

University of Manchester
School of Computer Science

Enhancing a Decision Support Tool with Sensitivity Analysis

Project Background Report

Student Name: Renzo Bertuzzi Leonelli

Programme: ACS and IT Management

Supervisor: John Keane

2012

Abstract

With the increased speed and complexity of today's world, and the ever increasing amount of data available, a decision support system is of critical importance.

Multi-Criteria Decision Making (MCDM) enables decision makers to make a decision considering several alternatives and multiple criteria of evaluation. The *Analytic Hierarchy Process* (AHP) is an area of research in MCDM where the problem is decomposed into a hierarchic model in order to rank the alternatives. *Sensitivity analysis* (SA) is a technique that determines the effects of changes in input values on a model results, hence, performing a SA on the results of a decision problem may provide valuable information to the decision maker about the robustness of the solution.

This report investigates various methods to carry out SA on AHP models. In addition, a framework for evaluating software in terms of AHP features and SA support is developed and a number of commercial and academic software tools are analysed using this framework. The result from this analysis shows that despite the existence of several approaches to performing SA, available software tools only provide a basic form of SA. As a consequence, an appropriate tool, selected based on the framework analysis, is to be enhanced to include an SA module. The implementation of additional features and improvements involves a re-engineering process of the AHP tool. As a result of this re-engineering, the enhanced tool will also be made available on web and mobile platforms. Finally, the methodology for the project is introduced, the approach for the design and re-engineering of the software is described, and the evaluation, current status and project plan are presented.

Contents

Abstract	ii
List of Tables	iv
List of Figures	iv
Glossary	v
1 Introduction	1
1.1 Project Description.....	2
1.2 Aims and Objectives.....	2
1.3 Report Outline.....	2
2 Background	3
2.1 Multi-Criteria Decision Making.....	3
2.2 Analytic Hierarchy Process.....	4
2.3 Pairwise Comparison.....	5
2.4 Sensitivity Analysis.....	7
2.4.1 Numerical Incremental Analysis.....	7
2.4.2 Probabilistic Simulations.....	8
2.4.3 Mathematical Models.....	10
2.5 Requirements for Evaluating AHP Software.....	12
2.5.1 Support for AHP and PC.....	12
2.5.2 Support for Sensitivity Analysis.....	13
2.6 Software Products.....	13
2.6.1 Expert Choice.....	13
2.6.2 MakeltRational.....	14
2.6.3 PriEsT.....	15
2.6.4 Web-HIPRE.....	15
3 Methodology	17
3.1 Software Methodology.....	17
3.2 Software Design and Implementation.....	17
3.2.1 Re-engineering PriEsT.....	17
3.2.2 Module for Sensitivity Analysis.....	18
3.3 Project Evaluation.....	18
3.3.1 Evaluation of Sensitivity Analysis.....	19
3.3.2 Evaluation of platform-specific versions.....	19
3.4 Project Progress.....	19
3.5 Project Plan.....	20
References	21
Appendix A	24

List of Tables

Table 2.1. Scale of Relative Importance [14].....	6
Table 2.2. Expert Choice Software Evaluation.....	14
Table 2.3. MakeltRational Software Evaluation.....	15
Table 2.4. PriEsT Software Evaluation.....	15
Table 2.5. Web-HIPRE Software Evaluation.....	16

List of Figures

Figure 2.1. A hierarchy model in AHP.....	5
Figure 2.2. Numerical Incremental Sensitivity Analysis.....	8
Figure 2.3. Box-plot Chart Presenting Simulation Results [33].....	9
Figure 2.4. Ranking of Alternatives in the Weight Space [5].....	11
Figure 3.1. Project Gantt Chart.....	20
Figure 3.2. UML Class Diagram for AHP Library.....	24

Glossary

AHP	Analytic Hierarchy Process
CI	Consistency Index
CR	Consistency Ratio
DM	Decision Maker
IAHP	Interval AHP
J2ME	Java 2 Micro Edition
JVM	Java Virtual Machine
MCDM	Multi-Criteria Decision Making
MIDlet	MIDP Application
MIDP	Mobile Information Device Profile
MVC	Model View Controller
NV	Number of Violations
PC	Pairwise Comparison
SA	Sensitivity Analysis
TD	Total Deviation
TDD	Test Driven Development
WSM	Weighted Sum Model

1 Introduction

Everyone has to make decisions, all the time. Some decisions are very simple and have such a small impact that can be taken without much analysis. In contrast, there are complex decisions with a significant impact that have to be taken with as much information, analysis and reflection as possible to decide on the possible alternatives. Managerial decisions may have a significant impact on the performance and success of companies. For example, Campbell et al. [1] and Janis [2] have identified examples where clearly very wrong decisions have been made. With the increased speed and complexity of today's world and the ever increasing amount of data available, decision support is of critical importance.

Decision making is a structured process that formalises the steps involved in making decisions. *Multi-Criteria Decision Making* (MCDM) is a discipline that enables decision makers to make a decision considering several alternatives and multiple criteria of evaluation. MCDM has become so attractive that several books, thousands of articles and many scientific journals are dedicated to the subject [3].

The *Analytic Hierarchy Process* (AHP) [4] is an area of research in MCDM where the problem is decomposed into a hierarchic model, represented as a tree. The top element of the hierarchy represents the overall goal, intermediate nodes correspond to the different criteria and the leaves of the hierarchy tree are the alternatives. The relative importance of the alternatives and criteria is assessed by using the Pairwise Comparison (PC) method. Once all nodes in the hierarchy tree are evaluated, then the aggregated importance of the alternatives is calculated and a solution or ranking is obtained, known as *preference vector*.

To assess the importance of alternatives and criteria using PC, only two elements are compared at a time. The value assigned to this comparison, or judgement, may have an objective or subjective origin. In either way, the judgements represent the direction of preference between the two elements and the strength of this relation. For each criterion a matrix is formed with the judgements and an elicitation method is used to obtain the preference vector for this matrix containing the weights of each element.

Often, decision makers (DM) are not completely confident about judgements they assign to each comparison; the judgements may be too subjective or may come from a group decision where members may have different opinions about each criterion [5]. In such cases, it is desirable to run a sensitivity analysis (SA) on the results to analyse how sensitive the solution is to changes in input data. SA is a technique that determines the effects of changes in input values on a model results. In this way, SA is a powerful tool to assess how the preference vector is affected by changes in the PC judgements or in the weights of the elements. SA may be useful in providing information equally important to the solution of the problem [6] such as analysis of the robustness of the recommended solution, identification of critical elements in the model and help to answer "what if" questions.

Several software tools provide support for both AHP and some level of SA; some have been in the market for many years, e.g. Web-HIPRE [7], while others have been developed recently, e.g. MakeltRational [8]. Given the importance and popularity of the AHP field and the usefulness of SA, various approaches in the literature for performing SA will be analysed. Based on these analyses, a framework of desirable features for AHP software will be constructed and various software tools will be assessed according to this framework. The objective of this framework is to assess the potential of AHP software to deliver a set of important features necessary to provide a satisfactory solution to a decision problem, and identify potential improvements that may help the DM to make more robust decisions.

The framework has identified *PriEsT* (Priority Estimation Tool), an AHP software tool developed at the University of Manchester ([9], [10]) available for Linux and Windows, as the best current offering despite it not offering support for SA. To implement additional SA

features in PriEsT, a process of re-engineering must be undertaken to analyse the application and to improve the modularity, code readability and the time needed to add new functionalities [11]. As a consequence of this re-engineering, the enhanced PriEsT tool will also be made available on web and mobile platforms.

1.1 Project Description

This project will:

1. Investigate how to perform sensitivity analysis and the different methods available in the literature.
2. Perform a survey of the software tools that support AHP and SA, and develop a framework that enables comparison in terms of desirable features or requirements.
3. Select an appropriate MCDM tool for development – PriEsT has been the selected.
4. Improve PriEsT with support for additional desirable features (SA and web and mobile deployment).

1.2 Aims and Objectives

The general aim of the project is to investigate methods to perform SA in AHP. Following this, the MCDM AHP tool PriEsT will be analysed, re-engineered as appropriate and enhanced by including a module for SA. As a consequence of this re-engineering various further enhancements will be identified and developed, such as additional platform availability.

To accomplish the goal of the project the following objectives are defined:

- Explore AHP and SA methods. Review the literature to identify the existing methods to perform SA.
- Re-engineer and re-structure PriEsT in a modular way. Create a library with the AHP functionalities and SA.
- Following this re-engineering further enhancements that will provide additional platform availability will be developed:
 - a web version of PriEsT as a Java Applet.
 - a mobile version of PriEsT as a J2ME MIDlet¹.

1.3 Report Outline

The rest of the report is structured as follows:

Chapter 2 - Background: provide a literature survey of relevant topics. Firstly, a study of MCDM, AHP and Pairwise Comparison is presented; then, different approaches to providing SA are identified; the chapter ends with a list of desirable features for an AHP software tool and a framework for comparison of existing products is given.

Chapter 3 - Methodology: this chapter presents the methodology of the project. Then, the requirements and design for the software tool to be implemented are laid out. Next, the methods for evaluating the project are described. Finally, current progress and a development plan are provided.

1 A MIDlet is an application written in Java Micro Edition targeted to run on mobile devices using the MIDP profile.

2 Background

This chapter presents the relevant background material. First, a short introduction to the Multi-Criteria Decision Making field is given. Then, the Analytic Hierarchy Process and the Pairwise Comparison methods are described. Next, different techniques for performing sensitivity analysis are discussed. Finally, desirable features of AHP software and existing tools are considered.

2.1 Multi-Criteria Decision Making

Multi-Criteria Decision Making (MCDM) is an area of Operational Research that supports the process of decision making. In MCDM, the goal is to rank different alternatives considering multiple, often conflicting, criteria. For example, consider buying a new car, some of the criteria to consider and evaluate are cost, fuel consumption, safety, capacity and style. After evaluating a list of possible cars against these criteria, a ranking of cars can be obtained and the most appropriate choice can be selected.

For an alternative to be judged by a criterion, a scale of possible values must be defined for the criterion. A scale is defined by the direction (cost or benefit) and the magnitude of the values. In addition, different types of criteria may be used including measurable, ordinal, probabilistic or fuzzy criteria [12].

To find the highest scoring alternative, the DM must evaluate all possible choices against each criterion. Then a prioritisation method is applied to aggregate all judgements and create a ranking of the alternatives. Finally, the DM uses this information as a recommendation to select one of the alternatives according to his/her requirements and preferences.

The MCDM methodology is a process that includes four main steps beginning with the definition of the decision problem and ending with the selection of an alternative [12]. The steps are as follows:

1. **Structuring the decision problem:** the DM establishes the problem and the set of possible alternatives to consider.
2. **Modelling the criteria:** the criteria and the way to measure the alternatives for each criterion are defined.
3. **Aggregating the preferences:** each alternative is judged against each criterion and an aggregation method is used to derive the ranking of alternatives.
4. **Recommendations:** recommendations are given to the DM based on the results from the previous step. The DM selects one of the alternatives.

There exist several prioritisation methods² [13] that aggregate the preferences in step 3 and different methods may yield different results. Although studies [18] have been carried out to compare different methods and to provide a framework for selecting the most appropriate one depending on the problem, according to a study from Wallenius et al. [20] in 2008, the most popular method in the literature since the 1970s is the Analytical Hierarchy Process (AHP). The AHP method is described in the next section.

² The most popular methods are: weighted sum model (WSM), weighted product model (WPM), analytic hierarchy process (AHP) [14], ELECTRE [15], TOPSIS [16], SMART [17], PROMETHEE [19], multi attribute utility theory (MAUT) [20], UTA (UTilités Additives) [21]

2.2 Analytic Hierarchy Process

AHP is a method of prioritisation that enable DMs to evaluate the relative importance of objective and subjective alternatives and criteria by using the pairwise comparison technique. It was introduced by Saaty in 1980 [14] and has proven to be very popular worldwide since its creation [20].

To apply the AHP method a decision problem must be decomposed into four steps [4]:

1. Define the problem and determine the outcome sought
2. Structure the problem as a hierarchy, where the top element is the goal of the decision. The intermediate levels define the criteria on which the set of alternatives in the lowest level will be judged.
3. Construct one PC matrix for every non-leaf node in the hierarchy and get the priority vector from each matrix. Each element in an upper level is used to evaluate its child elements with respect to it.
4. Aggregate the priorities obtained from the PC matrices. The priorities in one node are used to weight the priorities in the level below and then are added to obtain the global priority. This method is known as the Weighted Sum Model (WSM). The alternative with the highest global priority is considered to be the best choice.

For instance, consider a decision problem with M alternatives A_i (for $i=1...M$) and N criteria C_j (for $j=1...N$). Let w_j be the weight of criterion C_j and $\delta_{i,j}$ be the performance value of alternative A_i for criterion C_j and P_i be the overall priority of alternative A_i .

AHP use normalised weights, that is, the sum of weights must be 1, as shown in formula (1).

$$\sum_{i=1}^M \delta_{i,j} = 1, \quad \sum_{j=1}^N w_j = 1, \quad \sum_{i=1}^M P_i = 1 \quad (1)$$

The overall priorities P_i of the alternatives are calculated using the WSM as shown in formula (2):

$$P_i = \sum_j \delta_{i,j} w_j, \quad \text{for } i=1, \dots, M \quad (2)$$

For simplicity only one level of criteria is presented. For more than one level, formula (2) is applied at every level of the hierarchy. For example, in Figure 2.1 a hierarchy with two levels of criteria and three alternatives is presented. To obtain the overall priority for alternative A_1 , firstly the local priorities for criterion C_1 and C_2 have to be calculated. Formula (2) is applied in node C_1 using the weights of C_{11} and C_{12} and the performance value of A_i for these two criteria to get the local priority in criterion C_1 :

$$P_{1,1} = \delta_{1,11} w_{11} + \delta_{1,12} w_{12} \quad (3)$$

For criterion C_2 the local priority $P_{1,2}$ is the same as $\delta_{1,2}$ as there is no intermediate nodes. Next, formula (2) is again applied to calculate the global priority:

$$P_1 = P_{1,1} w_1 + P_{1,2} w_2 \quad (4)$$

Using this method any hierarchy, independent from the number of levels, can be analysed and global priorities can be obtained.

To assess the alternatives against each criterion a scale of dimensionless numbers is used.

That is, even though the criterion may be referring to cost or time, when the PC method is used the units of measure become abstract units and in this way the aggregation is possible for different scales.

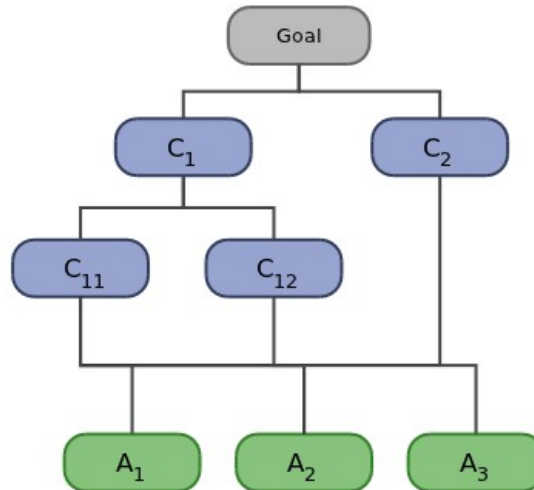


Figure 2.1. A hierarchy model in AHP

Belton and Gear [22] found that when adding an alternative that is identical or similar to an existing one, the ranking of options may be reversed. To prevent this, they developed the *Ideal Mode AHP*. In this mode, instead of having the performance values of the alternatives for a given criterion sum to 1, each value is divided by the maximum value in the vector so the resulting vector has a maximum value of 1 instead of the sum being 1.

2.3 Pairwise Comparison

To assess the alternatives under a given criterion, it is often very hard for DMs to assign an absolute score. Qualitative and quantitative data may be unavailable or necessary information to quantify the performance of alternatives may be incomplete. Therefore, the PC method is used to determine the relative importance or weights of the alternatives and criteria with respect to each criterion in the decision problem.

Under this approach, the DM has to analyse only two elements at a time. To make this comparison, the DM has to choose a value indicating how many times more important, preferred or dominant one element is over another element in terms of a given criterion. This value has to be given in reference to a predefined scale. Saaty [14] proposed a ratio scale from 1 to 9 (see Table 2.1). Other scales have been proposed; Triantaphyllou et al. [23] evaluated 78 different scales and concluded that there is no scale that outperforms all other scales.

<i>Intensity of Importance</i>	<i>Definition</i>	<i>Explanation</i>
1	Equal Importance	Two activities contribute equally to the objective
3	Moderate importance	Experience and judgement slightly favour one activity over another
5	Strong importance	Experience and judgement strongly favour one activity over

Intensity of Importance	Definition	Explanation
		another
7	Very strong importance	An activity is favoured very strongly over another; its dominance demonstrated in practice
9	Extreme importance	The evidence favouring one activity over another is of the highest possible order of affirmation
2,4,6,8	Intermediate values	
1.1-1.9	If the activities are very close	May be difficult to assign the best value but when compared with other contrasting activities the size of the small numbers would not be too noticeable, yet they can still indicate the relative importance of the activities

Table 2.1. Scale of Relative Importance [14]

When comparing elements A_i and A_j , A_i is said to be a_{ij} times more important than A_j , and the reciprocal value of the judgement is used to score the inverse comparison, that is, A_j is a_{ji} times more important than A_i . The self-comparison a_{ii} is always scored 1 (See formula 5).

$$a_{ij} = \frac{1}{a_{ji}}, \quad a_{ii} = 1 \quad (5)$$

All judgements of the elements with respect to a given criterion are organised in a PC matrix (PCM). For n elements, the matrix is of size $n \times n$, and because of the constraints in formula (5), only $n(n-1)/2$ elements are provided by the DM.

$$A = \begin{bmatrix} 1 & a_{12} & a_{13} & \cdots & a_{1n} \\ 1/a_{12} & 1 & 1/a_{23} & \cdots & a_{2n} \\ 1/a_{13} & 1/a_{23} & 1 & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1/a_{1n} & 1/a_{2n} & 1/a_{3n} & \cdots & 1 \end{bmatrix} \quad (6)$$

There exists the possibility that the DM may provide inconsistent judgements. For instance, if $a_{12}=3$, $a_{23}=2$, then we would expect that $a_{13}=6$. However, this is rarely the case. A PC matrix is said to be consistent or acceptable if its corresponding Consistency Ratio (CR³) is less than 0.1 [14]. There exist two types of consistency [10], cardinal consistency (CC) and ordinal consistency (OC). For a matrix to be cardinally consistent, the condition $a_{ij}=a_{ik}a_{kj}$ must hold true for all i, j, k . The ordinal consistency refers to the transitivity of the preferences, if alternative A_i is preferred over A_j , and A_j is preferred over A_k , then A_i should be preferred over A_k . When this condition is not met, the matrix is said to be ordinally inconsistent and it is not possible to find a weight vector that satisfies all preferences directions and priority violations will occur.

To derive the weights or priorities from the PC matrix different methods may be used. For consistent matrices, generally all methods yield similar results only differing in intensities. Siraj [10] analysed several methods and proposed two new methods based on graph theory and multi-objective optimisation. Multi-objective optimisation outperformed all other methods

3 Also known as the Consistency Index (CI)

for inconsistent matrices. Based on their simplicity, however, the most common methods used to elicit priorities in AHP are Eigenvector (EV), Geometric Mean (GM) and Normalised Column Sum (NCS). A study by Choo and Wedley [24] evaluated 18 different methods and recommended GM and NCS for their easy calculation and good performance on consistent and inconsistent matrices.

A number of techniques exist to estimate the performance of a prioritisation method. The most common approach is calculating the total deviation (TD) or quadratic error between the priority vector and the provided judgements, as shown in formula (7).

$$TD = \sum_{i=1}^n \left(\sum_{j=1}^n \left(a_{ij} - \frac{w_i}{w_j} \right)^2 \right)^{\frac{1}{2}} \quad (7)$$

Another approach is to calculate the number of priority violations (NV). If the alternative A_i is preferred over A_j but the derived priorities are such that $w_j > w_i$, then a priority violation is present [25]. The formula (8) to calculate NV is as follows:

$$NV = \sum_{i=1}^n \sum_{j=1}^n I_{ij}, \text{ where} \quad (8)$$

$$I_{ij} = \begin{cases} 1 & \text{if } (w_i > w_j) \text{ and } (a_{ji} > 1) \\ 0.5 & \text{if } (w_i = w_j) \text{ and } (a_{ij} \neq 1) \\ & \text{or } (w_i \neq w_j) \text{ and } (a_{ij} = 1) \\ 0 & \text{otherwise} \end{cases}$$

2.4 Sensitivity Analysis

The solution to a decision problem, the global ranking of alternatives, may not provide enough information to the DM to make a final decision. There are several reasons why a sensitivity analysis (SA) should be conducted on the results. For instance, the judgements for some criteria may be subjective or there may be uncertainty in the data that leads to the preference value. In addition, the preference judgements may come from a group decision where there are different opinions. Moreover, different prioritisation methods may yield different results for the same PC matrix; at the same time different performance scoring scales used in evaluating alternatives may produce different rankings [26]. An SA provides more insight about the problem and in this way the DM should be able to make a more informed decision.

Methods to perform SA on AHP problems may be grouped into three main categories [6]: numerical incremental analysis, probabilistic simulations and mathematical models.

2.4.1 Numerical Incremental Analysis

This approach involves changing the weight values and calculating the new solution. The method, also known as One-at-a-time (OAT), works by incrementally changing one parameter at a time, calculating the new solution and graphically presenting how the global ranking of alternatives changes. This is the most commonly used method in associated software tools ([27], [8], [28], [29], [30]) and, according to Chen and Kocaoglu [6], is also the most popular in the literature where AHP is used to solve problems.

As AHP uses WSM to aggregate local priorities, the global weights are a linear function

depending on the local contributions. Given this property, the global priorities of alternatives can be expressed as a linear function of the local weights. Furthermore, if only one weight w_i is changed at a time, the priority P_i of alternative A_i can be expressed as a function of w_i using the following formula:

$$P_i = \frac{P_i'' - P_i'}{w_i'' - w_i'} (w_i - w_i') + P_i' \quad (9)$$

where P_i'' and P_i' are the priority values for w_i'' and w_i' respectively. With this method, only two iterations are necessary to produce a chart with the values of priorities of all alternatives for the range 0 to 1 of one of the weights, as show in Figure 2.2.

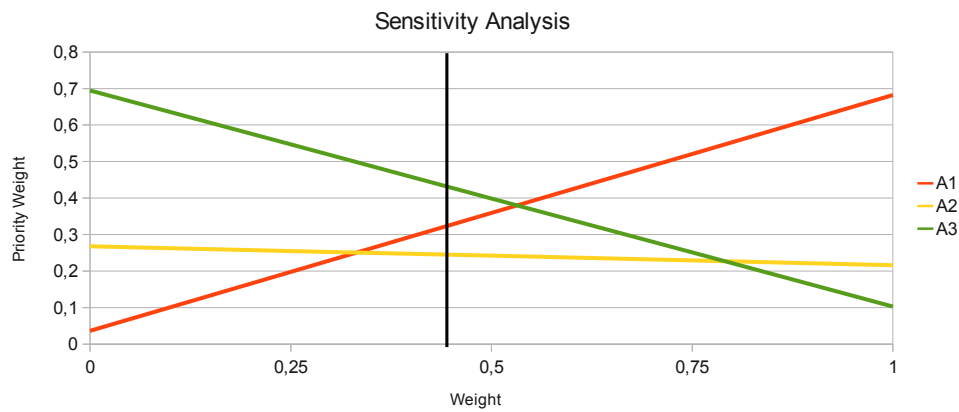


Figure 2.2. Numerical Incremental Sensitivity Analysis

The chart in Figure 2.2 is a common graphical tool to represent how the global ranking of alternatives (y-axis) is altered when the weight of one element (x-axis) is modified. The vertical back line represents the current weight of the element in the solution. In this example, it can be noted that if the weight of the element is increased above 0.5, the ranking is reversed and the alternative A1 becomes the preferred choice.

Barker and Zabinsky [31] developed a MCDM model for a logistics problem and proposed a method for finding the weights where a rank reversal⁴ is produced in the model. The approach they used can be generalised to other problems. As only one weight w_k is changed at a time, all other weights w_i can be expressed as a function of w_k . Then, formula (2) can be expressed as a function of only one variable weight w_k . By solving the equation $P_i = P_j$, the value w_k where the ranking of both alternatives is the same (rank reversal) can be found.

Another approach proposed by Hurley [32] multiplies the preference judgements in the PC matrix by a constant value and then the solution is recalculated using this new matrix. For constant values greater than 1, weights diverge; for values lower than 1, weights converge. Although this method will not produce a rank reversal, Hurley states that it may be useful to test the magnitude of the numerical weights elicited from the matrix.

2.4.2 Probabilistic Simulations

Simulation methods replace judgements in the PC matrix with values from probability distributions and perform a number of simulations to calculate the expected ranking of

⁴ Rank reversal occurs in line intersections in Figure 2.2

alternatives. As probabilistic input is used, the problem is no longer deterministic. In contrast with the previous method, this approach allows for changing more than one parameter at a time.

Butler et al. [33] proposed a method using Montecarlo simulations that allows random change of all weights simultaneously to explore the effect on the ranking. They presented three types of simulations:

1. Random weights

All criteria weights are generated completely at random in order to discover how the ranking of the alternatives changes under any conditions. To generate n weights, $n-1$ random numbers are generated in the interval 0-1 using a uniform random number generator. These numbers are sorted so that $1 > r_{n-1} > r_{n-2} > \dots > r_2 > r_1 > 0$. In addition, let $r_n=1$ and $r_0=0$. The value for the weight w'_i is calculated as $w'_i = r_i - r_{i-1}$, for $i=1..n$. The vector $W'=[w'_1, \dots, w'_n]$ will sum to 1 and is uniformly distributed.

By performing many repetitions of these steps a great number of times (5.000 iterations are used in the example), the entire domain of possible weight combinations can be explored.

For every iteration, the solution is calculated with random weights and the ranking of the alternatives is registered. After the simulation is finished, a statistical analysis can be performed to find information about the distribution of the ranking of alternatives. For instance, a box-plot chart may be created (see Figure 2.3) to depict information about the ranking of alternatives.

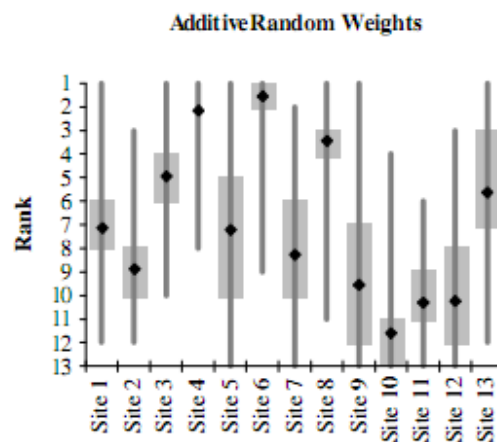


Figure 2.3. Box-plot Chart Presenting Simulation Results [33]

The black diamonds correspond to the mean ranking, the grey box encloses quartiles Q1-Q3 (25%-75% of the samples had this ranking), and the minimum and maximum ranks are the endpoints of the grey lines.

2. Random weights preserving rank order

If the ranking of criteria weights are of importance and have to be preserved, then the procedure is similar to the previous one with the difference that the random weights are ranked according to the original criteria ranking and then are used to calculate the solution. The chart presented in Figure 2.3 is still valid when using this method.

3. Random weights from a response distribution

This type of simulation considers the current weights as means of probability distributions and the random weights for the simulation are generated from these distributions. Even

though the random weights may be relatively close to the real weights, this approach may generate weights with a different rank order than the actual weights.

The formula to generate the random weights is:

$$w'_i = \frac{X_i}{X_i + \dots + X_n}, \text{ where } X_i \sim \text{Gamma}(w_i, \beta) \quad (10)$$

The variation from the mean of the generated values can be controlled with the β parameter. Again, the procedure is similar, changing only the way random weights are generated.

Another similar simulation approach, introduced by Hauser and Tadikamalla [34], explores the effects of changing the judgements in the PC matrices. The method varies the preference values from the pairwise comparisons by creating an interval for each element of the matrix. The midpoint of the interval is the current value of the judgement and the width of the interval is defined by a constant c indicating the distance from the central point as a percentage. If a_{ij} is the preference value, then the interval I_{ij} may be represented as $I_{ij} = [a_{ij} - c a_{ij}, a_{ij} + c a_{ij}]$. The next step is to generate random numbers for each interval for each of the matrices in the model. Any probability distribution may be used. Hauser and Tadikamalla used the uniform distribution and the triangular distribution to test the method. Then, the random numbers have to be converted to the AHP scale using the formula (11):

$$a'_{ij} = \begin{cases} \frac{1}{2-r} & \text{if } r < 1 \\ r & \text{if } r \geq 1 \end{cases} \quad (11)$$

For every iteration, the solution is calculated with the random judgements and the ranking of the alternatives is registered. As with Butler's method [33], a statistical analysis can be performed to find information about the distribution of the ranking of alternatives.

In addition, Hauser and Tadikamalla proposed a formula to express the final ranking of the alternatives as statistical weights representing the expected weights for the given probability distribution, according to formula (12).

$$ES_i = \sum_{k=1}^n p_{ik} (n+1-k), \text{ for } i=1 \dots n$$

$$EW_i = \frac{ES_i}{\sum_{k=1}^n ES_k}, \text{ for } i=1 \dots n \quad (12)$$

2.4.3 Mathematical Models

This group of SA methods uses mathematical models when it is possible to express the relationship between the input data and the problem solution. Mathematical models have better performance and are more efficient than the previous methods as they do not require iteration. Moreover, their results are much more accurate when using verified formulas.

Several authors have developed mathematical models for SA in AHP. Masuda [35] studied how changes throughout the whole domain in the weights of criteria may affect the ranking of alternatives and proposed a sensitivity coefficient representing the possibility of rank reversal from these changes. Huang [36] found an inconsistency in Masuda's coefficient as a large

value of the coefficient may not necessarily mean that a rank reversal will occur, and vice versa, a low value of the coefficient may produce a rank reversal. To overcome this, Huang proposed a new sensitivity coefficient and demonstrated it reveals the sensitivity of an AHP model with more accuracy than Masuda's coefficient.

Erkut and Tarimcilar [5] presented a method to analyse and visualise in the weight space (see Figure 2.4) the ranking of alternatives with respect to all possible combinations of weights for the first level of criteria. The method works by partitioning the weight space T into n subsets S_i , where n is the number of criteria. The weight space refers to all possible combinations of weights, as shown in formula (13):

$$(w_1, w_2, \dots, w_n), \quad \text{where } 0 \leq w_i \leq 1 \wedge \sum_{i=1}^n w_i = 1 \quad (13)$$

For three criteria the weight space is a triangle-shaped plane with corners in $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$. For $n=3$ criteria and m alternatives, Erkut and Tarimcilar provide a method for partitioning T into m subsets where in each subset some alternative has the highest ranking. The relative areas of these subsets indicate which alternatives are more likely to be selected.

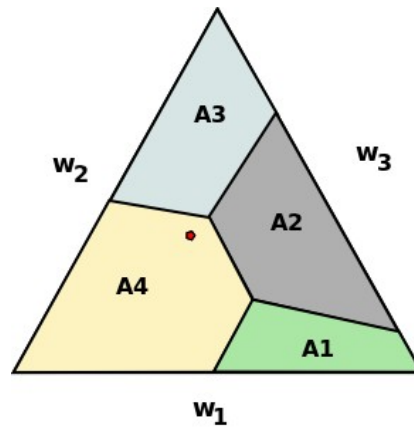


Figure 2.4. Ranking of Alternatives in the Weight Space [5]

The triangle represents all possible combination for weights w_1 , w_2 , w_3 . The red dot represents the actual weights in the solution. Each area A_i is the top-ranked alternative for the combination of w_1 , w_2 and w_3 .

However, there are limitations in their method. It does not provide a method for performing sensitivity analysis when the problem has more than one level of criteria. In addition, if more than three criteria are simultaneously considered, then visual representation is not possible and the weight space T is no longer a triangle, it becomes a convex polyhedron. Moreover, two extra algorithms have to be used to perform the partitioning [37] and the calculation of the volumes [38] (instead of areas) of the subsets.

Trantaphyllou and Sánchez [39] investigated the sensitivity analysis for problems with one level of criteria by examining the impact of changes in the weights of criteria and changes in the weights of alternatives with respect to a single criterion at a time. They proposed a method to identify the most critical criterion (for criteria weights and for alternative weights) and the quantity by which it needs to be modified so that the top-ranked alternative will change. In addition, they introduced a sensitivity coefficient as an indicator of how sensible to changes are the elements in the decision problem.

Another study to identify the most sensitive criterion was presented by Armacost and Hosseini [40]. They proposed a method to find which criteria are determinant in the final ranking of alternatives using the direct dual questioning determinant attribute analysis

(DQDA). The DQDA approach identifies the determinant criteria by measuring the *importance* ratings for the criteria and the *difference* ratings among the alternatives with respect to each criterion. The product of these two ratings for each criterion is the *determinance* score. The criteria with the highest *determinance* scores are said to be the determinant attributes of the problem.

Taji and Matsumoto [41] worked on the PC matrix sensitivity. They proposed a method to estimate intervals in which the PC values of a matrix can move without exceeding a consistency index (CI) threshold without causing recalculation of the eigenvalue and eigenvector of the matrix. As the method does not need to recalculate the eigenvalue and eigenvector of the matrix, a quick estimation of the interval is possible.

Chen and Kocaoglu [6] proposed an algorithm to examine the impact of single or multiple changes to the weights of criteria or alternatives at any level of the AHP hierarchy. Their approach finds the allowable range of changes in order to prevent a rank reversal. In addition, two sensitivity coefficients are presented and the most critical criterion in each level of the hierarchy is identified.

Iryanto [42] investigated the sensitivity of Interval AHP (IAHP). In IAHP, judgements in the PC matrix are provided as intervals instead of single values. Weights are elicited from the PC matrix using linear programming by minimising the error between the weight vector and the geometric mean of each interval. Iryanto concluded that when changes are introduced to an interval, as long as the limits of the new interval remain inside the original interval, the ranking is preserved.

2.5 Requirements for Evaluating AHP Software

This section focuses on common criteria for evaluating AHP tools. Attention is on features for the AHP and PC techniques described above. The aim is to create a framework to analyse available software tools and examine to what extent they comply with these features.

There is limited literature on this subject. A number of studies ([43], [44], [45]) have reviewed MCDM software in terms of general attributes such as platform, price and methodology⁵. Other study [46] evaluated AHP software based on the model ISO/IEC 9126 [47]. However, no study has focused on evaluating AHP features and SA.

From the material presented above (Sections 2.1 to 2.4), the main features expected in an AHP software tool are described below (Sections 2.5.1 and 2.5.2). They are divided in two groups: support for AHP and PC, and support for sensitivity analysis.

2.5.1 Support for AHP and PC

AHP software should support hierarchical models, pairwise comparison, numeric and verbal scales, consistency analysis, methods for elicitation and provide performance measures.

A1. Hierarchical models: a tool should allow a hierarchy model with arbitrary levels of criteria and number of alternatives. In this way, any decision problem can be modelled as a hierarchy and solved using AHP.

A2. PC for criteria and alternatives: pairwise comparison should be available to provide judgements for criteria and for alternatives with respect to a criterion. This means that at every level of the hierarchy the weights of the elements should be elicited from pairwise comparisons.

5 Methodology refers to the alternative ranking method used, e.g. AHP [14], MAUT [20], ELECTRE [15].

A3. Numerical and verbal scales: the scale to provide the judgements should be numerical and verbal as well, as shown in Table 2.1

A4. Consistency analyser: to assist the DM to improve consistency of the PC matrix, its consistency index (CI), ordinal consistency and cardinal consistency should be displayed.

A5. Prioritisation methods: a software tool should support more than one prioritisation method so DMs can evaluate different methods and choose a solution according to their needs.

A6. Error measures: Error measures (TD, NV) must be provided so the DM can get an indicator of the deviation of the solution from the provided judgements.

2.5.2 Support for Sensitivity Analysis

As discussed in Section 2.4 there are several approaches to conduct an SA in AHP. We conclude that an AHP tool should provide three forms of SA: numerical incremental analysis, probabilistic simulation and mathematical models. In addition, the most sensitive criteria should be presented.

B1. Numerical incremental analysis: a module for performing the one-at-a-time sensitivity analysis should be provided, along with a graph similar to the one presented in Figure 2.2 to display the sensitivity of a criterion.

B2. Probabilistic simulation: a module for performing probabilistic simulations should be provided in order to find the sensitivity of alternatives to simultaneous changes in all weights, including a graph similar to the one depicted in Figure 2.3.

B3. Mathematical models: using the mathematical approach to SA the threshold values for the weights that produce a rank reversal should be presented.

B4. Most sensitive criteria: along with the results of the SA, the most sensitive criteria should be presented and sensitivity coefficients should be included so the DM can have an estimation of how sensitive is each criterion.

B5. Number of simultaneous changes: the number of elements that can be changed at the same time to analyse the results; more than one simultaneous change should be supported.

2.6 Software Products

In this section, four tools that support AHP and PC are analysed in terms of the framework described above: Expert Choice [28], MakeltRational [8], PriEsT [9], and Web-HIPRE [29].

For each feature (A1-A5, B1-B5) a score between 0 and 3 will be assigned: 0 if the feature is not present, 1 if it is poorly implemented, 2 if it is moderately supported, and 3 if it is fully supported. The overall score for a tool will be the sum of the individual scores.

2.6.1 Expert Choice

Expert Choice ([28], [48], [49]) is a commercial desktop-based application for Windows that enables DMs to prioritise and evaluate criteria and alternatives using AHP. The analysis is detailed in Table 2.2. It is worth noting that Expert Choice only supports the basic form of SA, i.e. changing one parameter at a time and presenting the results. Its overall score is 13 out of 33.

Feature	Score	Description
A1	3	Expert Choice provides a GUI to build an arbitrary-depth hierarchy (student version is limited to 3 levels). Elements can be added, edited and removed from the model
A2	3	The software supports pairwise comparison in all levels of the hierarchy
A3	3	The judgements may be entered using a numeric or verbal scale
A4	1	Only the consistency index is displayed. No information about cardinal or ordinal inconsistency is presented
A5	0	Only the eigenvector method is included in the software. There is no option to change the method or add new ones
A6	0	No error measures are presented when calculating the weight vector
B1	2	Numerical incremental analysis is supported with three graphs that allows to change the weights of one criterion. It is not possible to change the weights of alternatives with respect to one criterion, only criteria weights
B2	0	Expert Choice does not support probabilistic simulations
B3	0	There is no support for identifying the threshold values for rank reversal. It has to be done manually using the numerical incremental analysis
B4	0	No sensitivity coefficient is provided and the most sensitive elements are not identified
B5	1	Only one parameter can be modified at a time

Table 2.2. Expert Choice Software Evaluation

2.6.2 MakeltRational

MakeltRational [8] is a commercial web-based application for decision making with a user-friendly interface. It requires the Silverlight plugin from Microsoft to run on a supported web-browser. The analysis is detailed in Table 2.3. Its overall score is 12 out of 33.

Feature	Score	Description
A1	3	MakeltRational provides a GUI to build an arbitrary-depth hierarchy. Elements can be added, edited and removed from the model
A2	3	Pairwise comparisons – or direct rating- are supported in all levels of the hierarchy
A3	2	Judgements are entered using a numerical scale, however, a description is available for each number of the scale
A4	3	The software includes the consistency index and hints for helping the DM to improve the ordinal and cardinal consistency when the judgements are inconsistent
A5	0	Only the eigenvector method is included in the software. There is no option to change the method or add new ones
A6	0	No error measures are presented when calculating the weight vector

Feature	Score	Description
B1	1	Numerical incremental analysis is supported with one static graph similar to the one in Figure 2.2. It does not allow to change the weights of criteria or alternatives
B2	0	No support for probabilistic simulations
B3	0	There is no support for identifying the threshold values for rank reversal. It has to be done manually using the graph from the numerical incremental analysis
B4	0	No sensitivity coefficients are provided and the most sensitive elements are not identified
B5	0	No changes are possible. The sensitivity graph is static

Table 2.3. *MakeltRational Software Evaluation*

2.6.3 PriEsT

PriEsT is an open source tool developed at the University of Manchester [10] that provides decision support based on AHP and PC. PriEsT is implemented in Java and is currently available for Linux and Windows. The overall score for this tool is 16 out of 33. The results from the analysis are presented in Table 2.4.

Feature	Score	Description
A1	2	PriEsT supports an unlimited number of decision elements and levels in the hierarchy. Elements can be added and edited, but not removed from the model
A2	3	Pairwise comparisons are supported in all levels of the hierarchy
A3	2	Judgements can be entered using only a numerical scale - a verbal scale is not supported
A4	3	The consistency index, cardinal and ordinal consistency measures are displayed as well as hints for helping the DM to improve overall consistency of the judgements
A5	3	PriEsT supports 11 prioritisation methods which can be selected to derive weights
A6	3	Error measures TD, NV and TD2 (total deviation using indirect judgements [10]) are presented
B1-B5	0	PriEsT has no support for sensitivity analysis

Table 2.4. *PriEsT Software Evaluation*

2.6.4 Web-HIPRE

Web-HIPRE [29] is a free web-based application for MCDM implemented as a Java applet. It was the first decision tool available online [7]. Table 2.5 presents the analysis of the software according to the previously defined features. The overall score is 14 out of 33.

<i>Feature</i>	<i>Score</i>	<i>Description</i>
A1	3	Hierarchies of up to 17 levels can be defined using the web interface – this is enough for most problems. Elements can be added, edited and removed from the model
A2	3	Pairwise comparisons are supported in all levels of the hierarchy
A3	3	Judgements can be entered using a numerical or verbal scale
A4	1	Only the consistency index is displayed. No information about cardinal or ordinal inconsistency is presented. No hints for improving the consistency of the judgements
A5	0	Only the eigenvector method is included in the software when working with AHP. There is no option to change the method or add new ones
A6	0	No error measures are presented when calculating the weight vector
B1	3	Numerical incremental analysis is supported with a dynamic graph similar to the one in Figure 2.2. It allows to change the weights of one element (criterion or alternative)
B2	0	No support for probabilistic simulations
B3	0	There is no support to identify threshold values for rank reversal. This has to be done manually for each criterion using the graph from the numerical incremental analysis
B4	0	No sensitivity coefficients are provided and the most sensitive elements are not identified
B5	1	Only one parameter can be modified at a time

Table 2.5. Web-HIPRE Software Evaluation

From a maximum score of 33, PriEsT comes first a score of 16 followed by Web-HIPRE with 14. The analysis shows that all tools have similar support for AHP and PC, the most significant difference is that only PriEsT provides more than one elicitation method and presents error measures with the results. PriEsT is the only tool with no support for SA; other tools provide a basic form of SA using the numerical incremental analysis. No tool provides sensitivity coefficients or identification of the most sensitive criteria. The lack of these features contributed to the relative low score of all the software tools.

The PriEsT tool will therefore be enhanced to include SA and the A1 and A3 features (it scored 2 for these two criteria). The associated methodology and plan is explained below.

3 Methodology

This section describes the main steps that will be taken throughout the project in order to achieve the objectives. First, the software methodology is presented. Then, re-engineering process and implementation of the sensitivity module activities are described. The section ends by considering the project evaluation, progress-to-date and the project plan.

3.1 Software Methodology

The main objective of this project is to enhance the PriEsT tool by including a module for conducting sensitivity analysis. The specific requirements to implement are presented in Section 2.5, specifically A1, A3 and B1 to B5. That list of requirements cannot be considered final as, in most software projects, requirements are subject to change throughout. In this case, requirements are likely to be revised as the project progresses. For this reason, the evolutionary prototyping development methodology [50] was selected. This method is capable of handling requirements changes by developing an initial prototype which implements well-understood requirements. Then, this prototype is validated with the stakeholders and new requirements discussed. Each successive prototype implements more functionality, and at the same time, refines previous work following stakeholder feedback.

The order in which the requirements will be implemented is shown in Section 3.5. However, as with each prototype requirements may be added or removed, that list may change.

In addition, good practice from agile methodologies will be used, such as Test Driven Development (TDD). Complex requirements, such as formulas and algorithms to perform SA, will be implemented using TDD. That is, a set of test units will be created before implementing the algorithms and will be used to build the actual methods. These test units should help ensure that mathematical formulas and algorithms are correctly implemented.

3.2 Software Design and Implementation

This section covers preliminary aspects of the re-engineering, re-design and enhanced implementation of the PriEsT tool undertaken to include SA. This process includes code restructuring and creation of a library to encapsulate the core functionalities of AHP and SA.

3.2.1 Re-engineering PriEsT

In software engineering, it is important to make a clear separation between the domain model (model) and the user interface objects (view) - normally known as the Model View Controller (MVC) pattern [50]. The motivation for this separation includes support for cohesive model definitions that focus on the domain processes rather than on UI objects, minimise the impact of changes in model requirements or user interface, allow creation of new views connected to an existing domain model, allow the model to be ported to a different user interface and allow the execution of the model without a user interface.

A detailed analysis of the design and source code of PriEsT has shown that there is no clear separation of functions between the GUI and the logic for solving AHP problems. The software is implemented using the Qt Jambi library [51]. This library uses native methods to communicate with the native QT library. For this reason, even though PriEsT is written in Java, it is not completely platform independent.

Before adding new features to PriEsT a re-engineering process is required to improve its modular structure, ensure a model-view separation and generally provide better coherence

and cohesion. Major steps in the re-engineering process are source code translation, program structure improvement, and program modularisation [11]. As part of the source code translation, the Qt Library will be replaced by the Java Swing framework to remove the dependency of native libraries and in this way improve portability. The structure improvement will involve a clear model-view separation to improve cohesion and code readability. Finally, to improve program modularity, all AHP related code will be grouped together in a Java library making it easier to re-use and maintain, hence, reducing time and effort to implement new features. A UML diagram with an initial design for this AHP library is included in Appendix A. As said previously, requirements may change and the design may be subject to modifications, but this will be the basis for developing the first prototype.

A positive outcome of this re-engineering process is that it enables extension of the platform availability of PriEsT. In recent years, there has been an increasing interest in web applications with examples like Google family of web applications, Microsoft Office 365, social networks, media, and many others. Similarly, mobile phones are increasingly being equipped with more processing power, memory and functionality. At the same time, they are becoming the main computing device for many users. Individuals carry them most of the time, and in areas of field activity such as sales and technical support, smart phones are gradually replacing the use of laptops.

There appears a natural evolution of tool provision on web and mobile platforms; however this is not apparent in MCDM and AHP software. Whilst several tools support AHP and PC (see Section 2.6), only Web-HIPRE [7], MakeltRational [8] and V.I.S.A. [52] are web-based (other tools only support Windows); there appears to exist no mobile version of AHP software.

The re-engineering enables relatively simple creation of both a Java applet and a J2ME mobile version by re-using the AHP library and implementing a user interface for the mobile platform, as Java applets can reuse the Swing user interface of a desktop application. The implementation of these platform-specific versions is part of the project plan (Section 3.5).

3.2.2 Module for Sensitivity Analysis

In Section 2.4 a literature review was carried out to identify the main developments in the area of SA for AHP. Three main groups of methods were presented describing the type of analysis they perform. To support these three types of SA, one algorithm from each group will be implemented within the PriEsT AHP tool. From the first group, an implementation of the method presented in Section 2.4.1 will be provided. From the probabilistic simulations group, the algorithms in [33] will be implemented as they allow to test the sensitivity to all possible combinations of the weights. From the mathematical models group, the method presented in [6] will be addressed as it allows multiple criteria to be changed in multiple levels of the hierarchy.

Where possible, graphical elements will be implemented, such as charts and visual aids to identify critical criteria in the hierarchy. The implementation will be done in a modular way to improve the cohesion of the application and to make it easier to port the application to a Java applet and J2ME mobile platform.

3.3 Project Evaluation

The project will include three main activities: the re-engineering process of the PriEsT application, implementation of the module for SA, and provision of desktop, web and mobile platforms. To evaluate these activities, two main steps are presented: evaluation of the SA implementation and evaluation of the various platform-specific versions of the software.

3.3.1 Evaluation of Sensitivity Analysis

The evaluation of SA will include two activities: verification of the correctness of the results and evaluation of the performance of the module.

Three SA algorithms will be implemented (Sections 2.4.1, 2.4.2, 2.4.3), each corresponding to one of the three approaches to perform SA. To ensure correctness of the results, the same decision problem(s) presented by the authors of the methods will be used to verify the output of the algorithms. In the case of the numerical incremental analysis and mathematical models, the decision problem to be used is given in [6]. For the simulation approach, the problem can be found in [33]. The expected output of the algorithms is the same as the one presented in the mentioned studies. In addition, the problem in [6] will also be used to test the simulation approach and compare its results with the mathematical approach to verify the accuracy of the former method.

The performance of the SA module will be evaluated by creating a number of random decision problems with varying number of criteria, alternatives and hierarchy levels. For each random problem, the ranking of alternatives will be calculated and the three SA algorithms will be executed. The results from these iterations will be useful for determining the execution time vs problem size, on what proportion the different methods identified the same most sensitive criterion and the conditions under which a rank reversal would occur.

3.3.2 Evaluation of platform-specific versions

The second part of the evaluation is to verify the fulfilment of requirements in each platform-specific version of the software. The same evaluation carried out in Section 2.6 using the framework presented in Section 2.5 will be performed with each platform-specific version (desktop, web and mobile). In order to do that, a decision problem [4] will be solved using each version and scores will be assigned depending on the fulfilment of the requirements.

In addition, performance tests will be executed to evaluate the response time in the mobile version, given the high-cost in terms of computational power required by the optimisation type of elicitation methods [10] and the probabilistic simulations of SA methods [33]. The results from these tests will provide feedback on the feasibility to include these kinds of methods in the mobile version.

3.4 Project Progress

Progress on the project can be summarised as follows:

- A literature review related to AHP and specifically about SA for AHP has been performed. Major aspects have been identified along with the details of algorithms and mathematical models.
- A framework for evaluating AHP software has been developed.
- A number of commercial tools (MakeltRational, Web-HIPRE, Expert Choice) have been analysed and compared in terms of features for AHP and SA using the framework.
- A full analysis of the PriEsT tool, including source code, has been carried out.
- The design of the AHP library using UML (see Appendix A) has been completed (changes are possible as the implementation progresses).
- The implementation of the AHP library has started. A basic version has been programmed in Java.

- An analysis of the requirements of the web and mobile platform was carried out to identify constraints that have to be addressed when developing the AHP library.

3.5 Project Plan

The Gantt chart in Figure 3.1 depicts the project work plan. The task in orange represents the exam period, during that timeframe the work on the project will be minimal. There is overlap between tasks because of the evolutionary prototyping methodology [50] being used. The black-horizontal line inside each task indicates the current progress to-date of that activity. The dissertation write-up will begin after the exam period, and will continue throughout the project to ensure that as soon an activity is finished it is written down.

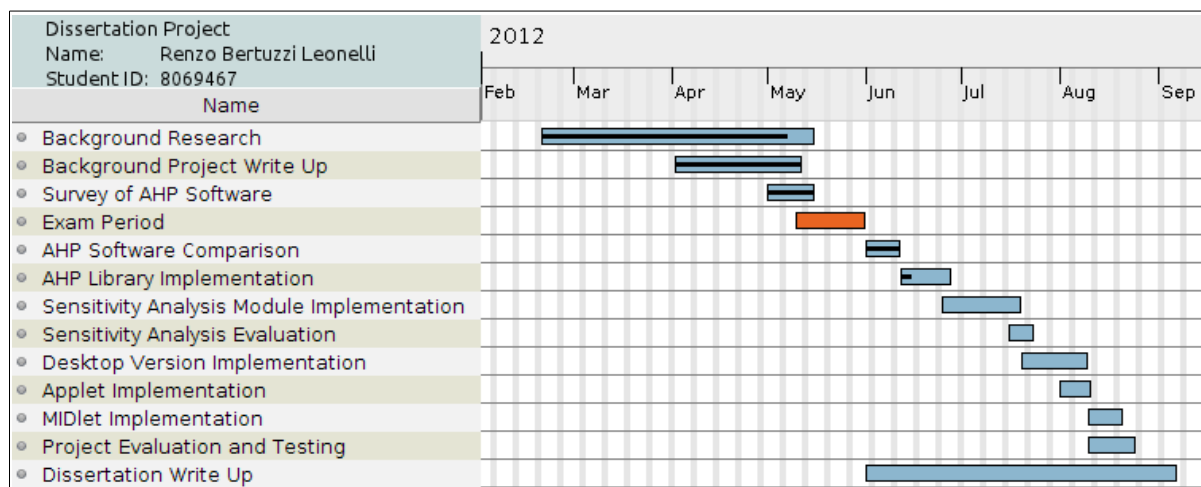


Figure 3.1. Project Gantt Chart

References

- [1] Cambell, A., Whitehead, J., and Finkelstein, S., 2009. Why Good Leaders Make Bad Decisions. *Harvard Business Review*, Vol. 87, No. 2, p.60-66.
- [2] Janis, I.L., 1972. *Victims of Groupthink: A Psychological Study of Foreign Policy Decisions and Fiascos*. Boston: Houghton-Mifflin.
- [3] Figueroa, J., Greco, S., and Ehrgott, M., 2005. *Multiple Criteria Decision Analysis: State of the Art Surveys*. Boston: Springer.
- [4] Saaty, T.L., 2008. Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, Vol. 1, No. 1, p.83-98.
- [5] Erkut, E., Tarimcilar, M., 1991. On Sensitivity Analysis in the Analytic Hierarchy Process. *IMA Journal of Management Mathematics*, Vol. 3, p.61-83.
- [6] Chen, H., Kocaoglu, D.F., 2008. A sensitivity analysis algorithm for hierarchical decision models. *European Journal of Operational Research*, Vol. 185, Issue 1, p.266-288.
- [7] Mustajoki, J. and Hämäläinen, R.P., 2000. Web-HIPRE: Global Decision Support by Value Tree and AHP Analysis. *Information Systems and Operational Research*, Vo. 38, No. 3, p.208-220.
- [8] MakeltRational, 2012. *Collaborative Decision Making Software*, [online] Available at: <http://makeitrational.com/> [Accessed 21 April 2012].
- [9] Sajid Siraj, 2012. *Priority Estimation Tool*, [online] Available at: <http://sourceforge.net/projects/priority/> [Accessed 3 March 2012].
- [10] Siraj, S., 2011. *Preference Elicitation From Pairwise Comparisons In Multi-Criteria Decision*. Ph.D. Thesis. School of Computer Science, University of Manchester.
- [11] Sommerville, I., 2000. *Software Engineering*. 6th Edition. Pearson Education.
- [12] Jacquet-Lagrèze, E. and Siskos, Y., 2001. Preference disaggregation: 20 years of MCDA experience. *European Journal of Operational Research*, Vol. 130, p.233-245.
- [13] Triantaphyllou, E., 2000. *Multi-Criteria Decision Making Methods: A Comparative Study*. Dordrecht: Kluwer Academic Publishers.
- [14] Saaty, T.L., 1980. *The Analytic Hierarchy Process, Planning, Priority Setting, Resource Allocation*. McGraw-Hill.
- [15] Roy, B., 1968. Classement et Choix en Présence de Points de Vue Multiples: la méthode ELECTRE. *Revue Française d'Informatique et de Recherche Opérationnelle*, Vol. 8, p.57-75.
- [16] Hwang, C.L., and Yoon, K., 1981. *Multiple Attribute Decision Making: Methods and Applications*. New York: Springer-Verlag.
- [17] Edwards, W., 1977. How to Use Multiattribute Utility Measurement for Social Decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-7, p.326-340.
- [18] Guitoni, A. and Martel, J.M., 1998. Tentative guidelines to help choosing an appropriate MCDA method. *European Journal of Operational Research*, Vol. 109, p.501-521.
- [19] Brans, J.P., Mareschal, B. and Vincke, P., 1984. PROMETHEE: A New Family of Outranking Methods in Multicriteria Analysis. *Operational Research*, Vol. 84, p.408-421.
- [20] Wallenius, J., Dyer, J., Fishburn, P., Steuer, R., Zionts, S., and Deb, K., 2008. Multiple Criteria Decision Making, Multiattribute Utility Theory: Recent Accomplishments and What Lies Ahead. *Management Science*, Vol. 54, No. 7, p.1336-1349.
- [21] Jacquet-Lagrèze, E. and Siskos, Y., 1982. Assessing a Set of Additive Utility Functions for Multicriteria Decision Making: The UTA Method. *European Journal of Operational Research*, Vol. 10, No. 2, p.151-164.
- [22] Benton, V. and Gear, T., 1983. On a Short-coming of Saaty's Method of Analytic Hierarchies. *The International Journal of Management Science*, Vol. 3, No. 1, p.228-230.
- [23] Triantaphyllou, E., Lootsma, F.A., Pardalos, P.M., and Mann, S.H., 1994. On the Evaluation and

- Application of Different Scales For Quantifying Pairwise Comparisons in Fuzzy Sets. *Journal of Multi-Criteria Decision Analysis*, Vol. 3, No. 3, p.133-155.
- [24] Choo, E.U, and Wedley, W.C., 2004. A Common Framework For Deriving Preference Values From Pairwise Comparison Matrices. *Computers and Operations Research*, Vol. 31, p.893-908.
 - [25] Mikhailov, L. and Singh, M. G., 1999. Comparison Analysis of Methods for Deriving Priorities in the Analytic Hierarchy Process. In: *IEEE, Systems, Man and Cybernetics, Conference Proceedings*. Tokyo, Japan, 12-15 Oct 1999.
 - [26] Steele, K., Carmel, Y., Cross, J. and Wilcox, C., 2009. Uses and Misuses of Multi-Criteria Decision Analysis (MCDA) in Environmental Decision Making. *Risk Analysis*, Vol. 29, p.26-33.
 - [27] SIMUL8 Corporation, 2008. *V.I.S.A Multiple Criteria Decision Making Software*, [online] Available at: http://www.visadecisions.com/visa_more.php [Accessed 30 April 2012].
 - [28] ExpertChoice, 2012. *Expert Choice Desktop*, [online] Available at: <http://www.expertchoice.com/products-services/expert-choice-desktop/> [Accessed 3 May 2012].
 - [29] System Analysis Laboratory, AALTO University, 2012. *HIPRE 3+ Software family*, [online] Available at: <http://www.sal.tkk.fi/en/resources/downloadables/hipre3> [Accessed 2 May 2012].
 - [30] Intelligent Decision Systems Ltd, 2010. *Making Reliable Decisions Under Uncertainty Using IDS*, [online] Available at: <http://www.e-ids.co.uk/> [Accessed 23 April 2012].
 - [31] Barker, T. and Zabinsky Z., 2011. A Multicriteria Decision Making Model for Reverse Logistics Using Analytical Hierarchy Process. *Omega*, Vol. 39, p.558-573.
 - [32] Hurley, W.J., 2001. The Analytic Hierarchy Process: A Note on an Approach to Sensitivity Which Preserves Rank Order. *Computers and Operations Research*, Vol. 28, p.185-188.
 - [33] Butler, J., Jia, J., and Dyer, J., 1997. Simulation Techniques for the Sensitivity Analysis of Multi-Criteria Decision Models. *European Journal of Operational Research*, Vol. 103, p.531-546.
 - [34] Hauser, D. and Tadikamalla, P., 1996. The Analytic Hierarchy Process in an Uncertain Environment: A Simulation Approach. *European Journal of Operational Research*, Vol. 91, p.27-37.
 - [35] Masuda, T., 1990. Hierarchical Sensitivity Analysis of Priority Used in Analytic Hierarchy Process. *International Journal of Systems Science*, Vol. 21, No. 2, p.415-427.
 - [36] Huang, Y., 2002. Enhancement on Sensitivity Analysis of Priority in Analytic Hierarchy Process. *International Journal of General Systems*, Vol. 31, No. 5, p.531-542.
 - [37] Balinski, M.L., 1961. An Algorithm for Finding All Vertices of Convex Polyhedral Sets. *Journal of the Society for Industrial and Applied Mathematics*, Vol. 9, No. 1, p.72-88.
 - [38] Cohen, J. and Hickey, T., 1979. Two Algorithms for Determining Volumes of Convex Polyhedra. *Journal of the Association for Computing Machinery*, Vol. 26, No. 3, p.401-414.
 - [39] Triantaphyllou, E. and Sánchez, A., 1997. A Sensitivity Analysis Approach for Some Deterministic Multi-Criteria Decision Making Methods. *Decision Sciences*, Vol. 28, No. 1, p.151-194.
 - [40] Armacost, R. and Hosseini, C., 1994. Identification of Determinant Attributes Using the Analytic Hierarchy Process. *Journal of the Academy of Marketing Science*, Vol. 22, p.383-392.
 - [41] Taji, K. and Matsumoto, K., 2006. Inverse Sensitive Analysis of Pairwise Comparison Matrices. *Journal of the Operations Research Society Japan*, Vol. 49, No. 4, p.332-341.
 - [42] Iryanto, 2006. Sensitivity Analysis of the Interval Analytic Hierarchy Process. In: *Universiti Sains Malaysia, Proceedings of the 2nd IMT-GT Regional Conference on Mathematics, Statistics and Applications*. Penang, 13-15 June.
 - [43] Buede, D., 1992. Software review. Three packages for AHP: Criterium, expert choice and HIPRE 3 +. *Journal of Multi-Criteria Decision Analysis*, Vol. 1, No. 2, p.119-121.
 - [44] Buede, D., 1996. Second Overview of the MCDA Software Market. *Journal of Multi-Criteria Decision Analysis*, Vol. 5, p.312-316.

- [45] Weistroffer, R. and Narula, S., 1997. The State of Multiple Criteria Decision Support Software. *Annals of Operations Research*, Vol. 72, p.299-313.
- [46] Ossadnik, W. and Lange, O., 1999. AHP-based Evaluation of AHP-Software. *European Journal of Operational Research*, Vol. 118, p.578-588.
- [47] Caia, L., Xie, X., and Huang, S., 2011. Software Quality Model Development - An Introduction. In: ESEP, *Energy Procedia*. 9-10 December.
- [48] Ishizaka, A. and Ashraf, L., 2009. Analytic Hierarchy Process and Expert Choice: Benefits and Limitations. *ORInsight*, Vol. 22, No. 4, p.201-220.
- [49] Chan, F.T.S., Chan, H.K., Ip, R.W.L. and Lau, H.C.W., 2007. A Decision Support System for Supplier Selection in the Airline Industry. *Proceedings of the Institution of Mechanical Engineers. Part B, Journal of Engineering Manufacture*, Vol. 221, No. 4, p.741-758.
- [50] Larman, C., 2004. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Third Edition. Addison Wesley Professional.
- [51] Qt Jambi Team, 2011. *The Qt library for Java*, [online] Available at: <http://qt-jambi.org> [Accessed 6 May 2012].
- [52] SIMUL8 Corporation, 2012. *What is V.I.S.A Decision Making Software?*, [online] Available at: http://www.visadecisions.com/visa_more.php [Accessed 30 April 2012].

Appendix A

UML class diagram showing the design of the AHP library. It includes a DecisionProblem with alternatives and the hierarchy of criteria, prioritisation methods (AbstractMethod) for solving the problem, PCMatrix and import/export interfaces.

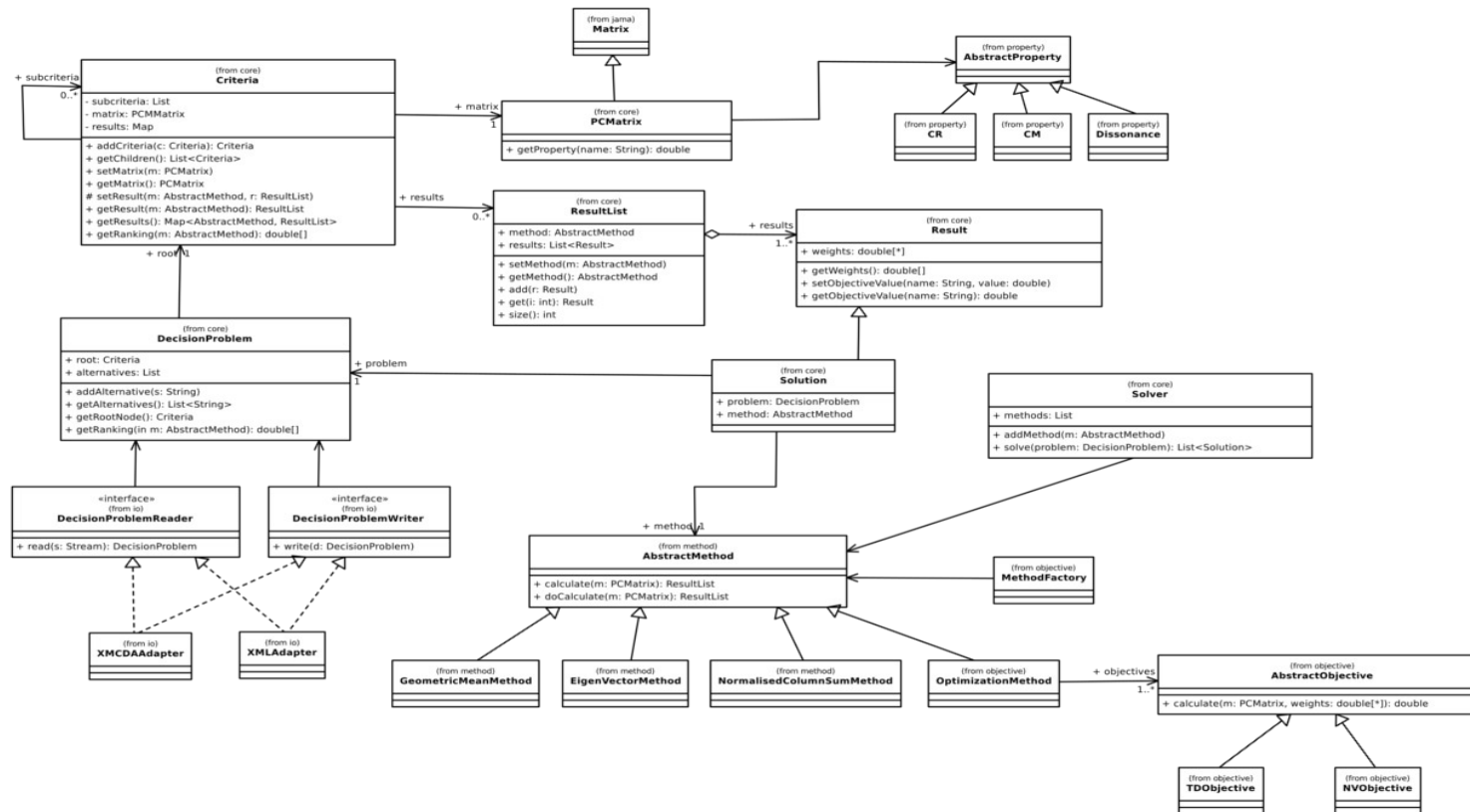


Figure 3.2. UML Class Diagram for AHP Library