning approach is implemented in Intelligent Decision System software (IDS) which is fully functional without any limits and is available for free.

The disadvantage of the method is that it is implemented in only one tool (IDS) which has the limitation on the size of the decision problem. It cannot handle decision problems with more than 50 alternatives and 50 criteria. This method is too complex to be manually implemented.

### 3.3 Classification

Classification is the process of assigning alternatives to a set of predefined classes or categories. Depending on the method used classes could be defined using training examples or manually by the DM.

**RSA**

RSA is an acronym for Rough Sets Approach. It is a specific type of multi-criteria decision analysis method based on a classification. It was introduced by Pawlak [53]. It is used in many areas such as breast cancer classification [54], approximation of sensor signals [55], measuring information granules [56], cognitive radio networks [57], etc.

The main objective of this method is extraction of the classification rules from a set of already classified examples. Extracted rules can be used to make partition of new data sets and they are presented in the form similar to human language as “if...then...” sentences.

The following description of the rough set approach is obtained from [58]. This method concerns with an experience that is recorded in a structure called an information system. The information system may contain various types of information (e.g. events, observation, states, etc.) in terms of their criteria (e.g. variables, characteristics, symptoms, etc.). There are two groups of criteria. The first group is called “condition criteria” and it takes into account results of some measurements or data from observations. The second group of criteria is called “decision criteria” and it concerns with the expert’s decisions. The most
interesting part in the analysis of information system is the question about cause-effect dependencies between these two groups of criteria (e.g. in a form of decision rules). The criteria in information table may have different nature. Thus, we can introduce criteria with qualitative character and criteria with quantitative character. In practice, a set of criteria is represented with a mixture of quantitative and qualitative criteria. An addition problem is that the values of quantitative criteria sometimes are not interpreted as precise numbers. However, the quantitative criteria values are often translated for practical interpretation into some qualitative terms (e.g. 1 can be assigned to Low value, 2 can be assigned to Medium value and 3 can be assigned to High value).

The basic concepts of the rough sets theory are introduced as follows. The starting point of the rough sets theory is indiscernibility of objects (i.e. difficult or impossible to discern or perceive two objects) in terms of available data due to imprecise information. Thus, the precise classification of objects is not possible because of indiscernibility of objects. Two possible operations that can be used on data are lower and upper approximation of a set. These operations are defined by indiscernibility relation. Using lower and upper approximation of a classification it is possible to define accuracy and a quality of approximation. The number from interval [0,1] defines how exactly we can describe the classification of objects using available information. The approximation space is constructed by the concept of an information system.

Now the indiscernibility relation will be defined as follows. The set \( S = (U, Q, V, p) \) consisted of 4 previously defined variables presents the information system and let define relation \( P \subseteq Q \times U \). “\( x \) and \( y \) are indiscernible by the set of attributes \( P \) in \( S \) (denotation by \( x \sim_P y \)) if \( r(x, q) = r(y, q) \) for every \( q \in P, x \in U \). Equivalence classes of relation \( \sim_P \) are called \( P \)-elementary sets in \( S \). \( Q \)-elementary sets are called atoms in \( S \). The family of all equivalence classes of relation \( \sim_P \) on \( U \) is denoted by \( \mathcal{P} \) [58].

The approximations of sets will be introduced as follows. “Let \( P \subseteq Q \) and \( Y \subseteq U \). The \( P \)-lower approximation of \( Y \), denoted by \( \underline{P}Y \), and the \( P \)-upper approximation of \( Y \), denoted by \( \overline{P}Y \) are defined as:

\[
\underline{P}Y = \bigcup X \{ X \in P^*, X \subseteq Y \}
\]

\[
\overline{P}Y = \bigcup X \{ X \in P^*, X \cap Y \neq \emptyset \}
\]

25)
The $P$-boundary ($P$ doubtful region of classification) is defined as

$$B_{nP}(Y) = \overline{PY} - \underline{PY}$$

Set $\overline{PY}$ is the set of all elements of $U$ which can be certainly classified as elements of $Y$, employing the set of attributes $P$; Set $\underline{PY}$ is the set of elements of $U$ which can be possibly classified as elements of $Y$, using the set of attributes $P$. The set $B_{nP}(Y)$ is the set of elements which cannot be certainly classified to $Y$ using the set of attributes $P$.

With every subset $Y \subseteq U$ it is possible to associate an accuracy of approximation of set $Y$ by $P$ in $S$, or in short, accuracy of $Y$, defined as:

$$\mu_p(Y) = \frac{\text{card}(PY)}{\text{card}(PY)}$$

Now the rough classification will be defined. “Let $S$ be the information system, $P \subseteq Q$, and let $X = (Y_1, Y_2, \ldots, Y_n)$ be a classification of $U$, i.e. $Y_i \cap Y_j = \emptyset, i, j \leq n, i \neq j$. $Y_i$ are called classes of $X$. By $P$-lower ($P$-upper) approximation of $X$ in $S$ we mean sets $\overline{PY} = \{\overline{PY}_1, \overline{PY}_2, \ldots, \overline{PY}_n\}$ and $\underline{PY} = \{\underline{PY}_1, \underline{PY}_2, \ldots, \underline{PY}_n\}$ respectively. The coefficient:

$$\gamma_p(X) = \frac{\sum_{i=1}^{n} \text{card}(\overline{PY}_i)}{\text{card}(U)}$$

is called the quality of approximation of classification $X$ by set of attributes, or in short, quality of classification $X$. It expresses the ratio of all $P$-correctly classified objects to all objects in the system” [58].

The last step is to define the reduction of attributes. “Set of attributes $R \subseteq Q$ depends on the set of attributes $P \subseteq Q$ in $S$ (denotation $P \rightarrow R$) if $\overline{P} \subseteq \overline{R}$. Discovering dependencies between attributes enables the reduction of the set of attributes. Subset $P \subseteq Q$ in $S$ if for every $P' \subseteq P$, $\overline{P'} \subseteq \overline{P}$; otherwise subset $P \subseteq Q$ is dependent in $S$.” [58]

In practical applications, one objective is reducing those attributes which are redundant in $S$ (i.e. we are interested in obtaining so called reducts). Subset $P \subseteq Q$ is a reduct of $Q$ in $S$ if $P$ is the greatest independent set in $Q$.

To find a reduct one can also use the quality of approximation of classification $\gamma_p(X)$. The least minimal subset which ensures the same quality of classification as the set of all attributes is a reduct in $S$. It is sometimes called a minimal set of attributes. The information
Rough Sets Approach method is able to handle quantitative and qualitative data. For the method to be used a set of examples is needed to be able to extract rules from.

The disadvantage of the rough set approach is the inability to handle inconsistency. For example, let’s consider two firms, A and B, evaluated for assessment of bankruptcy risk by a set of criteria including the “debt ratio” (total debt/total assets). If firm A has a low value while firm B has a high value of the debt ratio, and evaluations of these firms on other attributes are equal, then, from bankruptcy risk point of view, firm A dominates firm B. Suppose, however, that firm A has been assigned by a DM to a class of higher risk than firm B. This is obviously inconsistent with the dominance principle. Within the original rough set approach, the two firms will be considered as just discernible and no inconsistency will be stated.

DRSA

DRSA is an acronym for Dominance-Based Rough Set Approach. It is used for classification, assignment of a set of objects to a set of pre-defined classes. Objects have a set of preference-ordered attributes called criteria. Based on criteria a set of objects is classified into pre-defined and preference-ordered classes. DRSA generalizes CRSA (Classical Rough Set Approach) by substituting the indiscernibility relation by a dominance relation. This makes it possible to discover the inconsistencies with respect to the dominance principle.

For the following method description [59] was used. DRSA partitions finite set of objects $U$ into a finite number of classes $Cl =\{Cl_t, t \in T\}$, $T =\{1,...,n\}$. The classes are preference-ordered with respect to class indices, i.e. for all $r, s \in T$, such that $r > s$, the objects from $Cl_r$ are preferred to objects from $Cl_s$. In the classification process objects are classified into a set of upward and downward unions of classes.

$$Cl^>_t = \bigcup_{s \geq t} Cl_s, \quad Cl^<_t = \bigcup_{s \leq t} Cl_s, \quad t = 1,...,n$$

For example union $Cl^>_t$ contains a set of objects belonging to a class $Cl_t$ or more preferred class. The complement of this upward union is the downward union $Cl^<_t = U - Cl^>_t$ that contains a set of objects belonging to a class $Cl_t$, or less preferred class.

DRSA defines domination using a weak preference function $\succ_q$ on a criterion $q$. The statement $x \succ_q y$ means that object $x$ is at least as good as object $y$ with respect to criterion $q$. The domination of $x$ over $y$ is denoted by $xDr_q y$ when $x \succ_q y$ for all $q \in P$ and it could be said that $x$ P-dominates $y$. Using the domination the dominating and dominated set are defined.

$$D^>_p(x) = \{y \in U : yDr_p x\}$$

is a set of objects dominating $x$, called P-dominating set.

$$D^<_p(x) = \{y \in U : xDr_p y\}$$

is a set of objects dominated by $x$, called P-dominated set.

C denotes a set of condition attributes which is a subset of $Q$, where $Q$ is a finite set of attributes. Given a set of criteria $P \subseteq C$, the set of all objects that belong to $Cl^>_t$ without any ambiguity constitutes the P-lower approximation of $Cl^>_t$, denoted by $\overline{P}(Cl^>_t)$, while the P-upper approximation of $Cl^>_t$, denoted by $\overline{P}(Cl^>_t)$, is constituted by the set of all objects that could belong to $Cl^>_t$. All the objects that belong to $Cl^>_t$ and $Cl^<_t$ unions of classes with some ambiguity constitute the P-boundary of $Cl^>_t$, denoted by $Bn_p(Cl^>_t)$, and the P-boundary of $Cl^<_t$, denoted by $Bn_p(Cl^>_t)$. They can be defined by upper and lower approximations.

$$Bn_p(Cl^>_t) = \overline{P}(Cl^>_t) - \overline{P}(Cl^>_t), \quad Bn_p(Cl^<_t) = \overline{P}(Cl^<_t) - \overline{P}(Cl^<_t), \quad \text{for } t=1,...,n.$$
P-lower approximations of unions of classes represent certain knowledge, P-upper approximations represent possible knowledge and the P-boundaries represent doubtful knowledge provided by criteria from $P \subseteq C$.

The quality of approximation of the multicriteria classification $Cl$ by set of criteria $P$ is defined for every $P \subseteq C$. Values are calculated as a ratio between the number of P-correctly classified objects (objects that do not belong to any P-boundary of unions $Cl_i^\leq$ and $Cl_i^\geq$, where $t=1,...,n$) and the number of all objects in the data sample set.

$$\gamma_p(Cl) = \frac{\text{card} \left( U - \bigcup_{i \in T} \text{Bn}_p(Cl_i^\geq) \right) }{\text{card}(U)} = \frac{\text{card} \left( U - \bigcup_{i \in T} \text{Bn}_p(Cl_i^\leq) \right) }{\text{card}(U)}$$

(31)

For $P$ set of criteria and $Cl$ classification, $\gamma_p(Cl)$ represents the measure of the quality of the knowledge extracted from the data matrix.

The reduct is a minimal subset $P \subseteq C$ such that $\gamma'_p(Cl) = \gamma'_c(Cl)$ . There could be more than one reduct for a data sample, and in this case the intersection of all reducts is called the core and is denoted by $\text{CORE}_{Cl}$. The core contains the data that could not be removed from the data set without hampering the quality of the extracted knowledge. Three categories of criteria exist:

1. indispensable criteria that is included in the core
2. exchangeable criteria that is included in at least two reducts but not in the core
3. redundant criteria that do not belong to neither indispensable nor exchangeable criterion category, so it not included in any reduct

DRSA rules that are extracted from the data matrix are in the form of if...then... rules. There are five possible types of decision rules. There are certain and possible rules for classification using upper ($D^\geq$) and certain and possible rules for classification using lower unions of classes ($D^\leq$). The last type of rules is the approximate decision rule ($D^\leq\geq$). Certain rules represent certain knowledge, possible rules represent possible knowledge and approximate rules represent doubtful knowledge.

In the classification process, the extracted rules are applied on the object that is being classified. When using $D^\leq$-decision rules, if the object matches one rule then the object belongs at least to the class $Cl_i$ or better. If it matches more than one rule, then it belongs to the highest upward union of classes. If object does not match any rule it is assigned to the lowest class, since there is no evidence against it. When using $D^\geq$-decision rules, if the object matches one rule then the object belongs to the class $Cl_i$ or worse. If it matches more than one rule, then it belongs to the lowest downward union of classes. When applying $D^\leq\geq$-decision rules, the object belongs to the union of classes defined by all matching rules.

DRSA method handles quantitative and qualitative data and deals with incomplete and inconsistent data. For the method to be used a set of examples is needed in order to be able to extract rules from. The method is not limited to a specific field and could be used for a wide variety of real-life problems.
3.4 Ranking and Classification

In this subchapter Electre and SMAA methods are presented. Both MCDA method families have methods for ranking and classification.

Electre III

Electre is a family of methods used in MCDA. The first Electre method, Electre I method, was created in 1965 by Bernard Roy. More Electre methods had been created since then, to adapt the method to various real-life problems. Electre is an acronym that stands for ELimination Et Choix Traduisant la REalité which in English mean ELimination and Choice Expressing REality. Electre III method was created in 1978 and is designed for ranking problems.

When the relative importance of criteria is available and desirable Electre III method should be used. Electre IV, which is described in the following subchapter, is the only Electre method that does not use criteria importance coefficients and it is used when the DM is not able, does not want or cannot express the relative importance of each criterion [60]. But for it to be valid no criterion can be superior in weight or negligible when compared to any subset of half of the criteria.

Electre methods consist of two phases. In the first phase outranking relations are constructed. Electre III method constructs a fuzzy outranking relations and Electre IV set of nested sequence of outranking relations. In the second phase, the exploitation procedure for ranking a final partial pre-order that combines two complete preorders is carried out [61].

Two important concepts in Electre methods are outranking and thresholds.

Preference is modeled by using outranking relation $S$ which means $a$ is at least as good as $b$ or $a$ is not worse than $b$. There are four possible situations that may occur [62]:

- $aSb$ and not$(bSa)$ – $aPb$ – $a$ is strictly preferred to $b$
- $bSa$ and not$(aSb)$ – $bPa$ – $b$ is strictly preferred to $a$
- $aSb$ and $bSa$ – $aIb$ – $a$ is indifferent to $b$
- not$(aSb)$ and not$(bSa)$ – $aRb$ – $a$ is incomparable to $b$

Electre methods use concordance/discordance principles for creating outranking relations.

- Concordance: for $aSb$ to be true, a sufficient majority of criteria have to be in favor of the assertion.
- Non-discordance: for the concordance condition to hold, none of the minority should oppose too strongly to the assertion $aSb$.

Assertion $aSb$ is validated when both conditions are fulfilled [60].

Electre III method introduced three thresholds: indifference, preference and veto threshold.

The preference threshold $p_i$ is the difference at which the DM strongly prefers one alternative over another for the criterion $i$. If $g_i(b) > g_i(a) + p_i(g_i(a))$, where $g_i(a)$ is a value of criterion $i$ for the alternative $a$, then alternative $b$ is strictly preferred to alternative $a$ in terms of criterion $i$ for increasing preference criteria [62].

The indifference threshold $q_i$ is a difference beneath which the DM is indifferent and/or is not able to recognize the difference between alternatives with regards to $j$ criterion. If
g_i(b) > g_i(a) + q_i(g_i(a)) then alternative a is weakly preferred to alternative b in terms of criterion i for increasing preference criteria [62].

The veto threshold prevents one alternative outranking the other and is optional in Electre III method. If the value of b is greater than the value of a by the amount greater than the veto threshold v_i, i.e. if g_i(b) ≥ g_i(a) + v_i(g_i(a)) for increasing preference criteria alternative a cannot outrank the alternative b [62].

Thresholds may have fixed values or they may be defined as a function of criteria values. Electre III/IV software uses the formula \( \alpha \times g_i(a) + \beta \) for thresholds definition, where \( \alpha \) and \( \beta \) are coefficients set by the DM [61].

The first step in Electre III method is defining the decision matrix and thresholds.

The second step is to calculate partial concordance index for each criterion. For the pair of alternatives (a, b) it measures whether the alternative a is at least as good as action b in criterion \( g_i \) [62].

\[
C_i(a, b) = \begin{cases} 
0, & \text{if } g_i(b) \geq g_i(a) + p_i(g_i(a)) \\
1, & \text{if } g_i(b) \leq g_i(a) + q_i(g_i(a)) \\
\text{otherwise}, & \frac{g_i(a) + p_i(g_i(a)) - g_i(b)}{p_i(g_i(a)) - q_i(g_i(a))}
\end{cases}
\]

Next, the overall concordance index (sometimes called comprehensive concordance index) is calculated using the following formula [62]:

\[
C(a, b) = \frac{\sum w_i C_i(a, b)}{\sum w_i}
\]

The discordance index for each criterion is calculated in the fourth step. It takes into account the fact that the criterion is more or less discordant with the assertion a outranks b. If the veto threshold \( v_i \) is not defined then \( D_i(a, b) = 0 \) for all pairs of alternatives [62].

\[
D_i(a, b) = \begin{cases} 
0, & \text{if } g_i(b) \leq g_i(a) + p_i(g_i(a)) \\
1, & \text{if } g_i(b) \geq g_i(a) + v_i(g_i(a)) \\
\text{otherwise}, & \frac{g_i(b) - g_i(a) - p_i(g_i(a))}{v_i(g_i(a)) - p_i(g_i(a))}
\end{cases}
\]

In the fifth step the fuzzy outranking relation is calculated for each pair of actions (a,b) as a credibility index. It expresses the probability that a outranks b taking into consideration both the overall concordance index and the discordance indexes for each criterion. It prevents the compensation of big loss on one criterion by a number of small gains on other criteria [62].

\[
S(a, b) = \begin{cases} 
C(a, b), & \text{if } D_i(a, b) \leq C(a, b) \forall i \\
C(a, b) \prod_{p_i(a \neq b) > C(a, b)} \frac{1 - D_i(a, b)}{1 - C(a, b)}, & \text{otherwise}
\end{cases}
\]
When there is no discordant criterion, the credibility index is the same as the overall concordance index. When some discordant criterion activates its veto power, when \( D_i(a,b) = 1 \) for some criterion, the credibility index is null, indicating that the assertion is not credible. When the overall concordance index is strictly lower than the discordance index on the discordant criterion, the credibility index becomes lower than the overall concordance index because of the opposition effect on this criterion [60].

The sixth step is the rank ordering [62].

- from the credibility matrix, the credibility index with the highest value is denoted by \( \lambda_{\text{max}} \) and calculated by \( \lambda_{\text{max}} = \max S(a,b) \)
- the cutoff value \( \lambda \) is calculated by \( \lambda = \lambda_{\text{max}} - s(\lambda) \), where \( s(\lambda) \) is a discrimination threshold calculated using the formula \( s(\lambda) = \alpha \lambda_{\text{max}} + \beta \). Coefficients \( \alpha \) and \( \beta \) are called distillation coefficients and have suggested values of -0.15 and 0.3 respectively.
- \( \lambda \)-strength is determined as the number of alternatives \( b \) with \( S(a,b) > \lambda \) for each alternative \( a \)
- \( \lambda \)-weakness is determined as the number of alternatives \( b \) with \( (1 - (0.3 - 0.15\lambda)) * S(a,b) > S(b,a) \) for each alternative \( a \)
- for each alternative its qualification is determined as the difference between \( \lambda \)-strength and \( \lambda \)-weakness
  - descending distillation is determined, and a set of alternatives with the largest qualification is called the first distillate (D1)
  - ascending distillation is determined, and a set of alternatives with the lowest qualification is called the first distillate (D1)
- if there are more than one D1 alternatives, the process is repeated on the D1 set until all alternatives have been classified.
- then the process is repeated with the original set minus the D1 set, until all alternatives have been classified
- there are several ways to do the final ranking, but the most common approach is the intersection of two outranking relations

Electre III method is a well-known and widely used method in many different fields i.e. urban development [63], project management [64] etc. There are a few existing tools that support the method, which has certainly helped in its wider application. It handles quantitative and due to thresholds inconsistent data. It does not support qualitative or missing data.

**Electre IV**

Electre IV method was created in 1982 as a result of a Paris subway network problem. It enabled the ranking of alternatives without using criteria weights. However, for it to be valid no criterion can be superior in weight or negligible when compared to any subset of half of the criteria [60] [61].

Since Electre IV does not use weight of criteria it is impossible to create the overall concordance index matrix. Electre IV method introduces five outranking relations (\( S_q, S_o, S_p, S_s \) and \( S_v \)) that must fulfil two constraints:

- "no criterion is preponderant when compared to any subset of half of the criteria
• no criterion is negligible when compared to any subset of half of the criteria” [61]

The relations are:

• Quasi-dominance relation, \( S_{q} \)
  
  o “The couple (b, a) verifies the relation of quasidominance if and only if:
    
    ▪ for every criterion, b is either preferred or indifferent to a,
    
    ▪ and if the number of criteria for which the performance of a is better than the one of b (a staying indifferent to b) is strictly inferior to the number of criteria for which the performance of b is better than the one of a” [62]

• Canonic dominance relation, \( S_{c} \)
  
  o “The couple (b, a) verifies the relation of canonic-dominance if and only if:
    
    ▪ for no criterion, a is strictly preferred to b,
    
    ▪ and if the number of criteria for which a is weakly preferred to b is inferior or equal to the number of criteria for which b is strictly preferred to a,
    
    ▪ and if the number of criteria for which the performance of a is better than the one of b is strictly inferior to the number of criteria for which the performance of b is better than the one of a.” [62]

• Pseudo-dominance relation, \( S_{p} \)
  
  o “The couple (b, a) verifies the relation of pseudo-dominance if and only if:
    
    ▪ for no criterion, a is strictly preferred to b,
    
    ▪ and if the number of criteria for which a is weakly preferred to b is inferior or equal to the number of criteria for which b is strictly or weakly preferred to a.” [62]

• Sub-dominance relation, \( S_{s} \)
  
  o “The couple (b, a) verifies the relation of subdominance if and only if:
    
    ▪ for no criterion, a is strictly preferred to b.” [62]

• Veto-dominance relation, \( S_{v} \)
  
  o “The couple (b, a) verifies the relation of vetodominance if and only if:
    
    ▪ either for no criterion, a is strictly preferred to b,
    
    ▪ or a is strictly preferred to b for only one criterion but this criterion not vetoing the outranking of a by b and furthermore, b is strictly preferred to a for at least half of the criteria.” [62]

For creating fuzzy outranking relations in Electre IV, the following calculations have to be done, i.e. for pair (a,b):

- \( n_{p}(a,b) \) - the number of criteria for which a is strictly preferred to b, \( aP_{b} \)
- \( n_{q}(a,b) \) - the number of criteria for which a is weakly preferred to b, \( aQ_{b} \)
\( n(a,b) \) - the number of criteria for which \( a \) is considered indifferent to \( b \), \( a I b \), but such that \( a \) has a better performance than \( b \)

\( n_0(a,b) = n_0(b,a) \) - the number of criteria for which \( a \) and \( b \) have the same performance \( g(a) = g(b) \).

\( n(b,a) \) - the number of criteria for which \( b \) is considered indifferent to \( a \), \( b I a \), but such that \( b \) has a better performance than \( a \)

\( n_q(b,a) \) - the number of criteria for which \( b \) is weakly preferred to \( a \), \( bQa \)

\( n_p(b,a) \) - the number of criteria for which \( b \) is strictly preferred to \( a \), \( bP a \)

Then using the calculated values the credibility degrees (i.e. \( \sigma(a,b) \) for the \((a,b)\) pair) can be calculated for the construction of the credibility matrix by [61]:

\[
\sigma(a,b) = \begin{cases} 
1 & \text{if } a S b \ (\text{quasi-dominance}) \\
0.8 & \text{if } a S b \ (\text{canonic-dominance}) \\
0.6 & \text{if } a S b \ (\text{pseudo-dominance}) \\
0.4 & \text{if } a S b \ (\text{sub-dominance}) \\
0.2 & \text{if } a S b \ (\text{veto-dominance}) \\
0 & \text{if no relation among } \{S_q, S_c, S_p, S_s, S_v\} \text{ for } (a,b) 
\end{cases}
\]

The distillation threshold function is defined as \( s(\lambda) = \alpha \times \lambda + \beta \), where \( \alpha \) and \( \beta \) constants are in the Electre III/IV software set to \( \alpha = 0 \) and \( \beta = 0.1 \). This value is used for ranking, transforming nested relations into a fuzzy relation. “In the first step of ranking, only the strongest dominance among those that have been verified will be taken into account. In the second step of ranking, the two strongest dominance will be taken into account etc” [61].

Electre IV was a response to the real-life problem, but overall it is not as popular as Electre III method. It enabled ranking of alternatives without using criteria weights. It handles quantitative and inconsistent data. There is the Electre IV tool that implements the method, and it is described later in the thesis. It does not support qualitative or missing data.

**ELECTRE TRI**

Electre TRI is the only method from Electre family of MCDA methods that supports classification. It is not field-specific and so far it has been successfully applied in different fields like human resources [65] and in IT for evaluation of software trustworthiness [66].

The method description is obtained from [67].

Electre TRI is used for assigning a set of alternatives \( A \) to pre-defined categories \( C \). The categories are ordered from the worst \((C_1)\) to the best \((C_k)\) and are defined by profiles that specify lower and upper limits of categories. The upper limit of category \( C_h \) is the lower limit of the category \( C_{h+1} \) and is denoted by \( b_h \).

In what follows, it is assumed, without any loss of generality, that preferences increase with the value of each criterion.

The assignment of the alternatives to the categories is done in two steps. In the first step the outranking relation \( S \) is constructed. It characterizes how alternatives compare to the limits of categories, i.e. if they are better or worse than profile values. In the second step the \( S \) relation is exploited in order to assign each alternative to a specific category.
In the first step the credibility index is calculated as in Electre III method [60]. Electre TRI builds outranking relation \( S \) using an index \( \sigma(a,b_h) \in [0,1] \) (and index \( \sigma(b_h,a) \) respectively) that represent the degree of credibility of the assertion \( a \triangleright S b_h \) (respectively \( b_h \triangleright S a \)), \( \forall a \in A, \forall h \in B \), where \( B \) represents the set of profile indices and \( A \) set of alternatives. As in Electre III and IV methods indifference and preference thresholds \( (q(b_h) \) and \( p(b_h)) \) are used to account for imprecise data. Veto threshold \( v_f(b_h) \) is optional and is used for calculating the discordance index.

In the second step \( \lambda \)-cutting level of the fuzzy relation is used to obtain crisp outranking relations from \( S \) outranking relation. It could be defined as the smallest value of the credibility index compatible with the assertion \( a \triangleright S b_h \). \( \lambda \)-cutting level determines the preference situation between \( a \) and \( b_h \):

- “\( \sigma(a,b_h) \geq \lambda \) and \( \sigma(b_h,a) \geq \lambda \) \( \Rightarrow a \triangleright S b_h \) and \( b_h \triangleright S a \), i.e., \( a \) is indifferent to \( b_h \)
- “\( \sigma(a,b_h) \geq \lambda \) and \( \sigma(b_h,a) < \lambda \) \( \Rightarrow a \triangleright S b_h \) and not \( b_h \triangleright S a \), i.e., \( a \) is preferred to \( b_h \) (weakly or strongly)
- “\( \sigma(a,b_h) < \lambda \) and \( \sigma(b_h,a) \geq \lambda \) \( \Rightarrow \) not \( a \triangleright S b_h \) and \( b_h \triangleright S a \), i.e., \( b_h \) is preferred to \( a \) (weakly or strongly)
- “\( \sigma(a,b_h) < \lambda \) and \( \sigma(b_h,a) < \lambda \) \( \Rightarrow \) not \( a \triangleright S b_h \) and not \( b_h \triangleright S a \), i.e., \( a \) is incomparable to \( b_h \)” [67]

The alternatives are compared to profiles to determine to which category alternative should be assigned. There are two possible assignment procedures:

- “Pessimistic (or conjunctive) procedure:
  - compare \( a \) successively to \( b_h \) for \( i=p, p-1, \ldots, 0 \);
  - \( b_h \) being the first profile such that \( a \triangleright S b_h \), assign \( a \) to category \( C_{h+1}(a \rightarrow C_{h+1}) \)

- Optimistic (or disjunctive) procedure:
  - compare \( a \) successively to \( b_h \) \( i=1, 2, \ldots, p+1; \)
  - \( b_h \) being the first profile such that \( b_h \triangleright S a \), assign \( a \) to category \( C_h(a \rightarrow C_h) \) [67]

With pessimistic rule for \( \lambda=1 \) the alternative \( a \) would have to have all criteria values \( g_j(a) \) equal (up to a threshold) or greater than the corresponding category boundaries \( g_j(b_h) \). With the optimistic rule the alternative would be assigned to the category \( C_h \) for which at least one of the lower profile category boundaries is (by a threshold value) greater than the corresponding criteria value \( g_j(a) \). “When \( \lambda \) decreases, the conjunctive and disjunctive characters of these rules are weakened” [67].

Electre TRI is used for classification of alternatives to predefined classes. It handles quantitative and due to thresholds inconsistent data. Electre TRI is a tool that implements the method and it is described later in the thesis. The method does not support qualitative or missing data.

**SMAA-2**

The SMAA acronym stands for Stochastic Multicriteria Acceptability Analysis. SMAA is a family of MCDA methods that are based on inverse parameter space analysis. This requires the calculation of multidimensional integrals that could be calculated using Monte Carlo method simulation. “The main results of the methods are rank acceptability indices, central weight vectors and confidence factors for different alternatives.”[68] The values of rank acceptability indices take into consideration criteria values and weights and are used for ranking the alternatives. The central weight vectors represent the preference of alternatives on different criteria. The confidence factors are used to assess if the criteria measurements are sufficiently accurate to make an informed decision.
SMAA methods have been developed with the goal of handling uncertain or inaccurate criteria values and when it is not possible to get the accurate criteria weights or there might be no weights available. The SMAA methods could not be calculated manually because of the use of the simulation. JSMAA is open-source multiplatform software that implements SMAA methods.

The SMAA-2 method is used for ranking a set of $m$ alternatives in term of $n$ criteria with multiple DMs. It uses value function $u(x_i,w)$ for the DMs to evaluate the alternatives. “Uncertain or imprecise criteria are represented by stochastic variables $\xi_{ij}$ with joint density function $f_{X}(\xi)$ in the space $X \subseteq \mathbb{R}^{m \times n}$[68]. The joint density function $f_{W}(w)$ in the weight space $W$ represents the DM’s unknown or partially known preference. When the weight is missing the uniform weight distribution is used.

The ranking function $rank(i,\xi,w) = 1 + \sum_{k=1}^{m} \rho(u(\xi_k,w) > u(\xi_i,w))$, where $\rho(true) = 1$ and $\rho(false) = 0$, is used to define the rank for each alternative. The set of favourable rank weights is calculated as $W'_i(\xi) = \{w \in W : rank(i,\xi,w) = r\}$.

The multidimensional integral over the criteria distributions and the favourable rank weights is used to calculate the rank acceptability index as $b'_i = \int_{\xi \subseteq X} f_{X}(\xi) \int_{w \in W'_i(\xi)} f_{W}(w)dwd\xi$. “The most acceptable (best) alternatives are those with high acceptabilities for the best ranks” [68].

The central weight vector $w'_i$ represents the centre of gravity of the favourable weight vectors for alternatives. It is calculated by a multidimensional integral over the criteria distributions and the favourable first rank weights as $w'_i = \int_{\xi \subseteq X} f_{X}(\xi) \int_{w \in W'_1(\xi)} f_{W}(w)dwd\xi / a_i$. It represents the DM’s preferences.

The confidence factor $p'_i$ represents the probability for an alternative to be the best one when the central weight vector is chosen. “The confidence factor is computed as a multidimensional integral over the criteria distributions using $p'_i = \int_{\xi \subseteq X \cdot \text{rank}(i,\xi,w'_i) = r} f_{X}(\xi)d\xi$ ” [68]. The low values of the confidence factors indicate that there is not enough information available for an informed decision.

SMAA-2 method has an extension that enables handling of ordinal criteria preference. The extension is called SMAA-O method. For the ordinal preference values the corresponding cardinal preference values are calculated.

SMAA-2 handles quantitative values, expressed as discrete values, ranges, functions or Gaussian distributions. Weights could be cardinal, ordinal (for SMAA-O) or missing. Disadvantage of the method is that it cannot handle missing values, qualitative values and the results are less traceable that of the other methods due to multidimensional integral calculation.
SMAA-TRI is an extension of Electre TRI classification MCDA method. It makes it possible to handle uncertain values like ranges and Gaussian distributions. The method description is obtained from [69].

Just like Electre TRI the input values are cutting level \( \Lambda \) with a valid range of values \([0.5, 1]\), weights \( w \) (if missing, uniform distribution is used), and set \( T = \{ M, B, q, p, v \} \) where \( M \) is criteria evaluation matrix, \( B \) a set of profiles, \( q \) indifference threshold, \( p \) preference threshold and \( v \) veto threshold.

SMAA-TRI calculates category acceptability indices \( \pi^h_i \) for all pairs of alternatives and categories. Category acceptability index \( \pi^h_i \) is a generalization of rank acceptability index of SMAA-2. “\( \pi^h_i \) describes the share of possible parameter values that have an action \( a \), assigned to category \( C_h \)” [69]. “A categorisation function that evaluates the category index \( h \) to which an alternative \( x_i \) is assigned by Electre TRI

\[
h = K(i, \Lambda, w, T)
\]

“[69] as well as the category membership function

\[
m^h_i(\lambda, w, T) = \begin{cases} 1, & \text{if } K(i, \Lambda, w, T) \\ 0, & \text{otherwise} \end{cases}
\]

have to be defined to be able to calculate the category acceptability index. It is obtained as a result of the numerical calculation of a multidimensional integral over the finite parameter space as:

\[
\pi^h_i = \int_{\lambda \in \lambda} \int_{w \in W} f_L(\lambda) f_w(w) m^h_i(\lambda, w, T) dw d\lambda
\]

It measures the stability of the assignment and could be interpreted as a fuzzy measure or a probability for membership in the category, so its values are within the range \([0, 1]\). Results of the SMAA-TRI method are category acceptability indices for all pair of actions and categories.

SMAA-TRI handles quantitative values, expressed as discrete values, ranges, functions or Gaussian distributions. Weights could be cardinal, ordinal or even missing. Drawbacks of the method are that it cannot handle missing values, qualitative values and the results are less traceable that for the other methods due to multidimensional integral calculation.
Chapter 4

MCDA TOOLS

Today there are a lot of different methods available which are able to choose the best alternative, the best subset of alternatives, rank alternatives or classify/sort them into predefined preference classes. For a lot of these methods there are software tools available to help the DM with the decision making.

We studied a number of different tools in order to find the one that satisfies as many requirements (defined in chapter 5.2) as possible. Numerous tools have been proposed for dealing with complex problems in decision making. Specifically, in this thesis, we have evaluated MCDA tool for ranking and classification.

The MCDA tools used for ranking in this thesis are Electre III/IV, MakeItRational, ANP Solver, Web ANP Solver, TransparentChoice, Triptych, SANA, Visual PROMETHEE Academic and Intelligent Decision System. The ranking can be defined as a process of putting a set of alternatives in order from the most desirable to the least desirable based on values of multiple criteria.

The MCDA tools used for classification in this thesis are ELECTRE TRI, 4eMka2, jMAF and ROSE2. The classification can be defined as a process of assigning alternatives to a set of predefined classes or categories. The classes could be defined using already classified examples or manually by the DM. The tool JSMAA is used both for ranking and classification.

Most of the tools implement only one method, but there are some tools that implement more than one method. Unfortunately, some tools do not implement the complete method(s), but only a simplified version where some steps are omitted or simplified. Not surprisingly, the proposed tools may result in a different indication of classification or ranking of all alternatives.
4.1 Ranking

Electre III/IV

Electre III/IV tool, as its name suggests, implements both Electre III and IV ranking methods, as described in chapters Electre III and Electre IV.

The tool enables ranking a set of alternatives according to multiple criteria, using Electre III method when criteria weights are known and Electre IV method when they are not available.

The Electre III/IV tool has been developed by the Dauphine University of Paris and Institute of Computer Science of the Poznan University of Technology. It is developed by using Borland C++ programming language. It runs on Windows 3.1 or newer operating systems [70].

The user starts by creating a new project, and selecting the method to be used. It is possible to use existing or create new data set. The data set is a file that contains all the data needed for ranking i.e. alternatives, criteria, criteria values etc. Since Electre III and IV methods have different parameters, the data set is method-specific. If the new data set is created, criteria and alternatives must be defined. All criteria values for all alternatives have to be specified together with threshold values. Indifference and preference thresholds are mandatory and veto threshold is optional. Threshold values can be specified as constant or as a function. Criteria weights have to be defined for Electre III method while the dominance relations have to be defined for Electre IV method. Distillation coefficient values could be modified if it is necessary.

The tool handles quantitative data, enables traceability by providing values of concordance, credibility and ranking matrices. Ascending and descending distillation results are provided in the graph form shown on Figure 6 together with the final ranking results.

![Electre III/IV tool](image)

Figure 6. Electre III/IV, distillation results and final ranking

The tool is limited to 64 criteria and does not support missing values.
MakeItRational is a decision-making tool that implements the Analytic Hierarchy Process (AHP), which is described in chapter 3.2. The main objective of the tool is ranking a finite set of alternatives in terms of a finite number of decision criteria. The best ranked alternative is found as the most suitable one for a DM.

The MakeItRational tool is available in both online and desktop versions. In the online version it is delivered as a software as a service (SaaS) solution so the user needs to install any web browser together with Microsoft Silverlight 4.0 to run it [71]. Requirements for a desktop version are Windows or Macintosh OS with Silverlight 4.0 installed. Since all projects are stored on server it is also necessary to establish the Internet connection. Online and desktop versions are commercial, but it is possible to sign up for trial version and get 30 days of full access to online version with no obligation [72]. After that the monthly price will be offered to prolong usage. Free trial option is not available for a desktop version.

The MakeItRational tool handles quantitative values, and provides results in form of numeric values and graph visualization. The input values are criteria, criteria weights (resulting from a 9-point scale, as shown in table 3), alternatives, alternative evaluations with respect to the criteria (resulting from a 9-point scale, as shown in table 3). The output is automatically generated in the form of graph visualization.

To describe the features of MakeItRational tool we will use a simple example of choosing appropriate leader for our organization. Our example consists of 3 alternatives (Chris, Eduardo, Ray) and of 4 criteria (Age, Charisma, Education, Experience). The whole procedure is defined with five steps. In the first step it is necessary to define list of all alternatives (i.e. define list of items that will be evaluated). Depending on a decision it will be a list of suppliers, projects, locations, etc.

![Figure 7. Defining alternatives](image-url)
In the second step the criteria of decision problem have to be defined (i.e. what is our goal? What criteria will be taken into account during evaluation?).

Once when criteria are defined, the next step is to prioritize them (i.e. compare criteria in pairs, each time choose the more important one and select the level of dominance on a 9-point scale, as shown in table 3)
In the fourth step we need to score alternatives (i.e. define rating scales that model our preferences and use them to score alternatives in context of criteria).

![Image of pairwise comparisons in context of Age](image1)

**Figure 10. Scoring alternatives**

In the last step we get the final ranking of alternatives. On the Figure 11 we can see that the most suitable team leader is Chris.

![Image of alternatives ranking](image2)

**Figure 11. Final ranking of alternatives**

To run the online version of tool, the user needs to install any web browser together with Microsoft Silverlight 4.0. Thus, the independence of platform is an advantage of using this tool.
On the other side, this tool is commercial and is free only for a period of 30 days. The limited number of alternatives and criteria are not specified. Only Windows and Macintosh OS are supported for a desktop version of tool.

**ANP Solver**

ANP Solver is a tool implementing the ANP method (as described in 3.2) which is free for academic use. The main objective of ANP Solver is to provide a reliable and user friendly tool able to obtain the ranking of a finite set of alternatives in terms of a finite number of decision criteria. The best ranked alternative is found as the most suitable one for a DM.

This tool was developed as a part of the dissertation thesis of Elena Rokou with the title “Development of software tool to support the analytical network process (ANP) for the evaluation of project portfolio”. It runs only on Windows platform [73].

[73]The ANP Solver tool handles quantitative values, and provides results in a form of numeric values in limited matrix, as described in chapter 3.2. The input values are clusters, clusters relationships, criteria, criteria weights (resulting from a 9-point scale, as shown in table 3), alternatives and alternatives evaluation with the respect to criteria (resulting from a 9-point scale, as shown in table 3). The output is automatically generated in the form of matrix.

The user starts by creating a new project. It is possible to use existing or create new data set. It is a file that contains all the data needed for ranking i.e. clusters, alternatives, criteria, criteria values, relationships between clusters, etc. If the new data set is created, clusters, clusters relationships, criteria, criteria prioritization, alternatives and alternatives scores must be defined. We will use a simple example of choosing appropriate leader for our organization. Our example consists of 3 alternatives (Chris, Eduardo, Ray) and of 4 criteria (Age, Charisma, Education, Experience). In the first step it is necessary to define two clusters (i.e. cluster for criteria and cluster for alternatives). The second step is dealing with adding 4 nodes of criteria to criteria cluster and 3 nodes of alternatives to alternatives cluster. Once when criteria are defined, the next step is to prioritize them (i.e. compare criteria in pairs, each time choose the more important one and select the level of dominance on a 9-point scale, as shown in table 3). In the fourth step we need to score alternatives (i.e. define rating scales that model our preferences and use them to score alternatives in context of criteria). The final step represents creating four matrices (i.e. cluster, super, weighted super and limit matrix) based on the ANP algorithm. The final ranking is obtained by looking at the first column of limit matrix, as shown on the Figure 12.
This tool is non-commercial and completely free for academic purposes. It has user friendly interface and is very easy to use.

On the other side, this tool is only supported by Windows platform. The maximum number of alternatives and criteria is not specified. For future development it would be nice to introduce graphical presentation of final results.

Web ANP Solver

Web ANP Solver is a multicriteria decision analyze web application that implements the Analytic Network Process (ANP) ranking MCDA method, as described in chapter 3.2. The main objective of Web ANP Solver is to provide a reliable and user friendly web application able to obtain the ranking of a finite set of alternatives in terms of a finite number of decision criteria. The best ranked alternative is found as the most suitable one for a DM.

To run this application the user needs a computer connected to the Internet having a web browser [74]. Every user has its personal account with disk space allocated on a server in order to create/open projects.

Since this web application represents an “online” version of the ANP Solver tool, which is described previously, all features, advantages and disadvantages are the same.
TransparentChoice

TransparentChoice is a MCDA tool that implements the AHP method, as described in chapter 3.2. The main objective of the tool is obtaining the ranking of alternatives from a finite set of alternatives with the respect to criteria. The most suitable alternative for a DM is always the best ranked.

Since TransparentChoice is delivered as a software as a service (SaaS) solution, the user needs to install any web browser together with flash plugin to run it [75]. All projects are stored on server so the Internet connection is needed to use it. This tool is commercial, but it is possible to sign up for trial version and get 30 days of full access to tool with no obligation. After that the monthly price will be offered to prolong usage [76].

The TransparentChoice tool handles quantitative values, and provides results in form of numeric values and graph visualization. The input values are criteria, criteria weights (resulting from a 9-point scale, as shown in table 3), alternatives, alternatives evaluations with the respect to criteria (resulting from a 9-point scale, as shown in table 3). The output is automatically generated in the form of graph visualization.
To describe the features of TransparentChoice tool we will use a simple example of choosing appropriate leader for our organization. Our example is consisted of 3 alternatives (Chris, Eduardo, Ray) and of 4 criteria (Age, Charisma, Education, Experience). The whole procedure is defined with five steps. In the first step it is necessary to define list of all alternatives (i.e. define list of items that will be evaluated). Depending on a decision it will be a list of suppliers, projects, locations, etc.

![Figure 14. Defining alternatives](image1)

In the second step the criteria of decision problem have to be defined (i.e. what is our goal? What criteria will be taken into account during evaluation?).

![Figure 15. Defining criteria](image2)
Once when all criteria are defined, the next step is to prioritize them (i.e. compare criteria in pairs, each time choose the more important one and select the level of dominance on a 9-point scale, as shown in table 3).

In the fourth step we need to score alternatives (i.e. define rating scales that model our preferences and use them to score alternatives in context of criteria).
In the last step we get the final ranking of alternatives. On the Figure 18 we can see that the most suitable team leader is Chris.

![Figure 18. Final ranking of alternatives](image)

In order to run this tool, the user needs to install any web browser together with flash plugin. Thus, the independence of platform is advantage of using this tool.

On the other side, this tool is commercial and is free only for a period of 30 days. The limited number of alternatives and criteria are not specified.

**Triptych**

Triptych is an Excel plugin made by the company Statistical Design Institute. It consists of a number of tools and one of them implements the TOPSIS method.

Triptych is one of two company’s products for System Engineering and Design for Six Sigma. Triptych provides a set of tools for documenting and clarifying the requirements, generating ideas and selecting design alternatives. Besides the full implementation of TOPSIS method, there is also a tool that implements part of the AHP method, used for calculating relative importance of items after performing pair-wise comparisons of items against each other in term of relative importance.

It requires Windows XP, Vista or 7 operating system, and Microsoft Excel version 2003, 2007 or 2010.

The TOPSIS tool handles quantitative values, and provides results in form of numeric values and graph visualization.

The input values are criteria, criteria weights, criteria preferences (minimization or maximization) and alternatives with their criteria values. The output is automatically generated in the form of closeness coefficient values and its graph visualization. The separations from positive and negative ideal solution are also provided.

The TOPSIS tool is limited to 200 criteria and 200 alternatives and does not support missing values.

Figure 19 shows the example with three alternatives and four criteria.
SANNA

SANNA is an Excel plugin made by Josef Jablonsky and Pavel Urban from Department of Economics of University of Economics Prague. Its name is abbreviation of System for ANalysis of Alternatives. It implements a number of MCDA methods of which Electre III, PROMETHEE II and TOPSIS are methods that are of interest in this thesis.

SANNA is written in Visual Basic for Application (VBA) and as already mentioned it is Microsoft Excel plugin. Its goal is to be powerful and user friendly MCDA tool, to help the wider utilization of MCDA methods on real world problems [78], [79].

It requires Windows operating system, and Microsoft Excel. The version of Excel for which the plugin was created is not specified.

The SANNA tool enables MCDA method selection, handles quantitative values, and provides results as numeric values and enables result traceability.

The project is started by creating a new Excel spreadsheet. After that a new data set is created. After filling in all the criteria, criteria weights, alternatives, its values and criteria preference (cost or gain) it is possible to check dominance and remove dominated alternatives. Next step in the process is to choose the MCDA method. In this step the new excel worksheet with method specific calculation is created, while the original data worksheet is not changed and could be used again for new calculations. When choosing the PROMETHEE method preference function has to be specified, and depending on the preference function some thresholds might be required. The example of the calculation is shown on Figure 20. In the case of Electre III and TOPSIS methods only the calculation and

Figure 19. Ranking example with Triptych Excel plugin
results are created. This means that the Electre III method is not completely implemented, which was confirmed by running the same example with Electre III/IV tool and comparing the results.

SANNA is limited to 50 criteria and 100 alternatives and does not support missing values. The plugin was used on Excel 2010 and some problems were experienced. Tool could be only used via menu, while the icons present at the ribbon were not functional and some expressions were not in English language as well as help which was completely in Czech language. The results of TOPSIS method were the same as the results obtained by Triptych tool and the results of PROMETHEE methods were the same as the results of Visual PROMETHEE Academic tool. On the other hand the Electre III method is not completely implemented and thus results were not the same as those obtained by Electre III/IV tool.

Visual PROMETHEE Academic

Visual PROMETHEE Academic is a multicriteria decision support software which implements both the PROMETHEE I and PROMETHEE II outranking methods, as described in chapter 3.2. Since it is based on building an outranking on the set of alternatives, the main objective of the tool is to find the alternative (i.e. the best ranked alternative) that is the most suitable for DM.

The tool has been developed by the University of Brussels and has been widely used worldwide in many different decision or evaluation problems (such as banking, human resources management, location of facilities, etc.) [80]. It is a MS-Windows program and runs on Windows XP, Vista, 7 and later operating system [81].

Visual PROMETHEE Academic can handle both quantitative and qualitative criteria values. Qualitative criteria values are based on qualitative scale which is defined by the number of ordered levels (from lowest to highest) or by the numerical values associated to ordered levels or whether the numerical values should be minimized or maximized (scale orientation). There are four predefined qualitative scales (y/n, impact 5-level scale ranging from “very low” to “very high”, classical 5-level scale ranging from “very good” to “very bad” and 9-point scale which extends 5-point scale with intermediate values). Visual
PROMETHEE Academic is also able to handle missing values by entering “?” in the information table and whenever a missing value is encountered, the resulting preference degree will be set to zero as if both actions had equal evaluations on that criterion. The six different types of preference function, as discussed in chapter 3, are also included in this tool. It has a MS Excel interface for importing/exporting data and is able to generate PDF and XLS reports.

Let’s demonstrate the features of Visual PROMETHEE Academic with an example of HW/SW software partitioning. The goal is to choose the component that is the most suitable for our embedded system. This MCDA problem consists of both quantitative and qualitative data and some values of criteria are unknown. The first step is a definition of alternatives and criteria hierarchy used for assessing alternatives. Example illustrated on the Figure 21 and Figure 22 consists of 5 alternatives and 30 criteria (16 criteria on the Figure 21 and 14 criteria on the Figure 22). For example, the criterion status is a qualitative measure of component and it has two possible values (Yes – active, No – not active). The criterion memory footprint has an unknown values for hardware components and quantitative values for software components. When alternative assessing on each criteria in hierarchy is done, the next step is to define weights and choose appropriate preference function for each criterion. Based on the user preferences it is possible to choose six different types of preference function, as discussed in chapter 3.2. The user without any preference may choose Usual type of preference function.

Figure 21. The user interface of Visual PROMETHEE Academic tool with 16 criteria and 5 alternatives
Since Visual PROMETHEE Academic offers several ways to present the rankings, the final ranking of all alternatives is shown on the Figure 23 and Figure 24. On the Figure 23 the PROMETHEE I Partial Ranking is presented. In this figure the left bar shows the ranking of the alternatives according to $\Phi^+$ and the right bar shows the ranking according to $\Phi^-$. We can conclude that alternative SW3 is preferred to all other alternatives in the PROMETHEE I ranking. This is confirmed by the PROMETHEE II complete ranking, as shown on the right figure.

Figure 22. The user interface of Visual PROMETHEE Academic tool with 14 criteria and 5 alternatives

Figure 23. PROMETHEE final rankings
The PROMETHEE table presented on the Figure 24 displays the Phi, Phi+ and Phi- scores. The alternatives are ranked according to the PROMETHEE II complete ranking. This result view can be very practical when the number of alternatives is large.

![PROMETHEE Flow Table](image)

**Figure 24.** PROMETHEE final rankings using table (the same result as on figure above)

One of the most important advantages of this tool is the ability to handle both qualitative and quantitative data. It is also possible to handle missing values at the pairwise comparison level. The final rankings could be displayed in a several ways. The academic edition of this tool is fully functional without any limits and is available for free for non-profit research and teaching only.

Visual PROMETHEE Academic has the limitation on the size of the decision problem. It can handle maximum 10000 alternatives and 10000 criteria. This tool is only supported by Windows operating system. Native Linux, MacOS and iOS versions of Visual PROMETHEE Academic are considered as a possible future development.

**Intelligent Decision System**

Intelligent Decision System (IDS) is a multi-criteria decision support tool which implements the Evidential Reasoning (ER) approach, as described in chapter 3.2. The main objective of tool is to evaluate several possible alternatives according to multiple conflicting criteria and rank them from the worst to the best one. The most suitable alternative for DM is always the best ranked.

The tool has been developed by the IDS Ltd and has been used in many different types of applications such as contract selection, supplier assessment, quality management, etc. It is fully functional without any limits and is available for free. This tool is only supported by Windows operating system [82].

The IDS tool is able to handle both quantitative and qualitative data. This software is also able to address different types of uncertainties such as randomness and ambiguity using belief structure based on Dempster-Shafer theory, as discussed chapter 3. It is capable to perform large-scale multi-criteria decision analysis problems with thousands of criteria [83].

Let’s demonstrate the features of IDS with an example of HW/SW partitioning. The goal is to choose the component that is the most suitable for our embedded system.
This MCDA problem consists of both quantitative and qualitative data and some values of criteria are unknown. The first step is a definition of alternatives and criteria hierarchy used for assessing alternatives. Example illustrated on the Figure 25 consists of 5 alternatives and 30 criteria. For example, the criterion status is a qualitative measure of component and it has two possible values (Yes – active, No – not active). The criterion memory footprint has an unknown values for hardware components and quantitative values for software components. When alternative assessing on each criteria in the bottom level of the criteria hierarchy is done, the next step is to define weights for each criterion. IDS provides two ways to assign weights systematically: visual scoring (Figure 26) and pairwise comparisons. The user who knows the AHP method very well may find the method of pairwise comparisons familiar. Pairwise comparison is more complicated than the visual scoring method and may not necessarily produce a better set of weights. Thus, it is recommended to use visual scoring.

Figure 25. IDS problem definition
The final ranking of all alternatives is shown on the Figure 27. The figure presents the performance scores for all the alternatives on the selected criteria. Because there are some missing data in the assessment information, the graph provides a lower and upper bound of the performance scores. Thus, the scores will be in the range marked in grey whatever the missing data turn out to be.

The advantage of using this tool is the ability to handle qualitative, quantitative and missing data. The IDS tool is fully functional without any limits and is available for free.

This tool has the limitation on the size of the decision problem. It cannot handle decision problems with more than 50 alternatives and 50 criteria. This tool is only supported by Windows operating system.

Figure 26. Visual scoring method for obtaining weights of each criterion

Figure 27. The final ranking of alternatives
4.2 Classification

Electre TRI

Electre TRI tool implements Electre TRI classification method and is used for assignment of alternatives in predefined classes/categories.

The Electre TRI tool has been developed by the Dauphine University of Paris and Institute of Computer Science of the Poznan University of Technology. It was developed by using C++ programming language. It runs on Windows 3.1 or newer operating systems [84], [85].

The tool handles quantitative data, provides results by category and alternatives its statistics and degrees of credibility. Alternatives’ values could be visualized on graph with profile values.

To start a new project must be created. After that, new criteria, alternatives and profiles have to be created. For the calculation to be done, all alternatives values have to be defined, as well as profile values, weights and thresholds for all criteria. The same as in Electre III/IV tool, veto threshold is optional, while preference and indifference thresholds are mandatory. When all the data are defined the results are calculated. Classification results could be then observed by category or by alternative. This is shown on Figure 28 as well as results statistics for a small example with three criteria and alternatives classified in two classes.

Figure 28. Electre Tri classification results and statistics
The tool is limited on 32 criteria and does not support missing values. Profile values have to be assigned manually by the DM which is not easy for complex problems.

4eMka2

4eMka2 is a tool that implements the DRSA method.

It was created with the purpose of solving MCDA classification problems in many different areas where vast data sets are analyzed.

4eMka2 was developed by a group of students at the Laboratory of Intelligent Decision Support System of the Institute of Computing science, Poznan University of Technology. It runs on Windows operating systems.

The tool enables extraction of the classification rules from a set of already classified examples. Extracted rules in the form of “if...then...” sentences are used to classify new examples. The tool supports qualitative and quantitative data as well as missing values [86].

Before using the tool data sets with training and testing examples have to be defined in text files with “isf” extension. The simplest typical use case would be opening the “isf” file with training examples, calculating the rules and using these rules for classification of testing examples. In the rules extraction process, one can choose “minimal cover” rules (minimal number of rules that can classify training examples) or manually choose minimal rules strength and maximum rules length. These rules can be saved and used next time. In the classification process it is also possible to choose rules that are going to be used. The tool provides a very good traceability which makes the classification process much easier. Figure 29 shows extracted rules in the left window and classification results in the right window. Lower windows parts provide a very good traceability. Rules window provide the training examples used for rules extraction, while classification results window provides rules used for example classification.

Figure 29. 4eMka2 tool, extracted rules on the left and classification results on the right
Positive side of the tool is that it supports both qualitative and quantitative data as well as missing values. It provides very good decision traceability which makes troubleshooting of weird results much easier. On the other hand, when there are a lot of examples to extract the rules from the tool cannot extract the rules in a reasonable time and the rules parameters have to be modified to limit the number of extracted rules. The other negative side of the tool is that the data cannot be modified within the tool, but only directly in the isf file.

**jMAF**

The jMAF is a tool for classification that implements Dominance-Based Rough Set Approach, as discussed in chapter 3.3. The main objective of this tool is extraction of the classification rules from a set of already classified examples. Extracted rules can be used to make partition of new data sets and they are presented as “if…then…” sentences.

jMAF is open source, cross-platform tool published under GNU General Public License, version 3 (GPL3) copyleft license. The tool is implemented in Java programming language. The current version requires JRE (Java Runtime Environment) version 1.6 or newer. Like any other Java program, it could just be run, it does not need installation. It can be downloaded for MAC, Linux and Windows operating systems [87].

jMAF handles quantitative values while handling missing values is not supported.

The main jMAF main window is divided into 4 windows: menu and toolbar, results, explorer and console window, as shown on the Figure 30.

![jMAF main window](image)

**Figure 30. jMAF main window**

The jMAF classification project starts by creating file with Information System File (ISF) format. A simple example of ISF file is:

```plaintext
**ATTRIBUTES
+ Crossing: (continuous)
+ Dribbling: (continuous)
+ Finishing: (continuous)
+ Heading: (continuous)
```
+ LongShot: (continuous)
+ LongThrow: (continuous)
+ State: [1, 2, 3]
decision: State

**PREFERENCES
+ Crossing: gain
+ Dribbling: gain
+ Finishing: gain
+ Heading: gain
+ LongShot: gain
+ LongThrow: gain
+ State: cost

**EXAMPLES
11 13 12 12 13 2 2
15 14 12 12 15 10 3
13 13 10 8 10 2 1
14 10 12 14 15 13 3
12 14 17 13 5 4 3

**END

The ISF file consists of three sections: **ATTRIBUTES, **PREFERENCES and **EXAMPLES. The sequence of sections cannot be changed, otherwise jMAF will report an error.

The **ATTRIBUTES section contains all criteria used in the decision problem. Symbol “+” denotes active criterion (i.e. this criterion should be used during analysis) and symbol “-“ denotes inactive criterion (i.e. this criterion should not be used during analysis). The symbol is followed by the name of the criterion followed by the domain definition (i.e. symbolic or numeric domain). The symbolic domain is defined with the list of possible values in square brackets. The numeric domains can be either integer (i.e. integer-valued criterion) or continuous (i.e. real-valued criterion). In the end of **ATTRIBUTES section is declaration of the decision criterion which has to be active with symbolic domain.

The **PREFERENCES section contains information on the direction of preferences for all criteria defined in the **ATTRIBUTES section. The name of the criteria is followed by the name of direction which can be gain (i.e. greater values are preferred for numeric domain and the last value in the definition list is preferred for symbolic domain), cost (i.e. lower values are preferred for numeric domain and the first value in the definition list is preferred for symbolic domain) or none (i.e. no preference is taken into account).

The **EXAMPLES section contains the values of the alternatives with respect to each criterion. Each line represents one alternative and the values should be given in the same order as the criteria in the **ATTRIBUTES section. The values between criteria can be separated either with “,” or tab character.

When ISF file is created, the first step is to calculate the approximations. The user should navigate to the top menu and click Calculate|Standard unions of classes to calculate DRSA unions and approximations. The second step is calculation of the list of all reducts by clicking Calculate|All reducts. Third step is dealing with induction of a set of decision rules with minimal covering rules, as shown on the Figure 31. The final step may be to reclassify testing examples (i.e. find out how good the obtained rules in the third step can classify new
examples). On the Figure 32 only the first example is classified correctly, while others are misclassified.

On the other hand jMAF does not handle missing values.
The ROSE2 is a tool designed to analyze data by means of the rough set theory. It implements basic elements of the rough set theory and all computations are based on the rough set approach as described in chapter 3.3.

The main objective of this tool is extraction of the classification rules from a set of already classified examples. Extracted rules can be used to make partition of new data sets and they are presented as “if...then...” sentences.

The tool has been developed at the Laboratory of Intelligent Decision Support Systems of the Institute of Computing Science in Poznan. It runs on Windows operating system, while the Linux and Mac OS are not supported [88].

The ROSE2 is a specific type of multi-criteria decision analysis tool based on a classification. This tool concerns an experience that is recorded in a structure called an information system. The information system may contain various types of information (e.g. events, observation, states, etc.) in terms of their criteria (e.g. variables, characteristics, symptoms, etc.). Thus, the input data to this tool is the information table where rows correspond to alternatives and columns correspond to criteria. There are two groups of criteria. The first group is called condition criteria and the second group is decision criteria. Input data is stored in a plain text file called the information system file (ISF), which is in different format compared to previously defined ISF file in jMAF. A simple example of ROSE2 ISF file is:

```
**ATTRIBUTES
PlatformHW: [Lower, Indifferent, Higher]
ImplementationFormat: [Lower, Higher]
P2Ptime: (continuous)
Heading: (continuous)
WCET: (continuous)
LOC: (continuous)
DECISION: [Hardware, Software]
decision: DECISION

**EXAMPLES
Lower   Lower 12 12 13 2 Hardware
Higher  Higher 12 12 15 10 Hardware
Indifferent  Lower 10 8 10 2 Software
Higher  Lower 12 14 15 13 3 Hardware
Lower  Higher 17 13 5 4 Software

**END
```

The **ATTRIBUTES section contains all criteria used in the decision problem. The name of the criterion is followed by the domain definition (i.e. symbolic or numeric domain). The symbolic domain is defined with the list of possible values in square brackets. The numeric domains can be either integer (i.e. integer-valued criterion) or continuous (i.e. real-valued criterion). In the end of **ATTRIBUTES section is declaration of the decision criterion with symbolic domain.

The **EXAMPLES section contains the values of the alternatives with respect to each criterion. Each line represents one alternative and the values should be given in the same order as the criteria in the **ATTRIBUTES section. The values between criteria can be separated either with “,” or tab character.
When the ISF file is finished, we can perform data analysis on it. Data analysis techniques are divided into five groups, as shown on the Figure 33. The first group is called preprocessing and consists of techniques used for preliminary data analysis and modifications (like discretization). Techniques dealing with the reduction of attributes could be found in the reduction group. Rule induction, the third one, consists of techniques used for generating decision rules while the validation of decision rules is grouped in validation group. The last one is called the similarity relation group and consists of techniques using similarity relation approach. Rough set approximation technique is not included in any of previously mentioned groups.

Figure 33. Data analysis methods

The final results (i.e. generated rules) are also stored in a plain text file, as shown on the Figure 34.

Figure 34. Example of generated rules
ROSE2 tool offers many features, such as handling incomplete information tables, data preprocessing, performing a standard and an extended analysis of data, search of a core and reducts of attributes, using sets of decision rules as classifiers, etc. [89].

ROSE2 tool is able to handle quantitative and qualitative data. For the tool to be used a set of examples is needed to be able to extract rules from.

This tool is only supported by Windows operating system and it does not offer feature for reclassification of learning data for which rules were induced. It has to be done manually by comparing the obtained rules and learning data.

4.3 Ranking and Classification

JSMAA

JSMAA tool implements SMAA-2 and SMAA-O ranking methods and SMAA-TRI method for classification.

JSMAA is open source, cross-platform tool implementing SMAA-2, SMAA-O and SMAA-TRI methods. It is published under GNU General Public License, version 3 (GPL3) copyleft license.

The tool is implemented in Java programming language. The current version requires JRE (Java Runtime Environment) version 1.6 or newer. Like any other Java program, it could just be run, it does not need installation. It was tested on Linux and Windows operating systems [90].

JSMAA handles quantitative values which can be expressed as discrete values, intervals, Gaussian distributions or functions. Criteria weights could be ordinal, cardinal or missing. It provides a very good results visualization in a table and graph form.

The ranking project is created by choosing SMAA-2 model from the menu. It implements SMAA-2 and since it supports ordinal criteria weights it also implements SMAA-O method which is extension of SMAA-2 method. In the left part of the window are listed alternatives and criteria while the right part is used for inserting the data and observing the results. When criteria are added, criteria weight could be specified. When the data is added or changed the calculation is performed automatically. Figure 35 shows the example of ranking results, with table and graph visualization of its values.
The classification project could be created by choosing SMAA-TRI model from the menu. It implements SMAA-TRI method. The application window and procedure are similar as for ranking. In the left part of the window, besides alternatives and criteria, are also listed categories (classes). The other difference is that profile values have to be specified before getting the results. Figure 36 shows the example of classification results, similar to ranking, the table and graph visualization are available.
The advantage of this tool is that it is cross platform, so it could be run on almost all operating systems. It is open source, so it could be modified as long as the terms of the license are not violated. It can handle range of values, distributions as well as cardinal, ordinal and missing criteria weights. The presentation of results is simple and easy to understand. On the other hand JSMAA does not handle missing values and profile values needed for classification have to be assigned manually by the DM which is not easy for complex problems.
Chapter 5

MCDA METHODS AND TOOLS FOR ENABLING THE PARTITIONING

5.1 Partitioning process

The goal of the thesis is to define an HW/SW partitioning process that will help the DM to make the decision using the MCDA methods and tools.

The HW/SW partitioning process is a set of rules and steps that had to be followed in order to decide which components should be implemented as hardware and which components as software. The partitioning process includes the definition of the input data, the MCDA method and tools used in the process as well the step-by-step instructions how to get the final result. The step-by-step instructions include the process limitations and other details that could negatively influence the final result.

In order to be able to define the MCDA-based partitioning process, a number of MCDA methods and tools had to be analyzed and tested, with their capabilities, advantages, disadvantages and limitations in mind. All the investigated methods and tools were presented in previous chapters.

Before being able to specify how the partitioning process based on MCDA approach can be carried out the requirements on the partitioning have to be defined, these latter are subsequently discussed.

After the process requirements have been defined, the most suitable methods and tools criteria were identified. Some criteria are methods specific (e.g. the rank reversal problem), while other criteria are tools specific (e.g. the visualization of the final result). A lot of the tools and methods criteria are identical (e.g. the maximum number of criteria and alternatives in a decision process).

The main objective of tools and methods criteria is to identify all methods and tools that are the most suitable for the partitioning process. Partitioning processes presented in the chapter 6.2 are defined based on these criteria.
5.2 Partitioning process requirements

In this section we present the key requirements that the overall MCDA method (or combination of methods) has to be able of handling/providing for enabling the partitioning process. The overview of all requirements is illustrated on the Figure 37. Each group of requirements is subsequently presented by a table. Additionally, the limitations or issues with respect to the evaluated MCDA methods and tools are discussed for each group.

![Diagram of requirements]

Figure 37. The structure of requirements
Table 7. Process Requirement Table

| Req 1.1 | The process has to be systematic and supportive, and it has to allow iterations at different steps. It has to have well-defined steps and activities. |
| Req 1.2 | The process has to be able to handle the classification and/or ranking of hardware and software units to be partitioned. |
| Req 1.3 | The process has to be able of providing feedback in terms of quality prediction, measuring the performance/accuracy of the solution proposed (related to uncertainty) and be able to get the sensitivity feedback. |
| Req 1.4 | The process has to be able of providing feedback in terms of rating the decision risk. |
| Req 1.5 | The process should provide metrics for assessment of the quality of the overall partitioning solution. |
| Req 1.6 | The process has to be able scalable/extendable (i.e. it must not be restricted to a limited number of units or alternatives). |
| Req 1.7 | The process has to be as much as possible transparent to the DMs and stakeholders. |
| Req 1.8 | The process has to be traceable (i.e. decisions have to be back and forth traceable). |

The requirement 1.3. cannot be handled by any tool/method presented in this report. Actually, the evaluated methods and tools are not able to provide any feedback about quality prediction, measuring the performance/accuracy of the solution proposed (related to uncertainty) and be able to get the sensitivity feedback.

Since it is not recommended to use AHP and ANP methods for problems with more than 10 criteria and alternatives, the requirement 1.6. can be partial fulfilled. Also, some tools (e.g. SANNA or Visual PROMETHEE Academic) have a limitation on the maximum number of criteria and alternatives.

Table 8. Alternatives Requirement Table

| Req 2.1 | There exist a finite number of alternatives - discrete set. Each alternative might have associated different criteria. |
| Req 2.2 | Limits on alternatives: min 2 - max number not limited. |
| Req 2.3 | There exists different type of alternatives HW or SW units (also referred as component), or virtual (i.e. not specified if the alternative is HW or SW). |
| Req 2.4 | Each alternative (also referred as variant) has to be classified and/or ranked with respect to the criteria. |
| Req 2.5 | Each alternative might have associated different properties (which means it will be associated to different criteria), as a consequence it might exist |
missing values for some criteria.

| Req 2.6. | Each alternative might have associated a criticality/importance/weight parameter. |
| Req 2.7. | Each alternative might have associated an accuracy parameter. |
| Req 2.8. | Each alternative might have associated a reliability parameter, indicating how reliable are the values with respect on how they are achieved (measured, estimated, simulated, etc.). |
| Req 2.9. | An alternative can be dependent by other alternatives, which do not belong to the same components - dependencies among alternatives have to be handled. |

The requirement 2.2. can be partially handled with some MCDA method or tool because for instance it is not recommended to use either AHP or ANP methods for problems which deal with more than 10 alternatives and criteria. Also, some tools like SANNA or Visual PROMETHEE Academic have a limitation on the maximum number of alternatives.

Only few methods and tools support the handling missing values (e.g. PROMETHEE, ER) thus the requirement 2.5. can be partially handled.

The requirement 2.9. cannot be fulfilled because the dependencies between alternatives are not supported by any tool or method.

Table 9. Criteria Requirement Table

| Req 3.1. | There exist a finite number of criteria - discrete set. |
| Req 3.2. | Limits on criteria: min 2 - max number not limited. |
| Req 3.3. | There exist different type of criteria, qualitative and quantitative, mixed types. |
| Req 3.4. | Criteria data type: discrete values, range (discrete or continue), missing values, etc. |
| Req 3.5. | A criterion can be dependent by other criteria - dependencies among criteria have to be handled. |
| Req 3.6. | Prioritization among criteria, or a group of has to be allowed. |
| Req 3.7. | Has to handle preference-ordered criteria (attributes with preference-ordered domains). |
| Req 3.8. | Has to handle conflict among criteria. |
| Req 3.9. | Has to handle incommensurable units. |

The requirement 3.2. can be partially handled because it is not recommended to use AHP and ANP methods for problems with more than 10 criteria. Also, some tools like SANNA or Visual PROMETHEE Academic have a limitation on the maximum number of criteria.
Only few methods and tools support handling missing values (e.g. PROMETHEE, ER) and qualitative values (e.g. ER) thus the requirements 3.3. and 3.4. can be partially handled.

The requirement 3.5. cannot be fulfilled by the tools and methods investigated in this work because they are not able to deal with dependencies between criteria.

Some methods (e.g. Electre IV and DRSA) do not support criteria weighting, so if these tools are used the requirement 3.6. would not be fulfilled.

Table 10. Decision Requirement Table

| Req 4.1. | Allowing to have multiple DMs in the process. |
| Req 4.2. | Allowing to express DMs’ preferences. |
| Req 4.3. | Allowing easy-review of the decision process at different stage. |
| Req 4.4. | Allowing to take into account history of previous decision. |

Since the classification methods/tools do not provide possibility for expressing DM’s preferences among criteria/alternatives, the requirement 4.2. can only be fulfilled with the ranking methods/tools.

Requirement 4.4. can be fulfilled only with classification methods.

Table 11. Prioritization Requirement Table

| Req 5.1. | Prioritization mechanisms have to be allowed based on DMs and stakeholders. |
| Req 5.2. | Prioritization among alternatives or a group of to be allowed. |
| Req 5.3. | Prioritization among criteria or a group of to be allowed. |
| Req 5.4. | Prioritization among partitioning solutions has to be provided, in case of multiple feasible solutions. |

Table 12. Filtering Requirement Table

| Req 6.1. | Filtering mechanisms has to be allowed, based on DMs and stakeholders. |
| Req 6.2. | Filtering among alternatives or a group of to be allowed/ performed. |
5.3 Partitioning-driven criteria for evaluating the MCDA methods

In order to evaluate and identify which MCDA methods are suitable to enable the partitioning process, the following listed criteria (acting in this case as properties of a method) have been identified. They are derived by the aforementioned requirements.

- Configurability
  - Many parameters and thresholds have to be set up in a multicriteria model (e.g. Promethee requires to associate a preference function and related threshold value to each criterion in order to model the way the DM perceives the measurement scale of the criterion). Some methods do support setting parameters values (e.g. in VIKOR) while other methods do not (e.g. AHP). Setting threshold values can be mandatory (e.g. in Electre III Indifference and preference thresholds) or optional (e.g. in Electre III Veto threshold).

- Dependence/interdependence between criteria
  - Does the method consider the interdependence relationship between the criteria into the evaluation based on criteria weight?

- Dependence/interdependence between alternatives
  - Does the method consider the interdependence relationship between the criteria into the evaluation based on alternatives weight?

- Grouping of criteria
  - A criteria group defines a subset of criteria. Each criteria group belongs to a cluster.

- Grouping of alternatives
  - An alternatives group defines a subset of alternatives. Each alternatives group belongs to a cluster.

- Dealing with Inconsistent, Quantitative, Qualitative and Missing Values
  - Each alternative might have associated different properties (which means it will be associated to different criteria), as a consequence might exist missing, quantitative or qualitative values for some criteria with respect to the considered alternative.
  - Check the inconsistency of decision by calculating the inconsistency ratio. If the inconsistency ratio is greater than predefined value, there is a need

| Req 7.1. | If not all of the information related to a components are available or if they are weakly formalized, the proposed method has to be able either of asking to complement the missing information or able of taking a decision, and notify that the decision was taken under uncertainties and missing values. |
| Req 7.2. | Has to be able of handling uncertainties and missing criteria values. |
to revise the subjective judgment (i.e. in AHP and ANP method it is necessary to revise pairwise comparisons between alternatives or criteria).

- Dealing with the range of values and distribution
  - Each alternative might have associated different properties (which means it will be associated to different criteria), as a consequence might exist range of values or distribution for some criteria with respect to the considered alternative.

- Allow subcriteria
  - Some methods allow organizing criteria into a three-level hierarchy. This is especially useful when the number of criteria is large.

- Scalability – maximum number of criteria, subcriteria and alternatives
  - Some methods have the limits relative to the dimensions of the decision problems they can handle (e.g. in AHP and ANP methods it is not recommended to handle large number of criteria and alternatives, e.g. for 10 criteria and alternatives it would be necessary to perform 45 pairwise comparisons between criteria and 45 pairwise comparisons between alternatives with respect to criteria)

- Allow Criteria Weighting
  - Weight the criteria within the criteria hierarchy.

- Allow Alternative Weighting
  - Weight the alternatives within the alternatives hierarchy.

- Decision Traceability
  - Following the life of a decision – from the idea to the implementation.

- Need for values normalization
  - We suppose here that the values in decision table are normalized in such a way that their sum is equal to 1 (100%).

- Need for weight values normalization
  - We suppose here that the weights are normalized in such a way that their sum is equal to 1 (100%).

- Rank reversal problem
  - Some methods (e.g. AHP and ANP) suffer from the rank reversal problem (i.e. the ranking can be reversed when a new alternative is introduced).

- Available tools that implements method
  - For some methods (e.g. VIKOR) there are no available tools developed that implements that method.

- Limitations
  - E.g. does the method have a limitation on the maximum number of criteria or alternatives? Does the method support handling missing/qualitative/etc. values?
• Benefits
  o What is a benefit of certain method? (e.g. this method is able to handle both missing and qualitative values, etc.)

5.4 Partitioning-driven criteria for evaluating the MCDA tools

MCDA tools have a lot of different properties. Some properties are tools-specific like MCDA methods implemented, while the other properties are the same as the methods properties like maximum number of criteria supported. Some of these properties influence the partitioning process very much and are identified as tools criteria.

These are tools properties that had been identified as important for the partitioning process:

• Configurability
  o There are a lot of different parameters that could be configured, like preference functions for PROMETHEE method or thresholds for Electre methods. Some of them might be mandatory and others optional.

• Scalability
  o MCDA tools should be able to handle big problems, to be able to use it even if the original problem grows beyond original plans.

• Dealing with Inconsistent Values
  o Check the inconsistency of the decision by calculating the inconsistency ratio. If the values is greater than the predefined value, the subjective judgment of the DM should be revised.

• Dealing with Missing Values
  o Since some values in the partitioning process are not known and would be hard to estimate, so it is preferred that the can handle these type of problems.

• Allow Criteria Weighting
  o Weight the criteria within the criteria hierarchy.

• Decision Traceability
  o Is it possible to get results at some calculation stages and not only the final result? Is it possible to understand and trace the reason if the result might seem weird and suspicious.

• Handle interdependencies between criteria
  o Does the tool consider the interdependence relationship between the criteria into the evaluation based on criteria weights?

• Handle interdependencies between alternatives
  o Does the tool consider the interdependence relationship between the alternatives into the evaluation based on alternative weights?

• Allow Alternative Weighting
  o Weight the alternatives within the alternatives hierarchy.
• Type of values supported
  o The values could be expressed as qualitative values (good, bad) or quantitative values (integer, float).
• Need for values normalization
  o It is supposed that the values are normalized in such a way that their sum is equal to 1 (100%).
• Need for weight values normalization
  o It is supposed that the weights are normalized in such a way that their sum is equal to 1 (100%).
• Results visualization
  o Result visualization could be some type of graph or even a table with different colors that makes it easier to distinguish between bad and good values.
• Manual/Help comprehensibility
  o User manual should provide assistance to the beginner users as well as more advanced users.
• Tool assistance (Customer support)
  o Assistance by the developer to their customers in form of support during installation, providing training, troubleshooting, and maintenance.
• System requirements
  o Some tools require additional software to be able to run.
• License type
  o Type of the software license, it could be closed and open-source both free and commercial.
• License cost
  o Price of the software.
• Limitations
  o Tools limitations could be both limitations of implemented methods (i.e. maximum number of criteria and alternatives) and tools not implementing complete methods (i.e. tool not implementing threshold values although the method supports them).
• Benefits
  o What are the benefits of a certain tool? (i.e. the possibility to handle missing values).
• Report generation
  o Export the data and the results of the analysis.
Chapter 6

RESULTS

6.1 Analysis and evaluation

MCDA methods evaluation

In the previous chapter the process requirements were described, and the important criteria for evaluating MCDA methods and tools have been defined. There are both ranking and classification methods in the same table.

Overall there are 14 methods that have been evaluated and its properties were put in the table provided in the Appendix A.

MCDA tools evaluation

The previously defined process requirements were used to identify the important tools properties. The methods exploration and testing has been done together with the methods exploration. As well as for methods, there was a table that was used to collect all the data.

Overall 14 tools have been tested. VIKOR method is the only method that does not have any tool that implements it, while there are several tools that implement more than one method.

For all the tools that are not free, the trial version was tested. Besides, the tools that are presented in the table there are some general web MCDA tools that are not mentioned in the table because they provide only basic functionalities, and there weren’t any details about the MCDA method implemented. The table summarizing the MCDA tools with respect to the properties is available in the Appendix B.

Methods vs tools

Table 14 visualizes all the tools with respect to methods that they implement. Tools that implement ranking methods are presented on the red background, those that implement classification methods have the blue background and JSMAA is the only tool that implements both classification and ranking methods. This latter has the purple background. It could be seen that there are much more methods and tools for ranking. This is because ranking methods were first MCDA methods developed, there are more ranking methods developed and are more popular than classification methods.
Table 14. MCDA tools and methods that implement

<table>
<thead>
<tr>
<th>Tools</th>
<th>ELECTRE III/IV</th>
<th>ELECTRE IV</th>
<th>AHP</th>
<th>ANP</th>
<th>VIKOR</th>
<th>TOPSIS</th>
<th>PROMETHEE I</th>
<th>PROMETHEE II</th>
<th>Evidential Reasoning</th>
<th>SMAR-2</th>
<th>SMAR-TRI</th>
<th>ELECTRE TRI</th>
<th>RSA</th>
<th>DRS-SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELECTRE III</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ELECTRE IV</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AHP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANP</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VIKOR</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOPSIS</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROMETHEE I</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROMETHEE II</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evidential Reasoning</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMAR-2</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMAR-TRI</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ELECTRE TRI</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSA</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRS-SA</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Partitioning process requirements vs. Methods and Tools

Table 15 shows the properties or characteristics that were identified as the most important for the method selection.

Table 15. The most important criteria used for the method selection

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>ELECTRE III</th>
<th>ELECTRE IV</th>
<th>AHP</th>
<th>ANP</th>
<th>VIKOR</th>
<th>TOPSIS</th>
<th>PROMETHEE I</th>
<th>PROMETHEE II</th>
<th>Evidential Reasoning</th>
<th>SMAR-2</th>
<th>SMAR-TRI</th>
<th>ELECTRE TRI</th>
<th>RSA</th>
<th>DRS-SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranking</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classification</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scalability</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interdependencies support</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Criteria weights assigned by the stakeholder</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Able to handle quantitative values</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Able to handle qualitative values</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Able to handle distributions</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Able to handle range of values</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Able to handle missing values</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Able to handle inconsistent values</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decision traceability</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tools available</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The most important methods properties that are shown in the previous table were used as criteria for the selection of the best MCDA methods for the HW/SW partitioning process. The search after the best methods began with the most popular ranking methods like Electre and AHP. AHP method is not recommended to be used with a large number of criteria and alternatives (not more than 10), and the partitioning process would certainly have more than 10 alternatives and criteria. ANP has the scalability problem as well. Topsis and VIKOR methods do not support missing values. Only Evidential Reasoning and PROMETHEE II methods support missing values. Moreover, Evidential Reasoning supports qualitative values out of the box, while others could support it if the values are quantified. Unfortunately none of the methods supports any kind of interdependencies.

As explained better in the next subchapter, although the HW/SW partitioning process could be done using only the ranking methods (using hardware and software components), the classification partitioning process provides the benefit of using virtual components defined only by their requirements. Otherwise the values for both software and hardware alternatives would have to be calculated or estimated. Four classification methods were considered. RSA and DRSA methods support qualitative and missing values. SMAA-TRI method could on the other hand handle ranges of values as well as distributions and stakeholders or DM could set criteria weights. Electre TRI satisfies the least number of important criteria.

Overall Evidential Reasoning and PROMETHEE II methods would be the best ranking methods for the HW/SW partitioning. The DRSA method would be the best method for the classification process. Although SMAA-TRI method has some good properties, the DM has to provide the profile values as well as threshold values for all the criteria. This is harder than using the existing SW and HW components for the knowledge extraction.

Table 16. The most important criteria used for the tool selection

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>ELECTRE II/IV</th>
<th>ELECTRE TRI</th>
<th>Makel\text{\textregistered} Rational Professional</th>
<th>ANP SOLVER</th>
<th>WEB/ANP SOLVER</th>
<th>Transparent Choice</th>
<th>TraPrych</th>
<th>SANA</th>
<th>Visual Promethee Academic</th>
<th>Intelligent Decision System (IDS)</th>
<th>SMAA</th>
<th>4PMa2</th>
<th>IMAF</th>
<th>ROSE 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranking</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classification</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scalability</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interdependencies support</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Criteria weights assigned by the stakeholder</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Able to handle quantitative values</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Able to handle qualitative values</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Able to handle distributions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Able to handle ranges of values</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Able to handle missing values</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Able to handle inconsistent values</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decision traceability</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
Table 16 shows the properties or characteristics that were identified as the most important for the tools selection. These were used as criteria for the tool selection and are the same as criteria used for the method selection. The tools mostly have the same drawbacks as the methods they implement.

The Visual PROMETHEE Academic tool that implements PROMETHEE II method and the Intelligent Decision System (IDS) that implements Evidential Reasoning method are the best suited tools that are used for ranking partitioning process.

The 4eMka2 and the jMAM tool both implement the DRSA method. Unlike 4eMka2 jMAM does not support the missing values and therefore it could only be used when there are no missing values. 4eMka2 satisfies the most number of important criteria and therefore is the best suited for the classification partitioning process.

6.2 MCDA-based partitioning process definition

Systems usually consist of multiple components. In the following subchapters are described two partitioning processes that are specified for the single component, so there must be used one such process per component.

There are two possible scenarios in the partitioning process. The first one is when there are existing component alternatives available and when there is the data available for new hardware and/or software components. In this case the ranking process should be used. The second scenario is when the component is newly designed and there are no component alternatives available. In this case the component is defined by its criteria and the decision has to be made whether it is better to implement it as software or hardware. In this case the classification process should be used. In order to use the classification process there must be component alternatives available for training examples used for knowledge extraction. In the second scenario it would be also possible to use ranking partitioning process if the data (calculated or estimated) is available for both software and hardware new component alternatives. In this case it would take more time to calculate or estimate values for at least two components than just calculating the requirements.

If the system consists of new and existing component, both processes could be used on the same system, depending on the particular components. The ranking process could be used for components with existing component alternatives and the classification process for new components that could be defined by their requirements.

Ranking

The ranking partitioning process is used when there is the data available for existing and new, hardware and software component alternatives. The process consists of three steps and it is shown on Figure 38.
The first step is the filtering process that is used to remove all the component alternatives that do not satisfy component and/or system requirements. These component alternatives are therefore not considered and the results calculation is faster and results are easier to understand.

In the second step, the data is transformed in such way that it could be handled by the selected tool/method. For example qualitative values could be quantified, and ranges of values and distributions could be represented by a single value. However, this process depends very much on the criteria. In some cases it might be the best to use mean value and in others minimum and/or maximum values.

In the third step the selected MCDA tool and method are used to obtain the final ranking of component alternatives.

Classification

The prerequisite of the classification partitioning process is to have existing database of the correctly classified component alternatives to extract the knowledge from. The ranking partitioning process could also be used but it would require more time and effort to calculate or estimate values for both software and hardware new component alternatives. This process consists of three steps shown on Figure 39.

In the step one the training examples data is transformed in such way that it could be handled by the selected tool/method, just like in the step two of the ranking process.

Step two is the knowledge extraction. In this step the knowledge is extracted from the training examples. The result of this step is a set of rules that should are able to correctly classify training examples. For this step to provide good results it is necessary to have a good set of training examples.

The third step is the classification of new component as hardware or software using the rules extracted in step two.

Figure 39. Classification partitioning process
6.3 Wind turbine – case study

In order to show the applicability of the proposed, we have used a wind turbine application which has to be deployed on a heterogeneous platform that consists of HW (FPGA) and SW (microcontroller).

The main objective of the wind turbine application is to control the transformation of the rotational mechanical energy of the rotor blades, caused by the wind, into electrical energy, which will be re-distributed via a power network. The core element of the application is the controller, which has to dynamically regulate the rotor blades at different wind profiles while maximizing the generation of electrical energy and avoiding any damage to the plant. The application is modeled as a number of components, which have to be deployed as HW component in the FPGA and SW components in the microcontroller.

To be able to run HW/SW partitioning examples using the previously defined processes, the partitioning criteria for the wind turbine application have been identified and the data has been estimated, or reused from existing components. There have been the following criteria identified:

- Variant related properties
  - HW implementation platform preference – qualitative value, better value is preferred
  - SW implementation platform preference – qualitative value, better value is preferred
  - Implementation format preference – qualitative value, better value is preferred
  - Release year – integer value, higher values is preferred
  - Status, active or not active, active is preferred
  - DM’s variant preference – qualitative value, better value is preferred

- Extra-functional properties
  - Input to output port execution time – float value, lower value is preferred
  - Worst case execution time – float value, lower value is preferred
  - Memory footprint (for SW component alternatives) – integer value, lower value is preferred
  - Lines of code – range of values, lower value is preferred
  - Number of gates per FPGA – range of values, lower value is preferred
  - Power consumption – float values, lower values is preferred
  - Number of subcomponents – Gauss distribution, lower values is preferred
  - Complexity, taking into consideration internal and external dependencies – qualitative value, lower value is preferred

- Project related properties
  - Time to rework – integer value, lower value is preferred
  - Time to design – integer value, lower value is preferred
  - Time to implement – integer value, lower value is preferred
  - Time to test – integer value, lower value is preferred
- Time to maintain – integer value, lower value is preferred
- Weeks before/over the design deadline – integer value, lower value is preferred
- Weeks before/over the implementation deadline – integer value, lower value is preferred
- Weeks before/over the test deadline – integer value, lower value is preferred
- Cost to buy (USD) – integer value, lower value is preferred
- Cost to test (USD) – integer value, lower value is preferred
- Cost to maintain (USD) – integer value, lower value is preferred
- Confidence level of the designer – qualitative value, better value is preferred

- Application related properties
  - Maximum execution time – integer, lower value is preferred
  - Portability – qualitative value, higher value is preferred
  - Upgradeability – percentage value, higher value is preferred
  - Maintainability, number of services over 10 years – Gauss distribution, lower value is preferred

All the criteria have the weights defined. Higher the criterion weight value, higher the criterion importance.

**Ranking example – Visual PROMETHEE Academic & IDS**

In this example we are considering one component of the wind turbine. It has 3 hardware and 3 software available instances. To be able to run HW/SW partitioning examples using the previously defined processes, the partitioning criteria have been identified and the data has been estimated or reused from existing components. The weight values for each criterion have been estimated together with the type of preference (Max means that the higher criteria values are preferable and Min means that the lower criteria values are preferable). The partitioning criteria and the data are shown in the Table 17. Since the table is rather big it is hard to have a good overview of the whole table. Thus table is divided into five small tables that are presented subsequently.

**Table 17. The partitioning criteria and the data**
Table 18. The partitioning criteria and the data

<table>
<thead>
<tr>
<th>Variant-related Properties</th>
<th>Weight [1..10] (low priority 1, high priority 10)</th>
<th>Preference Scale Among Variants (Lower, Indifferent, Higher)</th>
<th>Status (Active? In use?) (YES/NO)</th>
<th>Release Year</th>
<th>Implementation Format (Source Code, Binary, Model) more enumerative</th>
<th>Preference Scale Among Variants (Lower, Indifferent, Higher - Preferable)</th>
<th>Implementation Platform - SW</th>
<th>Implementation Platform - HW</th>
<th>Type (HW, SW, Virtual_HW, Virtual_SW), one enumerative</th>
<th>VARIANT_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Higher</td>
<td>Higher</td>
<td>?</td>
<td>2011</td>
<td>Source Code, Model</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>CS1</td>
</tr>
<tr>
<td></td>
<td>Higher</td>
<td>Higher</td>
<td>?</td>
<td>2012</td>
<td>Binary</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>CS2</td>
</tr>
<tr>
<td></td>
<td>Higher</td>
<td>Higher</td>
<td>?</td>
<td>2013</td>
<td>Preference Scale Among Variants (Lower-Preferable, Indifferent, Higher - Preferable)</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>CS2.1</td>
</tr>
<tr>
<td></td>
<td>Higher</td>
<td>Higher</td>
<td>?</td>
<td>2009</td>
<td>Preference Scale Among Variants (Lower-Preferable, Indifferent, Higher - Preferable)</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>CS2.2</td>
</tr>
<tr>
<td></td>
<td>Higher</td>
<td>Higher</td>
<td>?</td>
<td>2009</td>
<td>Preference Scale Among Variants (Lower-Preferable, Indifferent, Higher - Preferable)</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>CS2.3</td>
</tr>
<tr>
<td></td>
<td>Higher</td>
<td>Higher</td>
<td>?</td>
<td>2009</td>
<td>Preference Scale Among Variants (Lower-Preferable, Indifferent, Higher - Preferable)</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>CS2.4</td>
</tr>
<tr>
<td></td>
<td>Higher</td>
<td>Higher</td>
<td>?</td>
<td>2009</td>
<td>Preference Scale Among Variants (Lower-Preferable, Indifferent, Higher - Preferable)</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>CS2.5</td>
</tr>
<tr>
<td></td>
<td>Higher</td>
<td>Higher</td>
<td>?</td>
<td>2009</td>
<td>Preference Scale Among Variants (Lower-Preferable, Indifferent, Higher - Preferable)</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>CS2.6</td>
</tr>
<tr>
<td></td>
<td>Higher</td>
<td>Higher</td>
<td>?</td>
<td>2009</td>
<td>Preference Scale Among Variants (Lower-Preferable, Indifferent, Higher - Preferable)</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>CS2.7</td>
</tr>
</tbody>
</table>
Table 19. The partitioning criteria and the data

<table>
<thead>
<tr>
<th>VARIANT</th>
<th>Type</th>
<th>Execution Time</th>
<th>Component Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Worst Case Execution Time (among the different ports)</td>
<td>Memory Footprint</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PostShort Time</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SW</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HW+SW</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HW</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SW</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HW+SW</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HW</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variant</th>
<th>Execution Time</th>
<th>Component Size</th>
<th>Extra-Functional Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variant</th>
<th>Execution Time</th>
<th>Component Size</th>
<th>Extra-Functional Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variant</th>
<th>Execution Time</th>
<th>Component Size</th>
<th>Extra-Functional Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variant</th>
<th>Execution Time</th>
<th>Component Size</th>
<th>Extra-Functional Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variant</th>
<th>Execution Time</th>
<th>Component Size</th>
<th>Extra-Functional Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variant</th>
<th>Execution Time</th>
<th>Component Size</th>
<th>Extra-Functional Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variant</th>
<th>Execution Time</th>
<th>Component Size</th>
<th>Extra-Functional Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variant</th>
<th>Execution Time</th>
<th>Component Size</th>
<th>Extra-Functional Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variant</th>
<th>Execution Time</th>
<th>Component Size</th>
<th>Extra-Functional Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variant</th>
<th>Execution Time</th>
<th>Component Size</th>
<th>Extra-Functional Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 20. The partitioning criteria and the data

<table>
<thead>
<tr>
<th>Type (HW, SW, Virtual_HW, Virtual_SW), one enumerative</th>
<th>YES/NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>VARIANT_ID</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 21. The partitioning criteria and the data

<table>
<thead>
<tr>
<th>Project-related properties</th>
<th>Weight [1..10]</th>
<th>YES</th>
<th>Min</th>
<th>Max</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
<th>Very High</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
<th>Very High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidence Level of the Designer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Low, Medium, High)</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight [1..10]</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cost-to-maintain (USD)</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight [1..10]</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cost-to-test (USD)</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight [1..10]</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cost-to-buy (COTS,Pr) (USD)</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight [1..10]</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weeks over/before test deadline</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight [1..10]</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weeks over/before implementation deadline</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight [1..10]</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weeks over/before design deadline</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight [1..10]</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time-to-Maintain (Hours)</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight [1..10]</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time-to-Test (Hours)</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight [1..10]</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time-to-Implement (Hours)</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight [1..10]</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time-to-Design (Hours)</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight [1..10]</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time-to-Rework (Hours)</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight [1..10]</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type (HW, SW, Virtual_HW, Virtual_SW), one enumeration</td>
<td>YES/NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VARIANT_ID</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
To be able to run the ranking example with the IDS and Visual PROMETHEE Academic tools, some data had to be transformed, to be able to handle it.

- Lines of code – the range of values is replaced with the mean value,
- Number of gates per FPGA – range of values is replaced with the mean value,
- Number of subcomponents – Gauss distribution is replaced with the mean value,
- Maintainability, number of services over 10 years – Gauss distribution is replaced with the value representing the maximum number of services with 95% confidence.
The final ranking of all alternatives in IDS tool is shown in the Figure 40. The figure presents the performance scores for all the alternatives on the selected criteria. Because there are some missing data in the assessment information (e.g. memory footprint, number of gates per FPGA), the graph provides a lower and upper bound of the performance scores. Thus, the scores will be in the range marked in grey whatever the missing data turn out to be. As shown in the Figure 40, the most suitable alternative for DM is alternative Hardware1.

Figure 40. The final ranking of alternatives in IDS tool

The final ranking of all alternatives in IDS tool is shown in the Figure 40. The figure presents the performance scores for all the alternatives on the selected criteria. Because there are some missing data in the assessment information (e.g. memory footprint, number of gates per FPGA), the graph provides a lower and upper bound of the performance scores. Thus, the scores will be in the range marked in grey whatever the missing data turn out to be. As shown in the Figure 40, the most suitable alternative for DM is alternative Hardware1.
The final ranking of all alternatives in Visual PROMETHEE Academic tool is shown in Figure 41. The PROMETHEE table displays the Phi, Phi+ and Phi- scores and the most suitable alternative for DM is Hardware1. Visual PROMETHEE Academic tool performs all the pairwise comparisons and whenever a missing value is encountered, the resulting preference degree is set to zero as if both actions had equal evaluations on specific criterion. The alternatives are ranked according to the PROMETHEE II complete ranking.
Classification example – 4eMka2

In this example only one component of the wind turbine is considered. There are 16 classified component alternatives used as the training examples, and four virtual component alternatives are being classified.

To be able to run the classification example with the 4eMka2 tool, some data had to be transformed, to be able to handle it.

- Lines of code – the range of values is replaced with the mean value
- Number of gates per FPGA – the range of values is replaced with the mean value
- Number of subcomponents – Gauss distribution is replaced with the mean value
- Maintainability, number of services over 10 years – Gauss distribution is replaced with the value representing the maximum number of services with 95% confidence

Existing examples were used as training examples for the knowledge extraction. To reduce the time it takes for rules calculation, the length of the rules was set to six. There were 170 rules calculated.

Using rules with strength higher than 60% one component variant was correctly classified as software, and other three components could not be classified.

Figure 42. The classifications results using 4eMka2
The results are presented on the Figure 42, while the ids file with the training data is presented in the Appendix C and the file with the testing data in the Appendix D.

In the classification process it is very important to have very good training examples, which are used for the rules calculation. If the new component is similar to some components that have been previously used, the classification could provide good results, while for components that are significantly different it might be difficult to classify them. For this process it would be very helpful to evaluate the classification decision after at the end of the project to improve the training data and prevent future bad classification results.
Chapter 7

DISCUSSION AND FUTURE WORK

In this chapter we discuss the proposed thesis work with respect to the research questions stated in the introduction.

7.1 Research questions

1. Could MCDA methods and tools be used for HW/SW partitioning?

The MCDA methods and tools could be used for the partitioning process since the major requirements like support for missing values and scalability are satisfied. However the initial idea of being able to handle dependencies among components and being able to create the process that could handle the partitioning for the whole system could not be developed using the existing MCDA tools and methods. Currently the process is limited to only one component and it has to be repeated as many times as many components the system consists of.

   a. What MCDA methods and tools exist and what are their characteristics?

   This work included a systematic research and study in which a number of methods and tools have been identified. The process requirements were defined and the important criteria for evaluating the MCDA methods and tools have been extracted from the requirements. Overall there are 14 methods that have been analysed and its characteristics were put in the table provided in the Appendix A. Overall 14 tools have been analyzed. The table summarizing the MCDA tools with respect to the properties is available in the Appendix B.

   b. What kind of MCDA-based process could enable HW/SW partitioning?

   There are two possible scenarios in the partitioning process. The first one is called the ranking scenario in which alternatives are ranked from the best to the worst. It is used when there are existing component alternatives available.

   The second scenario is called the classification scenario in which the component is newly designed and there are no component alternatives available. In this case the result is the classification of the components as the hardware or software.

   If the system consists of new and existing components, both processes could be used to partition the system, depending on the particular components.
7.2 Future work

This thesis presents the two HW/SW partitioning process, one using ranking and the other using classification method and tool. However there are still many questions and limitations which need further investigations.

The initial idea was to create a MCDA-based process that could provide the partitioning of the whole system at the same time. This could not be done using the existing tools and methods, but it would probably be possible with the implementation of a tool that is customized for this specific problem.

Each alternative or criterion might have associated a reliability/accuracy parameter, indicating how reliable/accurate the values are with respect on how they are achieved (measured, estimated, simulated, etc.). Currently the accuracy and reliability of the provided data is not taken into consideration in the HW/SW partitioning process.

There is also a question of handling ranges of values and distributions. Currently a range of values or a distribution is manually transformed into one value by the DM, for all the criteria. Some criteria might be the best represented by the mean value, while for others the minimum or maximum values could be more important. The customized tool could automatize the process.

The evaluated methods and tools are also not able to provide any feedback about the quality prediction or reliability/accuracy of the solution proposed.

In addition, the current process does not provide the information about decision risks that the DM has to be aware of. Finally, the process should also provide metrics for assessment of the quality of the overall partitioning solution. That could be achieved with the tool that implements the complete partitioning process.
Chapter 8

CONCLUSION

At the start of the thesis the concept of the partitioning process is described in details and the related works regarding partitioning process are presented. Although MCDA methods and tools are in wide use for decades, the research of the scientific papers has not shown any examples of their usage in the HW/SW partitioning process. The concept of MCDA is introduced followed by the systematic research in which a number of MCDA methods and tools have been identified. The HW/SW partitioning process requirements are then proposed. The number of the most important criteria are extracted from the requirements and used for MCDA methods and tools comparison. The final result of comparison is selection of the best tools and methods for the partitioning process. Two possible scenarios in the partitioning process are then proposed followed by the example of how processes work. The first scenario is when there are existing component alternatives available. In this case the ranking process should be used. The second scenario is when the component is newly designed and there are no component alternatives available. In this case the component is defined by its criteria and the decision has to be made whether it is better to implement it as software or hardware. In this case the classification process should be used. If the system consists of new and existing component, both processes could be used on the same system, depending on the particular components. The ranking process could be used for components with existing component alternatives and the classification process for handling the partitioning of new components for which the existing alternatives are not available.

This thesis has shown that the MCDA methods could be used for HW/SW partitioning. It could be used for the partitioning of new components as well as for the reuse of existing components in new systems. However not all partitioning process requirements could be met with the existing tools and methods. Some of these requirements like handling ranges of values and distributions could be fulfilled if the custom tool is created. Other requirements like feedback about the quality prediction or reliability/accuracy of the solution requires further investigations.
Chapter 9

REFERENCES


[38] Ignatius, J., Behzadian, M., Malekan, H. S., & Lalitha, D. (2012, June). Financial performance of Iran’s Automotive sector based on PROMETHEE II. In Management of Innovation and Technology (ICMIT), 2012 IEEE International Conference (pp. 35-38). IEEE.


[78] Jablonsky, J. & Urban, P, MS Excel based system for multicriteria evaluation of alternatives, University of Economics, Department of Econometrics


[82] Intelligent Decision System software, http://php.portals.mbs.ac.uk/Portals/49/docs/IDS50StudentVersion_000.rar, [Online, accessed 02-August-2013]


Appendix A. Methods table

The process requirements described in chapter 5.2 were used to identify the important properties for evaluating the MCDA methods. Overall 14 methods have been analysed and 33 properties have been extracted.

These are the list of properties that have been used for the methods evaluation:

- Type of problem
- Dependence/Interdependence between criteria
- Dependence/Interdependence between alternatives
- Grouping of criteria
- Grouping of alternatives
- Missing values
- Quantitative values
- Qualitative values
- Range of values
- Distribution
- Allow subcriteria
- Allow sensitivity assignment
- Inconsistency handling
- Criteria value type
- Needs for values normalization
- Needs for weight values normalization
- Allow alternative weighting
- Allow criteria weighting
- Criteria weight type
- Mandatory thresholds
- Optional thresholds
- Setting parameters
- Size: Max numbers of criteria
- Size: Max numbers of subcriteria
- Size: Max numbers of alternatives
- Solution finding procedure
- Decision traceability
- Rank reversal problem
- Integration with other method
- Tools available
• Limitations
• Benefits
• Other comments

These are the list of methods that have been evaluated with the respect to the aforementioned properties:

• ELECTRE III
• ELECTRE IV
• ELECTRE TRI
• AHP
• ANP
• VIKOR
• TOPSIS
• PROMETHEE I and II
• DRSA
• Evidential Reasoning
• SMAA- 2
• SMAA- TRI
• RSA
<table>
<thead>
<tr>
<th>Method requirements</th>
<th>ELECTRE III</th>
<th>ELECTRE IV</th>
<th>ELECTRE TRI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Able to handle</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type of problem</td>
<td>Ranking</td>
<td>Ranking</td>
<td>Classification</td>
</tr>
<tr>
<td>Dependence/interdependence between criteria</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Dependence/interdependence between alternatives</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Grouping of criteria</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Grouping of alternatives</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Missing values</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Quantitative values</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Qualitative values</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Range of values</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Distribution</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Allow subcriteria</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Allow sensitivity assignment</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Inconsistency handling</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Input values &amp; Configurability</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Criteria value type</td>
<td>Quantitative</td>
<td>Quantitative</td>
<td>Quantitative</td>
</tr>
<tr>
<td>Needs for values normalization</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Needs for weight values normalization</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Allow alternative weighting</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Allow criteria weighting</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Criteria weight type</td>
<td>Quantitative</td>
<td>Not applicable</td>
<td>Quantitative</td>
</tr>
<tr>
<td>Optional Thresholds</td>
<td>Yes (Indifference and preference thresholds)</td>
<td>Yes (Indifference and preference thresholds)</td>
<td>Yes (Indifference and preference thresholds)</td>
</tr>
<tr>
<td>Setting parameters</td>
<td>No</td>
<td>Yes (Dominance relations)</td>
<td>Yes ((\lambda) - cutting level)</td>
</tr>
<tr>
<td><strong>Scalability &amp; Properties</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size: Max Numbers of criteria</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Size: Max Numbers of subcriteria</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Size: Max Numbers of alternatives</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Solution finding procedure</td>
<td>Outranking flows</td>
<td>Outranking flows</td>
<td>Outranking flows</td>
</tr>
<tr>
<td><strong>Limitations</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decision Traceability</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Rank reversal problem</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Integration with other method</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>List methods</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>General Comments</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tools available</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Limitations</td>
<td>Criteria weights are not supported</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Benefits</td>
<td>Does not need criteria weights</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AHP</td>
<td>ANP</td>
<td>VIKOR</td>
<td>TOPSIS</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>--------</td>
</tr>
<tr>
<td>Ranking</td>
<td>Ranking</td>
<td>Ranking</td>
<td>Ranking</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No more than 10</td>
<td>No more than 10</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>No more than 10</td>
<td>No more than 10</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>No more than 10</td>
<td>No more than 10</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Priority vector</td>
<td>Priority vector</td>
<td>Synergy value Q</td>
<td>By closeness coefficient value</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No more than 10</td>
<td>No more than 10</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>No more than 10</td>
<td>No more than 10</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>No more than 10</td>
<td>No more than 10</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Allows grouping of criteria</td>
<td>Allows grouping of criteria</td>
<td>No tools available</td>
<td>No tools available</td>
</tr>
<tr>
<td>Small amount of documentation available</td>
<td>Small amount of documentation available</td>
<td>Yes (values for certain preference function)</td>
<td>Yes (values for certain preference function)</td>
</tr>
<tr>
<td>DRSA</td>
<td>Evidential Reasoning</td>
<td>SMAA-2</td>
<td>SMAA-TRI</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------</td>
<td>--------</td>
<td>----------</td>
</tr>
<tr>
<td>Classification</td>
<td>Ranking</td>
<td>Classification</td>
<td>Classification</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Quantitative/Qualitative</td>
<td>Qualitative</td>
<td>Quantitative</td>
<td>Quantitative</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Not applicable</td>
<td>Quantitative</td>
<td>Quantitative</td>
<td>Quantitative</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes (Indifference and preference thresholds)</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Rules extraction from existing data</td>
<td>Dempster-Schafer Theory</td>
<td>Weight space exploration</td>
<td>Outranking and weight space exploration</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Partial</td>
<td>Partial</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Not specified</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Limited by the previous experience</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Handles missing values and qualitative data</td>
<td>Able to handle missing values, qualitative values</td>
<td>Handles uncertainties</td>
<td>Handles uncertainties</td>
</tr>
</tbody>
</table>

Evidence-based reasoning process to reach a conclusion.
Appendix B. Tools table

The process requirements defined in chapter 5.2 were used to identify the important properties for evaluating MCDA tools. Overall 14 tools have been analyzed and 39 properties have been extracted.

These are the list of properties that we found as the most interesting for the tools evaluation:

- Description of tool
- Method
- Type of problem
- Tool Version
- Support Retrocompatibility
- Available @
- Change parameters
- Parameters configurable
- Size: Max numbers of criteria
- Size: Max numbers of subcriteria
- Size: Max numbers of alternatives
- Dealing with inconsistent values
- Dealing with missing values
- Allow criteria weighting
- Allow alternatives weighting
- Allow sensitivity assignment
- Decision traceability
- Handle interdependencies between criteria
- Handle interdependencies between alternatives
- Format value(s) for alternatives
- Qualitative/Quantitative/Mixed values for alternatives
- Alternatives values need normalization?
- Format value(s) for criteria
- Qualitative/Quantitative/Mixed values for criteria
- Criteria values need normalization?
- Visualization of results
- Format value(s) of results
- Automatic report generation
- Manual/Help
- Manual/Help comprehensibility
- Tool assistance (Customer support?)
• Integration with other tools/methods
• System requirements
• Licensing solution
• Commercial/free
• Cost
• Limitations
• Benefits
• Other comments

These are the list of tools that have been evaluated with the respect to the aforementioned properties:

• ELECTRE III/ IV
• MakeItRational Professional
• ANP SOLVER
• WEB ANP SOLVER
• Transparent Choice
• Triptych
• SANA
• Visual PROMETHEE Academic
• Intelligent Decision System (IDS)
• JSMAA
• 4eMka2
• jMAF
• ROSE 2
• ELECTRE TRI
<table>
<thead>
<tr>
<th>Tool requirements</th>
<th>ELECTRE III / IV</th>
<th>MakeItRational Professional</th>
<th>ANP SOLVER</th>
<th>WEB ANP SOLVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short Description</td>
<td>Tool for MCDA using Electre III and IV methods</td>
<td>Decision-making tool based on AHP method</td>
<td>Tool that implements the ANP method</td>
<td>Web based tool that implements the ANP method</td>
</tr>
<tr>
<td>Type of problem</td>
<td>Ranking</td>
<td>Ranking</td>
<td>Ranking</td>
<td>Ranking</td>
</tr>
<tr>
<td>Tool Version</td>
<td>3.1b</td>
<td>v.1.0.1</td>
<td>v.1.0.1</td>
<td>v.1.0.1</td>
</tr>
<tr>
<td>Support Retrocompatibility?</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Configurability - Setting</td>
<td>Change parameters</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Parameters configurable</td>
<td>Method, its parameters and precision</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Characteristics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size: Max Numbers of criteria</td>
<td>64</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Size: Max Numbers of subcriteria</td>
<td>Not applicable</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Size: Max Numbers of alternatives</td>
<td>More than 500</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Dealing with Inconsistent Values</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Dealing with Missing Values</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Allow Criteria Weighing</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Allow Alternative Weighting</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Allow sensitivity assignment</td>
<td>Yes</td>
<td>No</td>
<td>Demo: No</td>
<td>No</td>
</tr>
<tr>
<td>Decision Traceability</td>
<td>Yes</td>
<td>No</td>
<td>Credibility, Ranking matrix</td>
<td>Yes</td>
</tr>
<tr>
<td>Handle interdependencies between criteria</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Handle interdependencies between alternatives</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Alternatives</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Format Value(s)</td>
<td>Discrete, Float</td>
<td>Discrete, Float</td>
<td>Discrete, Float</td>
<td>Discrete, Float</td>
</tr>
<tr>
<td>Qualitative/Quantitative/Mixed</td>
<td>Quantitative</td>
<td>Quantitative</td>
<td>Quantitative</td>
<td>Quantitative</td>
</tr>
<tr>
<td>Values Need Normalization</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Criteria - weights</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Format Value(s) of weights</td>
<td>Number</td>
<td>Number</td>
<td>Number</td>
<td>Number</td>
</tr>
<tr>
<td>Qualitative/Quantitative/Mixed</td>
<td>Quantitative</td>
<td>Quantitative</td>
<td>Quantitative</td>
<td>Quantitative</td>
</tr>
<tr>
<td>Values Need Normalization</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Output/Results</td>
<td>Visualization</td>
<td>Graph</td>
<td>Graph</td>
<td>No visualization</td>
</tr>
<tr>
<td>Format Value(s)</td>
<td>Number</td>
<td>Number</td>
<td>Number</td>
<td>Number</td>
</tr>
<tr>
<td>Documentation</td>
<td>Automatic Report Generation</td>
<td>No</td>
<td>Yes/pdf</td>
<td>No</td>
</tr>
<tr>
<td>Manual/Help</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Manual/Help:Comprehensibility</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Tool assistance (Customer support?)</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Integration with other tools/method</td>
<td>No</td>
<td>No</td>
<td>AHP</td>
<td>No</td>
</tr>
<tr>
<td>Tool requirements</td>
<td>Systems requirements</td>
<td>Windows</td>
<td>Windows or Mac on OS with Silverlight 4.0</td>
<td>Windows - Windlenser 3.1 .net framework</td>
</tr>
<tr>
<td>Licensing Solution</td>
<td>Commercial</td>
<td>Commercial</td>
<td>Commercial</td>
<td>Commercial</td>
</tr>
<tr>
<td>Cost</td>
<td>Free</td>
<td>Free</td>
<td>Free</td>
<td>Free</td>
</tr>
<tr>
<td>General Comments</td>
<td>Limitations</td>
<td>Benefit</td>
<td>Other</td>
<td></td>
</tr>
<tr>
<td>Transparent Choice</td>
<td>Triptych</td>
<td>SANA</td>
<td>Visual Promethee Academic</td>
<td>Intelligent Decision System (IDS)</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------</td>
<td>------</td>
<td>---------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Decision-making tool that implements AHP</td>
<td>A tool for MCDA – Excel plugin</td>
<td>MCDA Excel plugin</td>
<td>MCDA tool for Promethee method</td>
<td>General multiple criteria decision analysis and support</td>
</tr>
<tr>
<td>AHP method</td>
<td>Triopsis</td>
<td>Electre I &amp; III, Topsis, Promethee II</td>
<td>Promethee I, Promethee II</td>
<td>Evidence reasoning approach</td>
</tr>
<tr>
<td>Ranking</td>
<td>Ranking</td>
<td>Ranking</td>
<td>Ranking</td>
<td>Ranking</td>
</tr>
<tr>
<td>v.1.0</td>
<td>4.4.47.15.38467</td>
<td>Not specified</td>
<td>1.3.1.0</td>
<td>Student version</td>
</tr>
<tr>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Not specified</td>
<td>Not specified</td>
<td>Not applicable</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Not specified</td>
<td>50</td>
<td>10000</td>
<td>50</td>
<td>Not specified</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>Yes/No</td>
<td>Yes/No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Discrete, Float</td>
<td>Discrete, Float</td>
<td>Discrete, Float</td>
<td>Discrete, Float, Enumerative</td>
<td>Discrete, Float, Enumerative</td>
</tr>
<tr>
<td>Quantitive</td>
<td>Quantitive</td>
<td>Quantitive</td>
<td>Quantitive, Tool quantifies qualitative data</td>
<td>Qualitative, Tool converts quantitative values into</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Number</td>
<td>Number</td>
<td>Number</td>
<td>Number</td>
<td>Number</td>
</tr>
<tr>
<td>Number</td>
<td>Not specified</td>
<td>Numbers</td>
<td>Number</td>
<td>Numbers</td>
</tr>
<tr>
<td>Quantitive</td>
<td>Quantitive</td>
<td>Quantitive</td>
<td>Quantitative</td>
<td>Quantitative</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Graph</td>
<td>Graph</td>
<td>Graph</td>
<td>Graph, tables, etc.</td>
<td>Graph</td>
</tr>
<tr>
<td>Number</td>
<td>Number</td>
<td>Number</td>
<td>Number</td>
<td>Numbers</td>
</tr>
<tr>
<td>Yes/Excel</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Web browser</td>
<td>Windows, Excel</td>
<td>Windows, Excel</td>
<td>Windows</td>
<td>Windows</td>
</tr>
<tr>
<td>Commercial</td>
<td>Annual, permanent</td>
<td>Not specified</td>
<td>License for Business Edition</td>
<td>Not specified</td>
</tr>
<tr>
<td>Commercial (Free trial 30 days)</td>
<td>Not specified</td>
<td>Academic version is free, business version is</td>
<td>1250 EURO</td>
<td>Not specified</td>
</tr>
<tr>
<td>Monthly 20€, 1 year-178/month, 3 years-123/month</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>v.1.0</td>
<td>4.4.47.15.38467</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes/No</td>
<td>Yes/No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>More than one method supported</td>
<td>Only supported on Windows platform, Linux</td>
<td>AHP method could be used for criteria weighting</td>
<td>Help is only in Czech language</td>
<td>It seems that values normalization is not supported</td>
</tr>
<tr>
<td>Feature</td>
<td>SMAA</td>
<td>IMAF</td>
<td>JMAF</td>
<td>ELECTRE TRI</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Ranking, Classification</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Consistency level parameter, type of rules</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Number of reducts, type of validation test, etc.</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Supports distributions</td>
<td>Discrete, Float, Interval, Gaussian distribution</td>
<td>Discrete, Float, Qualitative</td>
<td>Discrete, Float, Qualitative</td>
<td>Discrete, Float</td>
</tr>
<tr>
<td>Can handle mixed data types</td>
<td>Mixed</td>
<td>Mixed</td>
<td>Mixed</td>
<td>Quantitative</td>
</tr>
<tr>
<td>Interface</td>
<td>Table</td>
<td>Graph</td>
<td>Table</td>
<td>Graph</td>
</tr>
<tr>
<td>Number of examples</td>
<td>&gt;1000</td>
<td>Not specified</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Users</td>
<td>Yes (optional)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Requires manual calculation</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Requires manual interpretation</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Free</td>
<td>Yes (optional)</td>
<td>Only for non-profit purposes</td>
<td>Free</td>
<td>Free</td>
</tr>
<tr>
<td>Main development language</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows</td>
<td>Windows</td>
<td>Windows</td>
<td>Windows</td>
</tr>
<tr>
<td>Licensing</td>
<td>GPL3</td>
<td>Free</td>
<td>Free</td>
<td>Free</td>
</tr>
<tr>
<td>Additional Features</td>
<td>No</td>
<td>Only with Consistency Dominance</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- SMAA: SMAA methods in Java
- IMAF: Implementing SMAA method
- JMAF: SMAA for RSA method
- ELECTRE TRI: MCDA tool implementing Electre TRI method

Additional features:
- Open source (GPL3) for SMAA
- Free for non-profit purposes for IMAF
- Free for SMAA
- Only for non-profit purposes for JMAF
- Free for ELECTRE TRI

Challenges:
- Unresponsive while entering a large problem
- Unable to handle missing values
- Supports distributions

For more details, visit:
- [http://smaa.fi/jsmaa/](http://smaa.fi/jsmaa/)
- [http://idss.cs.put.poznan.pl/site/60.html#c80](http://idss.cs.put.poznan.pl/site/60.html#c80)
Appendix C. 4eMka2 training examples

The "isf" file that is used for the rules calculation by the 4eMka2 tool is presented. It consists of 30 criteria and 16 training examples. In the first part of the document, the "attributes" part, the decision criteria are defined. The criterion is defined by the plus sign that indicates that the criterion is going to be used or minus sign otherwise. This is followed by the criterion name and type. The criteria could be quantitative or qualitative. For the quantitative criteria "continuous" is put in the brackets and for the qualitative all the options from the worst to the best is put in the square brackets. At the end of the first part of the document is defined which criterion specifies the classes of the final result. In the second part of the document, the criteria preference is defined. If the preference increases with higher values the criterion is defined as gain criterion. Otherwise the criterion is defined as cost. In the third part of the document are the training examples. One row of the document corresponds to one training example. The data values are separated with tabs and arranged in the same order as defined in the first and the second part of the document. All the missing data values are replaced with the question mark sign.

**ATTRIBUTES
+ hwPlatform: [Low, Indifferent, High]
+ swPlatform: [Low, Indifferent, High]
+ implementationFormat: [Low, High]
+ releaseYear: (continuous)
+ statusActive: [No, Yes]
+ variantPreference: [Low, Indifferent, High]
+ p2pTime: (continuous)
+ wcet: (continuous)
+ memoryFootprint: (continuous)
+ linesOfCode: (continuous)
+ gatesPerFPGA: (continuous)
+ powerConsumption: (continuous)
+ modularity: (continuous)
+ complexity: [Low, Medium, High]
+ timeToRework: (continuous)
+ timeToDesign: (continuous)
+ timeToImplement: (continuous)
+ timeToTest: (continuous)
+ timeToMaintain: (continuous)
+ weekOverDesignDeadline: (continuous)
+ weekOverImplementationDeadline: (continuous)
+ weekOverTestDeadline: (continuous)
+ costToBuy: (continuous)
+ costToTest: (continuous)
+ costToMaintain: (continuous)
+ designerConfidence: [Low, Medium, High]
+ maxExecutionTime: (continuous)
+ portability: [Low, Medium, High]
+ upgradeability: (continuous)
+ maintainability: (continuous)
+ State: [SW, HW]
decision: State

**PREFERENCES
hwPlatform: gain
swPlatform: gain
implementationFormat: gain
releaseYear: gain
statusActive: gain
variantPreference: gain
p2pTime: cost
wcet: cost
memoryFootprint: cost
gatesPerFPGA: cost
powerConsumption: cost
modularity: cost
complexity: cost
timeToRework: cost
timeToDesign: cost
timeToImplement: cost
timeToTest: cost
weekOverDesignDeadline: cost
weekOverImplementationDeadline: cost
weekOverTestDeadline: cost
costToBuy: cost
costToTest: cost
costToMaintain: cost
designerConfidence: gain
maxExecutionTime: cost
portability: gain
upgradeability: gain
maintainability: cost
State: gain

**EXAMPLES**

<p>| | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>2009</td>
<td>Yes</td>
<td>Low</td>
<td>12</td>
<td>20</td>
<td>700</td>
<td>250</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>300.67</td>
<td>10</td>
<td>Low</td>
<td>40</td>
<td>5</td>
<td>15</td>
<td>10</td>
<td>15</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>150</td>
<td>Low</td>
<td>10</td>
<td>High</td>
<td>30</td>
<td>6.163</td>
<td>SW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>High</td>
<td>2009</td>
<td>Yes</td>
<td>High</td>
<td>12</td>
<td>20</td>
<td>700</td>
<td>250</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>300.67</td>
<td>10</td>
<td>Low</td>
<td>40</td>
<td>5</td>
<td>15</td>
<td>10</td>
<td>15</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>150</td>
<td>Low</td>
<td>10</td>
<td>High</td>
<td>30</td>
<td>6.163</td>
<td>SW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indifferent</td>
<td>High</td>
<td>High</td>
<td>2009</td>
<td>Yes</td>
<td>High</td>
<td>12</td>
<td>13</td>
<td>700</td>
<td>250</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>300.67</td>
<td>10</td>
<td>Low</td>
<td>40</td>
<td>5</td>
<td>15</td>
<td>10</td>
<td>15</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>150</td>
<td>Low</td>
<td>10</td>
<td>High</td>
<td>30</td>
<td>6.163</td>
<td>SW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Low</td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>50</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>146</td>
<td>High</td>
<td>5</td>
<td>Low</td>
<td>10</td>
<td>6.163</td>
<td>HW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Low</td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>50</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>146</td>
<td>High</td>
<td>5</td>
<td>Low</td>
<td>10</td>
<td>6.163</td>
<td>HW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>High</td>
<td>2013</td>
<td>Yes</td>
<td>Low</td>
<td>8</td>
<td>13</td>
<td>800</td>
<td>250</td>
<td>?</td>
<td>250</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Medium 35</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>15</td>
<td>-3.00</td>
<td>-3.00</td>
<td>-3.00</td>
<td>170</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>Medium 20</td>
<td>High</td>
<td>40</td>
<td>6.163</td>
<td>SW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indifferent</td>
<td>Indifferent</td>
<td>High</td>
<td>2012</td>
<td>No</td>
<td>Indifferent</td>
<td>9</td>
<td>17</td>
<td>600</td>
<td>250</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td>300</td>
<td>10</td>
<td>Medium 40</td>
<td>5</td>
<td>15</td>
<td>10</td>
<td>15</td>
<td>-2.00</td>
<td>-2.00</td>
<td>-2.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>15</td>
<td>130</td>
<td>Medium 20</td>
<td>High</td>
<td>40</td>
<td>6.163</td>
<td>HW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>2011</td>
<td>Yes</td>
<td>High</td>
<td>10</td>
<td>18</td>
<td>700</td>
<td>350</td>
<td>?</td>
<td>487</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>High</td>
<td>40</td>
<td>5</td>
<td>15</td>
<td>10</td>
<td>15</td>
<td>-1.00</td>
<td>-1.00</td>
<td>-1.00</td>
<td>150</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>Medium 20</td>
<td>Medium 40</td>
<td>6.163</td>
<td>SW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>2009</td>
<td>Yes</td>
<td>High</td>
<td>5</td>
<td>10</td>
<td>?</td>
<td>150</td>
<td>150</td>
<td>50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Medium 50</td>
<td>5</td>
<td>15</td>
<td>10</td>
<td>20</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>200</td>
<td>43</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>Medium 5</td>
<td>Low</td>
<td>5</td>
<td>6.163</td>
<td>HW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>2006</td>
<td>No</td>
<td>Low</td>
<td>3.5</td>
<td>11</td>
<td>?</td>
<td>150</td>
<td>250</td>
<td>50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Medium 50</td>
<td>5</td>
<td>15</td>
<td>10</td>
<td>20</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>200</td>
<td>43</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>Medium 5</td>
<td>Low</td>
<td>5</td>
<td>6.163</td>
<td>HW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>2011</td>
<td>Yes</td>
<td>High</td>
<td>10</td>
<td>18</td>
<td>800</td>
<td>350</td>
<td>?</td>
<td>310</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>-----</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>------</td>
<td>-----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Medium 40</td>
<td>10</td>
<td>19</td>
<td>5</td>
<td>15</td>
<td>-1.00</td>
<td>0.00</td>
<td>2.00</td>
<td>200</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>230</td>
<td>Medium 15</td>
<td>Low</td>
<td>30</td>
<td>2.520</td>
<td>SW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indifferent</td>
<td>Indifferent</td>
<td>High</td>
<td>2012</td>
<td>Yes</td>
<td>High</td>
<td>9</td>
<td>17</td>
<td>600</td>
<td>300</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>487</td>
<td>10</td>
<td>Medium 30</td>
<td>10</td>
<td>18</td>
<td>5</td>
<td>10</td>
<td>-1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>160</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>230</td>
<td>Medium 15</td>
<td>Medium 40</td>
<td>5.040</td>
<td>SW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>High</td>
<td>2013</td>
<td>Yes</td>
<td>Indifferent</td>
<td>8</td>
<td>13</td>
<td>500</td>
<td>250</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>5</td>
<td>Low</td>
<td>20</td>
<td>10</td>
<td>17</td>
<td>5</td>
<td>15</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>200</td>
<td>High</td>
<td>10</td>
<td>High</td>
<td>50</td>
<td>2.164</td>
<td>SW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>High</td>
<td>2008</td>
<td>Yes</td>
<td>Indifferent</td>
<td>2</td>
<td>8</td>
<td>?</td>
<td>150</td>
<td>150</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>5</td>
<td>Low</td>
<td>30</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>20</td>
<td>-1.00</td>
<td>0.00</td>
<td>-1.00</td>
<td>150</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>173</td>
<td>High</td>
<td>5</td>
<td>High</td>
<td>15</td>
<td>6.163</td>
<td>HW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>Indifferent</td>
<td>Low</td>
<td>2007</td>
<td>No</td>
<td>Low</td>
<td>3</td>
<td>9</td>
<td>?</td>
<td>225</td>
<td>275</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>7</td>
<td>Medium 40</td>
<td>10</td>
<td>20</td>
<td>5</td>
<td>25</td>
<td>0.00</td>
<td>0.00</td>
<td>3.00</td>
<td>250</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>193</td>
<td>Medium 10</td>
<td>Medium 10</td>
<td>8.376</td>
<td>HW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indifferent</td>
<td>Low</td>
<td>Low</td>
<td>2006</td>
<td>No</td>
<td>Low</td>
<td>5</td>
<td>10</td>
<td>?</td>
<td>200</td>
<td>375</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>High</td>
<td>50</td>
<td>15</td>
<td>25</td>
<td>5</td>
<td>30</td>
<td>1.00</td>
<td>0.00</td>
<td>4.00</td>
<td>300</td>
<td>63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>203</td>
<td>Low</td>
<td>15</td>
<td>Low</td>
<td>5</td>
<td>12.849</td>
<td>HW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**END**
Appendix D. 4eMka2 testing examples

The "isf" file that consists of four examples that are being classified by the 4eMka2 tool is presented. As well as the training examples, testing examples also have 30 criteria. The document has the same structure as the document with the training examples. The criteria are defined in the first part of the document, followed by the criteria preference in the second part while the data values are in the final part of the document.

**ATTRIBUTES
+ hwPlatform: [Low, Indifferent, High]
+ swPlatform: [Low, Indifferent, High]
+ implementationFormat: [Low, High]
+ releaseYear: (continuous)
+ statusActive: [No, Yes]
+ variantPreference: [Low, Indifferent, High]
+ p2pTime: (continuous)
+ wcet: (continuous)
+ memoryFootprint: (continuous)
+ linesOfCode: (continuous)
+ gatesPerFPGA: (continuous)
+ powerConsumption: (continuous)
+ modularity: (continuous)
+ complexity: [Low, Medium, High]
+ timeToRework: (continuous)
+ timeToDesign: (continuous)
+ timeToImplement: (continuous)
+ timeToTest: (continuous)
+ timeToMaintain: (continuous)
+ weekOverDesignDeadline: (continuous)
+ weekOverImplementationDeadline: (continuous)
+ weekOverTestDeadline: (continuous)
+ costToBuy: (continuous)
+ costToTest: (continuous)
+ costToMaintain: (continuous)
+ designerConfidence: [Low, Medium, High]
+ maxExecutionTime: (continuous)
+ portability: [Low, Medium, High]
+ upgradeability: (continuous)
+ maintainability: (continuous)
+ State: [SW, HW]

decision: State

**PREFERENCES
hwPlatform: gain
swPlatform: gain
implementationFormat: gain
releaseYear: gain
statusActive: gain
variantPreference: gain
p2pTime: cost
wcet: cost
memoryFootprint: cost
linesOfCode: cost
gatesPerFPGA: cost
powerConsumption: cost
modularity: cost
complexity: cost
timeToRework: cost
timeToDesign: cost
timeToImplement: cost
timeToTest: cost
timeToMaintain: cost
weekOverDesignDeadline: cost
weekOverImplementationDeadline: cost
weekOverTestDeadline: cost
costToBuy: cost
costToTest: cost
costToMaintain: cost
designerConfidence: gain
maxExecutionTime: cost
portability: gain
upgradeability: gain
maintainability: cost
State: gain

**EXAMPLES

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>?</td>
<td>20</td>
<td>20</td>
<td>15</td>
<td>20</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>150</td>
<td>11</td>
<td>155</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>10</td>
<td>Medium</td>
<td>30</td>
<td>8.163</td>
<td>SW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>?</td>
<td>130</td>
<td>15</td>
<td>15</td>
<td>10</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>50</td>
<td>33</td>
<td>130</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>5</td>
<td>Low</td>
<td>10</td>
<td>9.163</td>
<td>HW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>Low</td>
<td>?</td>
<td>?</td>
<td>Low</td>
<td>26</td>
<td>18</td>
<td>100</td>
<td>475</td>
<td>?</td>
<td>300.67</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>10</td>
<td>40</td>
<td>?</td>
<td>5</td>
<td>14</td>
<td>-1</td>
<td>7</td>
<td>7</td>
<td>100</td>
<td>?</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>14</td>
<td>High</td>
<td>30</td>
<td>10.52014839</td>
<td>SW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>High</td>
<td>?</td>
<td>?</td>
<td>High</td>
<td>26</td>
<td>6</td>
<td>?</td>
<td>?</td>
<td>150</td>
<td>100</td>
<td>5</td>
</tr>
<tr>
<td>Medium</td>
<td>5</td>
<td>80</td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>150</td>
<td>?</td>
<td>125</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>5</td>
<td>Low</td>
<td>20</td>
<td>5.040296776</td>
<td>HW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**END