Enhancing PriEsT with Group Decision Making

Progress Report

May 2013

Author: Wei Wei
Supervisor: John Keane
Abstract

Multi-Criteria Decision Making (MCDM) allows Decision Makers (DM) to consider many alternatives under various criteria. As an active research area, the Analytic Hierarchy Process (AHP) provides standardised solutions to rank these alternatives. In AHP, people use pairwise comparison (PC) method to compare only two alternatives or criteria at each time under one criterion. Many findings claim decision made by a group is probably more reliable and objective. Group Decision Making (GDM) is a technique that aggregates individual judgements from DMs to generate a group result.

PriEsT is a priority estimation tool using the PC method. It allows DMs to express and revise their judgements based upon the consistence measures and graphical aids it provides. The primary aim of the project is to realise a GDM feature for PriEsT. This report initially investigates several methods and models to carry out GDM, including the aggregation of individual judgements (AIJ) and the aggregation of priorities (AIP), both of which will be implemented. After a set of background research, a framework for assessing AHP and GDM features is created, and several commercial and academic tools are analysed using that framework. The result illustrates that PriEsT has a huge potential, as it is already equipped with a wide range of prioritisation methods and three forms of SA algorithm. The requirements for implementing GDM module are determined in the report. Then it gives the information about the architecture of the enhanced PriEsT system. It will also introduce the project methodology and project evaluation methods.
List of Contents

Abstract ................................................................................................................................. 2

List of Figures ....................................................................................................................... 4

List of Tables ......................................................................................................................... 4

List of Acronyms .................................................................................................................. 5

1. Introduction ....................................................................................................................... 6
   1.1. Aim and Objectives ....................................................................................................... 7
   1.2. Deliverables .................................................................................................................. 7
   1.3. Report Overview ......................................................................................................... 8

2. Literature Review .............................................................................................................. 9
   2.1. Multiple Criteria Decision Making ............................................................................ 9
   2.2. XMCDA ...................................................................................................................... 9
   2.3. Pairwise Comparison Method ................................................................................... 10
   2.4. Analytic Hierarchy Process ....................................................................................... 11
   2.5. Prioritisation Methods .............................................................................................. 12
   2.6. Sensitivity Methods ................................................................................................. 14
   2.7. Group Decision Making ........................................................................................... 15
      2.7.1. Preference Relations ............................................................................................ 15
      2.7.2. Consensus Model ............................................................................................... 16
      2.7.3. Selection Model .................................................................................................. 17
   2.8. Software Products ...................................................................................................... 18
      2.8.1. Marking Scheme ................................................................................................. 18
      2.8.2. Evaluation of Software ..................................................................................... 19
   2.9. Summary .................................................................................................................... 22

3. Analysis and Requirements ............................................................................................... 23
   3.1. Current PriEsT ........................................................................................................... 23
      3.1.1. Functions and Architecture ............................................................................... 23
      3.1.2. Analysis of Dependency and Quality of Code .................................................... 23
   3.2. Functional Requirements ............................................................................................ 25
      3.2.1. Concerning Factors ........................................................................................... 25
      3.2.2. Functional Requirements .................................................................................. 25
      3.2.3. Non-functional Requirements ............................................................................ 26
   3.3. Use Case Diagram ..................................................................................................... 26

4. Progress ............................................................................................................................. 27
   4.1. Architecture of GDM ................................................................................................. 27
      4.1.1. Redesign ............................................................................................................. 27
      4.1.2. Architecture of New Module ............................................................................. 27
   4.2. Logic of GDM ............................................................................................................ 28
4.3. Interface Design ............................................................................................................. 29
4.4. Implementation Environment and Tools ..................................................................... 30
4.5. Implementation and Testing Methodology .................................................................... 31
4.6. Project Evaluation ......................................................................................................... 32
  4.6.1. Evaluation of Web Version ....................................................................................... 32
  4.6.2. Evaluation of Mobile Version ................................................................................... 32
4.7. Progress .......................................................................................................................... 33
4.8. Project Plan ..................................................................................................................... 33
5. Reference ......................................................................................................................... 34

List of Figures
Figure 1: AHP architecture ................................................................................................. 12
Figure 2: Numerical incremental sensitivity analysis .......................................................... 14
Figure 3: Box-plot Chart representation of simulation results ........................................... 14
Figure 4: ranking of alternatives in the weight space ......................................................... 14
Figure 5: A framework of GDM processes (Wu, and Xu, 2012) ......................................... 15
Figure 6: Class dependency analysis .................................................................................. 24
Figure 7: show bugs by type ............................................................................................... 24
Figure 8: show bug instances by class ................................................................................. 24
Figure 9: show bug instances by type ................................................................................ 24
Figure 10: show bug instances by class .............................................................................. 24
Figure 11: Use case diagram .............................................................................................. 26
Figure 12: Architecture of the enhanced PriEsT ................................................................. 28
Figure 13: Initial desktop interface of facilitator view ....................................................... 30
Figure 14: Initial interface of the participant view ............................................................. 30
Figure 15: GDM process diagram ...................................................................................... 30
Figure 16: The incremental process model (Pressman, 2010) ............................................. 31
Figure 17: Gantt chart for project plan ................................................................................. 33

List of Tables
Table 1: Random consistency index (Siraj et al, 2012a) ..................................................... 11
Table 2: Prioritisation methods ......................................................................................... 12
Table 3: Marking scheme for the evaluation of SA and GDM feature ................................. 19
Table 4: Progress ............................................................................................................... 33
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHP</td>
<td>Analytic Hierarchy Process</td>
</tr>
<tr>
<td>DM</td>
<td>Decision Maker</td>
</tr>
<tr>
<td>DSS</td>
<td>Decision Support Systems</td>
</tr>
<tr>
<td>EV</td>
<td>Eigenvector</td>
</tr>
<tr>
<td>GDM</td>
<td>Group Decision Making</td>
</tr>
<tr>
<td>GM</td>
<td>Geometric Mean</td>
</tr>
<tr>
<td>MCDM</td>
<td>Multi-Criteria Decision Making</td>
</tr>
<tr>
<td>MPDM</td>
<td>Multi-Person Decision Making</td>
</tr>
<tr>
<td>MVC</td>
<td>Model-View-Controller</td>
</tr>
<tr>
<td>NV</td>
<td>Number of Priority Violations</td>
</tr>
<tr>
<td>PC</td>
<td>Pairwise Comparison</td>
</tr>
<tr>
<td>PCM</td>
<td>Pairwise Comparison Matrix</td>
</tr>
<tr>
<td>PriEsT</td>
<td>Priority Estimation Tool</td>
</tr>
<tr>
<td>PrInt</td>
<td>Prioritisation using Indirect Judgement</td>
</tr>
<tr>
<td>XMCDA</td>
<td>eXtensible Multiple Criteria Decision Analysis</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Makeup Language</td>
</tr>
<tr>
<td>SA</td>
<td>Sensitivity Analysis</td>
</tr>
<tr>
<td>TD</td>
<td>Total Deviation</td>
</tr>
</tbody>
</table>
1. Introduction
In daily life, people normally make decisions according to intuition, such as selecting an object among a range of options. However, the human brain is probably unreliable if there are many factors to consider simultaneously, or alternatives are conflicting such as cost and quality. Under those circumstances, techniques and tools such as decision support systems (DSSs) are necessary to support the people who are responsible for making decisions, known as decision makers (DMs) (French et al, 2009).

Multiple criteria decision making (MCDM) is a supportive decision support technique allowing DMs to make a decision after evaluation of multiple elements (can be criteria or alternatives) (Belton and Stewart, 2001). However, comparing multiple elements at the same time is difficult for human beings. In MCDM, the pairwise comparison (PC) method allows a DM to compare any one pair of two alternatives or criteria at a time (Siraj et al, 2012b), with the help of a pairwise comparison matrix (PCM) whose rows represent alternatives to be judged and columns are correspond attributes or criteria.

After simply assigning a value to one pair of elements using a PC, the preference between these two elements and the strength of the preference are built. However, the preference could be inconsistent owing to human nature. For example, if a DM has made A larger than B (A>B) and B larger than C (B>C), the direction of A and C should be A>C, however the DM makes it C>A. After removal of all inconsistencies, there exist a certain numbers of matrixes, each of which indicates the preferences of alternatives under one criterion. The final ranking is produced by combining all matrixes with their correspond weights of importance.

In MCDM, there is an active area called Analytic Hierarchy Process (AHP) where PCM is used. AHP decomposes decision problems into a tied hierarchy tree including a number of smaller sub-trees (Saaty, 2008a). Sub-problems can be easily analysed individually or further decomposed if necessary. One element or node in the tree can denote any factor of a problem, no matter it is tangible or intangible, understandable or badly understood, or a precise figure or estimated one etc. Once the AHP hierarchy tree is created, a DM identifies judgements for all nodes using PCM. The procedure of AHP ends with the generation of a preference vector containing rankings of alternatives as mentioned.

Most often decisions are made by a group of a reasonable size (French et al, 2009). To obtain a result from individual preferences, people in the group should reach a certain degree of consensus beforehand. These two processes, consensus and selection, are two of the main processes of an activity, called Group Decision Making (GDM) (Dong et al, 2010). In recent years, many GDM problems are modelled with the use of AHP. AHP-GDM applications are the extension of individual-AHP problems (Srdjevic and Srdjevic, 2013). The two most widely known algorithms for AHP-GDM are the aggregation of individual judgements (AIJ) and the aggregation of priorities (AIP) (Dong et al, 2010). They are used in different contexts, depending on whether the team is viewed as a unit seeking for a consensus or a group with individuals holding different value systems.

One emerged question is that given a group goal, different DMs might use different prioritisation approaches to elicit their personal judgements as there exists not widely
accepted the most suitable method (Siraj et al, 2012b). Such approaches or algorithms are developed in isolated software solutions and thus are poor in interoperability (Decision Deck, 2013). The advent of eXtensible Multi-Criteria Decision Analysis (XMCDA) removes that barrier. Sponsored by Decision Deck, XMCDA is a new XML based standard, designed to make MCDA algorithms interoperable and offer software a better portability.

PriEsT is a priority estimation tool, developed at the University of Manchester, which has just been enhanced by Sensitivity Analysis (SA). SA allows group members to observe how changes in their judgments or in the weights of the criteria affect the final preference vector (Erkut and Tarimcilar, 1991). The goal of the project is to enhance PriEsT with GDM. To achieve this goal, the most challenging part is how to aggregate individual judgements from multiple participants. Accordingly, the report will show two algorithms and depict the process of making a group decision. The report also contains background research for other system features implemented and relevant techniques. Then, it presents an evaluation model for assessing SA and GDM features of PriEsT and other similar DSSs. The model uses the general requirements as the marking scheme. After the project design section, a methodology to develop GDM is described. Finally, the methods used to evaluate the enhanced system will be presented.

1.1. Aim and Objectives
The primary aim of this project is to enhance PriEsT using GDM. In order to fulfil this aim, there are three objectives:

1) Adapt the software architecture into a web service to enable GDM;
2) Apply the XMCDA standard to PriEsT for better integration and migration in the future;
3) Add GDM functionality for users regardless of their locations via mobile devices;

1.2. Deliverables
As a development-focused dissertation, there are three software deliverables and two reports.

1) Adapted PriEsT
   a) Modification of the architecture to make PriEsT a distributed web service-based system;
   b) Extension of the system to support XMCDA;
   c) UML diagrams to describe the enhanced system.

2) GDM enabled PriEsT
   a) Design and implementation of new interfaces for GDM;
   b) Creation of new logic;
   c) Capability test for GDM performance.

3) Smart phone version PriEsT
   a) Creation of UI for Android mobile phone;
   b) Implementation of mobile PriEsT with GDM;
   c) Communication test between mobile devices.

4) Progress Report
   a) Background review, including PC, AHP, MCDM, SA,GDM;
   b) Evaluation of SA and GDM feature;
c) Initial design and implementation methodology.

5) Final report
   a) Extended background research;
   b) Evolved project requirement and design as appropriate;
   c) Implementation of each version of software;
   d) Testing and analysis;
   e) Development of case study;
   f) Source code.

1.3. Report Overview
The remaining part of the report consists of the following contents:

Chapter 2 Literature Review: this chapter reviews relevant background literature. It briefly introduces MCDM, XMCDA, PC, AHP, prioritisation methods and SA. Next, it reviews two general models in GDM and two relevant algorithms, AIP and AIJ. Finally, this chapter compares PriEsT with several similar software products.

Chapter 3 Analysis and Requirement: this chapter examines the current version of PriEsT. After that, it defines functional and non-functional requirements of the project.

Chapter 4 Progress: this chapter begins by giving architecture of the enhanced system and the main process of GDM. Then it describes the interface design, followed by the methodology and evaluation methods for the project. Finally, the progress is summarised and project plan is proposed.
2. Literature Review
This chapter reviews relevant background literature. To begin with, it gives a definition of MCDM, the XMDCA language, PCM, AHP, and prioritisation methods in sequence. Then it introduces the process of Group Decision Making as well as two associated algorithms. An evaluation of DSSs is presented at the end.

2.1. Multiple Criteria Decision Making
MCDM is an active research area that ranks alternatives under different criteria to support decision-making activities. It has made a significant contribution to the improvement of DSS in terms of its theory research and applications since the early 1980s (Zeleny, 1982). Dozens of MCDM approaches have been developed to identify and access alternatives. Each one is differentiated in terms of information aggregation, according to Baltussen et al (2006). Those methods are broadly categorised into three main sub-groups: value measurement, goal programming and outranking models (Belton and Stewart, 2001).

Value measurement methods require a DM to determine preference between criteria or alternatives, e.g. AHP that will be introduced in section 2.4. The weighted sum model (WSM) is also a good illustration of a value-oriented method, and firstly appeared in the study of Fishburn, P.C (1967). The algorithm is popular because of its simplicity. A similar approach is the weighted product model (WPM). In contrast, goal-oriented methods ask a DM to present their desired levels of criterion attainment that are not common in the first group of methods (Anndavey and Olson, 1998). An example is the Step Method (Benayoun et al., 1971). ELECTRE belongs to the third group, first suggested by Bernard Roy in France (Hatami-Marbini and Tavana, 2011).

MCDM has been applied in various domains. Wang et al (2011) proposed a model as a strategic planning tool for environmental supply chain issues. Because the majority of SEMs are currently suffering from low profits because of the shortage of knowledge management, Hung et al (2010) presented a method to tackle this problem by adoption of MCDM. In the context of incident control systems, Peng et al (2010) described a new framework with an MCDM component for immediately assessing the latest situation, searching solutions and accordingly reacting. Hakanen et al (2010) created an interactive tool for improving performance of wastewater treatment design by considering both environmental and financial aspects.

2.2. XMCDA
XMCDA is a XML-based data standard designed specifically for the representation of MCDA elements and objects. It is maintained by Decision Deck (2013). In this project, all the necessary materials e.g., J-XMCDA library and documentation, that are associated with XMCDA are provided by the Decision Deck project. It also provides a number of web services that can be used directly. This reduces the total development time.

The advantages of using XMCDA include 1) easy interaction between MCDA algorithms; 2) the ability of executing several algorithms on one instance of the same problem and 3) the ability to show graphic interface of MCDA concepts and data structure via universal platforms. Examples of projects, which have already supported XMCDA, are diviz software, the kappalab R library, the Python digraphs, the J-MCDA project and d2, d3 software etc.
The structure of a XMCDA is similar to that of XML in that there are several tags in different hierarchies under the root element. Generally, tags are divided into the following categories:

- Project or file description;
- Output messages (e.g. Log or error messages) from the method and input messages like parameters of methods;
- The performance table;
- The description of major MCDA concepts including criteria, attributes, alternatives, categories;
- Further information or description related to those major MCDA concepts above.

(Decision Deck, 2013)

XMCDA has several elementary types, which are values, intervals, points, scales, functions, and description. More information and detailed documentation about XMCDA is available in (Decision Deck, 2013)

2.3. Pairwise Comparison Method

In MCDM, rating each option is a natural way to represent a preference of a DM or a decision team. However, in many cases, it is difficult for a person to evaluate several alternatives of a criterion directly. The PC method provides a more tangible manner where two stimuli (i.e. Alternatives) are compared at a time.

Assume there are n elements named E1, E2 ... En. They represent either criteria or alternatives in MCDM. Let aij denotes the judgement of relative importance, i.e. the degree of preference, of any two elements Ei and Ej. If Ei is more important, aij > 1, otherwise aij < 1 or aij =0 if they are equally important. In AHP, judgements of all elements are structured in a positive reciprocal matrix (i.e. aij = 1/aji), called a pairwise comparison matrix (PCM). Broadly speaking, it is called the multiplicative preference relation. Both terms will appear frequently in the following sub-chapters.

$$A = \begin{bmatrix}
1 & a_{12} & a_{13} & \ldots & a_{1n} \\
1/a_{12} & 1 & a_{23} & \ldots & a_{2n} \\
1/a_{13} & 1/a_{23} & 1 & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots \\
1/a_{1n} & 1/a_{2n} & \ldots & \ldots & 1
\end{bmatrix}$$

After determination of such a matrix, the problem becomes a mathematical one: generating a priority vector (w1,w2, ...,wn)^T, also known as prioritisation (Siraj et al, 2012a). It is applied to estimate a preference intensity vector (r1, r2, ..., rn)^T, which is unknown to DMs. Different types of prioritisation methods are introduced in 2.5.

Consistency of PC Judgements

(1) Cardinal Consistency

In PC, an on-going issue is the consistency (Siraj et al, 2012a). From Saaty (1980), a DM is said to be cardinally consistent, if the following conditions are satisfied:

$$a_{ij} = \frac{1}{a_{ji}} \text{ for all } i \text{ and } j;$$
B. $a_{ij} = a_{ik} \times a_{kj}, \text{where } j > k > i$

If the judgements of the DM are of consistence, the priority vector and preference vector are the same. The fact is DMs are frequently cardinally inconsistent and accordingly, for some $i, j, k$, $a_{ij} \neq a_{ik} \times a_{kj}$. Satty (1980) proposed a measurement to judge whether the PCM is of acceptable inconsistence. A consistency index (CI) is computed by:

$$CI = \frac{\lambda_{\text{max}} - n}{n-1},$$

where $n$ is the size of the comparison matrix and $\lambda_{\text{max}}$ is the largest eigenvalue. Satty also defined the random consistency index (RI). The Value of RI can be referred to Table 1. Then the consistency ratio (CR) is measured by:

$$CR = \frac{CI}{RI}$$

If $CR < OR CR = 0$, the w vector appropriately estimates the unknown preference vector and the PCM is acceptably consistent. In contrast, $CR$ greater than 0.1 means the priority vector contains many errors and the DMs should update the subjective value of the PCM.

**Table 1: Random consistency index (Siraj et al, 2012a)**

<table>
<thead>
<tr>
<th>N</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>0.52</td>
</tr>
<tr>
<td>4</td>
<td>0.89</td>
</tr>
<tr>
<td>5</td>
<td>1.11</td>
</tr>
<tr>
<td>6</td>
<td>1.25</td>
</tr>
<tr>
<td>7</td>
<td>1.35</td>
</tr>
<tr>
<td>8</td>
<td>1.40</td>
</tr>
<tr>
<td>9</td>
<td>1.45</td>
</tr>
</tbody>
</table>

(2) **Ordinal Consistency**

Also known as transitivity, the ordinal consistency is defined as:

$$\text{If } E_i \rightarrow E_j \rightarrow E_k, \text{then } E_i \rightarrow E_k$$

However, if at the meantime time $E_k \rightarrow E_i$, the judgements are ordinal inconsistence and a three-way cycle is generated (Kendall and Smith, 1940).

In the context of PCM, $E_i \rightarrow E_j$ denotes that $a_{ij} > 1$. A three-way cycle means that the associated judgements are $a_{ij} > 1, a_{jk} > 1$ and $a_{ki} > 1$. The relationships among these elements can also be displayed in a graph as a triangle.

2.4. **Analytic Hierarchy Process**

The foundation of PriEsT is AHP that is one of the methods where PC is used. AHP was developed by Saaty in 1980. Owing to its simplicity and powerfulness, it has been widely used in applications where decision selection is needed. Bhushan and Rai (2004) summarised such into choice selection, prioritisation or evaluation, resource allocation, benchmarking and quality management.
A typical AHP architecture is shown in Figure 1, which is adapted from a classic application: “Choosing a leader” (Saaty, 2008). Bhushan and Rai (2004) gave the classical AHP procedure consisting of several steps:

**Step 1:** Breaking the problem into a hierarchy of goal, criteria, sub-criteria (optional) and alternatives (refer to Figure 1). This step is believed to be the most critical and creative part of the whole procedure.

**Step 2:** Depending on the hierarchical architecture obtained, to assign value to each attribute using the pairwise comparison method based on data collected by the analysts, experts or DMs.

**Step 3:** Moving the figures generated at step 2 into a PCM.

**Step 4:** The various criteria being compared are assigned values for their importance.

**Step 5:** Evaluation of the consistency of the matrix of size $n$.

**Step 6:** Calculation of local ratings by multiplying each sub-criterion if exists with each corresponding weight and calculation of final ratings including every criteria and its correspond weight.

### 2.5. Prioritisation Methods

As mentioned in 2.3, a prioritisation method can extract a priority vector from PCM, Choo and Wedley (2004) summarised 18 methods and compared them in a numerical way. Some examples are additive normalisation (AN), Eigenvector method (EV), direct least squares (DLS), weighted least squares (WLS), logarithmic least squares (LLS) or row geometric mean method (RGMM), Logarithmic least absolute values (LLAV) etc. These methods are concluded below with their mathematical formulas (Siraj et al., 2012b).

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Mathematical Formula</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN</td>
<td>$\bar{a}<em>{ij} = \frac{a</em>{ij}}{\sum_{i}a_{ij}}$, $w_i = \frac{1}{n} \sum_{j} \bar{a}_{ij}$</td>
<td>The advantage is ease of calculation. The preference vector is obtained by normalising the columns of a positive symmetrical reciprocal matrix and averaging the figures in each row.</td>
</tr>
<tr>
<td>Method</td>
<td>Formula</td>
<td>Notes</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>EV</td>
<td>$E$</td>
<td>EV is not suitable for PCM with great inconsistencies (CR &gt; 0.1).</td>
</tr>
<tr>
<td>DLS</td>
<td>$d_{DLS}(w) = \sum_{i=1}^{n} \sum_{j=1}^{n} (a_{ij} - \frac{w_i}{w_j})^2$</td>
<td>It greatly reduces the difference between judgements and human estimations. Limited use as a nonlinear solution.</td>
</tr>
<tr>
<td>WLS</td>
<td>$d_{WLS}(w) = \sum_{i=1}^{n} \sum_{j=1}^{n} (w_j a_{ij} - w_i)^2$</td>
<td>It is able to tackle both linear and nonlinear optimisation problems.</td>
</tr>
<tr>
<td>LLS (RGMM)</td>
<td>$d_{LLS}(w) = \sum_{i=1}^{n} \sum_{j=1}^{n} (\log ij - \log w_i + \log w_j)^2$</td>
<td>This method is used on the basis of the assumption that the best vector is the one with a minimal sum of logarithmic squared deviations. The solution is always unique.</td>
</tr>
<tr>
<td>LLAV</td>
<td>$d_{LLAV}(w) = \sum_{i=1}^{n} \sum_{j=1}^{n}</td>
<td>a_{ij} - \frac{w_i}{w_j}</td>
</tr>
</tbody>
</table>

Siraj (2011) proposed two new methods based on the graph theory: Enumerating All Spanning Tree (EAST) and Prioritisation using Indirect Judgement (PrInT). Both are available in PriEsT. PrInT outperforms all other methods regarding inconsistent PCM, while EV is one the most common methods as well as the Geometric Mean (GM) and Normalise Column Sum (NCS), thanks to their simplicity (Leonelli, 2012).

**Criteria for Comparison**

One question still remains unclear that which criterion to be used for comparison of prioritisation methods. Two commonly used criteria are total deviation (TD) and priority violations. Other criteria including conformity, computation complexity, and sensitivity to modifications in initial judgements are also used to measure prioritisation methods; however, they are believed to be less significant (Siraj et al., 2012b).

TD is calculated by summing all quadratic errors (residuals) between the given PCM and an estimated preference vector. The DLS method performs better than any other methods if TD is the only factor considered for ranking all prioritization methods. The reason is that $TD(w) = d_{DLS}(w)$. TD is Mathematically denoted as:

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

Where $a_{ij}$ and $w_i, w_j$ are defined in section 2.3.

The term priority violations came from the concept of the tournament-ranking problem that was firstly used by Ali et al. (1986). Assume there are two stimuli $w_i$ and $w_j$, they should preserve the direction from $w_i$ to $w_j$ if $S_i -> S_j$. However, $S_j$ could become larger so that $w_i < w_j$ when eliciting preferences. In this case, a violation occurs. Yuen (2010) introduced a form to compute the number of violations (NV) as follows:

$$NV = \sum_{i=1}^{n} \sum_{j=1}^{n} l_{ij}, \text{ where}$$
2.6. Sensitivity Methods

Whatever the problem is, in AHP it turns out to be a series of mathematical calculations. In all mathematical procedures, the input usually determines the output. Therefore, a method called the sensitivity method (SA) that can observe the behaviour of output depending on the change of input is undoubtedly necessary.

Two approaches to vary the inputs are deterministic variation and stochastic variation (French et al, 2009). The former involves setting bounds to conduct computation to predict the possible conditions of output. The airline example proposed by French (1992) suggested a description of how to apply the SA methodology to solve an airline-purchase problem. In another way, the inputs change stochastically.

There are three main categories of method to perform SA in AHP, namely numeric incremental analysis, probabilistic simulations and mathematical models. More information about SA is available in Leonelli (2010). Please refer to chapter 2.4 in Leonelli’s work for more information. Figure 2, Figure 3 and Figure 4 provide an overview of the probable representation of each (Leonelli, 2012).

\[
I_{ij} = \begin{cases} 
1, & w_i > w_j \text{ and } a_{ij} > 1 \\
0.5, & w_i = w_j \text{ and } a_{ij} \neq 1 \\
0.5, & w_i \neq w_j \text{ and } a_{ij} = 1 \\
0, & \text{otherwise}
\end{cases}
\]
2.7. Group Decision Making

The fact that decisions are becoming difficult to be made requires everyone in a group contributes to them. Raiffa et al (2003) defined group decision-making (GDM) as “a process in which the group interacts and discusses a series of solutions, converging through negotiations on a deal or policy that all find acceptable”. Current PriEsT is a single user targeted DSS without GDM support. As in practical groups outperform single DM, the four primary reasons are:

1) A greater amount of knowledge, intelligence and understanding of the problem
2) Increased motivation to work hard as the performance is open to all DMs
3) More creativity according to other’s ideas
4) Fewer errors and less ambiguity

(French et al, 2009)

2.7.1. Preference Relations

Preference relations are powerful tools for modelling experts’ preferences in GDM in two forms, namely linguistic preference and numerical preference relations. The latter can be further classified into multiplicative preference relations (also known as a PC judgement matrix in AHP) and fuzzy preference relations (Herrera-Viedma et al, 2002). This report focuses on the investigation of multiplicative preference relations as it is used in PriEsT. In general, the flow of GDM activities with multiplicative preference relation is outlined in Figure 5.

As indicated in Figure 5, two models (processes), selection model and consensus model are applied as long as using preference relations in GDM. The remainder of this section will introduce them in depth. It begins with an introduction of the consistency control process that comes before them.

![Figure 5: A framework of GDM processes (Wu, and Xu, 2012)]
2.7.2. Consensus Model

Ideally, the definition of the consensus model is a complete agreement on all possible alternatives by all the DMs. In practice, it uses the concept of consistency measure that is also known as the soft consensus degree (Herrera-Viedma et al., 2002; Herrera et al., 1996). The consistency measure is a metric of the difference among all DMs’ preferences. In GDM, it involves two sub-problems: 1) Considered individually, when a DM is consistent and 2) when a team is consistent (Herrera-Viedma et al., 2004). The procedures of solving these two problems are briefly shown in the second and third frame in Figure 5.

Individual Consistency

The individual consistency control feature has already been realised in PriEsT, and this part introduces its technical background.

Consistency index is a widely used term reflecting the degree of consistency. In section 2.3, the computation of that term using Satty’s approach has already been described, as well as other concepts of consistency.

There are many tactics to decrease the inconsistency of PCMs. Some modify the entire PCM by a series of auto-adaptive model (Xu and Wei, 1999). Some modify the most inconsistent data of a PCM and preserve the remainder (Cao et al., 2008). Siraj (2012a) argued that ordinal inconsistency is the main reason why priorities obtained do not correspond to the DM’s preferences. Hence increasing ordinal consistency could be seen as a significant solution to optimise the decision making process.

A heuristic algorithm (Siraj et al., 2012a) is useful for removal of intransitive judgements (Ordinal inconsistency), and has been implemented in PriEsT. The algorithm detects all three-way cycles (e.g. $A \rightarrow B \rightarrow C \rightarrow A$), at the first step. One cycle means the existence of a pair of intransitive judgements. Then all judgements affecting the intransitivity of the overall PCM can be determined. By using an optimization method, a minimal number of data into a PCM are changed to remove all the three-way cycles, i.e. all the intransitive judgements. In PriEsT, those cycles are visible on a graph. That provides a straightforward advice for a DM to correct his judgements (Figure 5).

Group Consistency

The consensus measure that counts the closeness among all people’s opinions is a vital factor of consensus models. A consensus process is dynamic and includes a chain of group discussion for reaching a consensus. During these iterative rounds, the DMs are advised by a moderator to change their preferences based on the consensus measure, if the difference is higher than a pre-defined threshold (Cabrerizo et al., 2010).

Different methods have been developed to assist the moderator to achieve group consensus. Herrera-Viedma et al. (2002) have proposed a model that is applicable for dealing with different preference structures in Multi-Person Decision Making (MPDM) problems. Choudury et al. (2006) has given an algorithm for solving advanced manufacturing technology problems. Ben-Ariehe and Easton (2007) introduced linear-time algorithms for problems in minimum cost consensus.
There are many consensus models using preference relations. Brysont (1996) proposed a framework that uses preference data to extract consensus relevant information. Under RGMM, Done et al (2010) gave two methods that are applicable of improving consensus indexes of PCM. Regarding fuzzy preference relations, Kacprzyk et al. (1992) proposed a model to generate the fuzzy majority by measuring the soft degree of consensus. In the context of linguistic, Herrera et al. (1997) presented a solution.

2.7.3. Selection Model
The selection model comprises two steps: aggregation (obtaining individual preferences) and exploitation (generating collective preference). Nowadays, many researchers in AHP-GDM are investigating the selection model (Dong et al, 2010). The challenge of this research area is how to aggregate individual information from more than one DM. The two renowned and useful methods are the aggregation of individual judgements (AIJ) and the aggregation of individual priorities (AIP) (Srdjevic and Srdjevic, 2013). They are explained below.

In order to solve that challenge effectively, there are three questions to be addressed, noted by Forman and Peniwati (1998):

1. How do you see the group? In other word, whether the group is treated as a unit (new individual) for reaching any agreement or the group is seen as a group as a collection of individuals holding different value systems. For the former an example is a group of executives who want to create a new organisational policy. An example of the latter could be a group of analysts and experts from different domains who are invited to analyse a company and to make a company-purchase decision. The use of AIJ or AIP is dependent on this question.

2. Which mathematical method is appropriate to use? Appropriate options are geometric mean and arithmetic means. The answer to this question is determined by the answer to the first one.

3. Should all individuals have an equal weight? In many circumstances where there is no equality, how to weight their judgements is a huge issue for the hand of the group. The performance of a group preference strongly associates with the answers to the question above.

Arithmetic and Geometric Mean
Before going to the methodology of AIJ and AIP, introductions of the Pareto principle, the arithmetic mean (i.e. mean or known as mathematical average) and geometric mean are needed.

James and James (1968) defined arithmetic mean and geometric mean as:

\[
\text{The arithmetic average of two numbers is the middle term in an arithmetic progression of three terms including the two given numbers.}
\]

\[
\text{The geometric average of two numbers is the middle term in a geometric progression of three terms including the two given numbers.}
\]

Thus, the arithmetic mean of 4 and 64, for example, is 34 since two numbers have the same interval 30 to 34. On the other hand, the geometric mean of 4 and 64 is 16.
since they have the equal ratio of $4$. In PriEsT the geometric mean (GM) method is used.

The Pareto principle or unanimity/agreement principle says that once there exists two alternatives $A_1$ and $A_2$, a group has to prefer $A_1$ than $A_2$ if all members within the group prefer $A_1$ (Ramanathan and Ganesh, 1994). This principle is one of four widely recognised social choice axioms.

**AIJ and AIP**

In AIJ, individual judgements are lost with aggregation occurs every time, and finally the group preference is generated (Forman and Peniwati, 1998). This method is used if the group is like a unit. Everyone in the group checks all individual judgements and the group may ask anyone to modify the judgements if they are of high inconsistence. Furthermore, some judgements should be discarded from a cluster after group agreement.

Assume there is a set of DMs $D = \{d_1, d_2, d_3, \ldots, d_m\}$, and the corresponding weight vector of them $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$, where $\lambda_k > 0, k = 1, 2, \ldots, n, \sum_{k=1}^{m} \lambda_k = 1$. Let the $A^{(k)} = (a_{ij}^{(k)})_{n \times n}$ be the matrix of DM $d_k (k = 1, 2, \ldots, m)$. For AIJ, the target is to obtain a collective matrix, $A^{c} = (a_{ij}^{(c)})_{n \times n}$, where

$$a_{ij}^{(c)} = \prod_{k=1}^{m} (a_{ij}^{(k)})^{\lambda_k} \quad (Dong \ et \ al., \ 2010)$$

The next stage will be using one prioritisation method to generate a priority vector $w^{(c)} = (w_1^{(c)}, w_2^{(c)}, \ldots, w_n^{(c)})^T$ from $A^{(c)}$ to rank the alternatives.

However, AIJ violates the Pareto principle (Ramanathan and Ganesh, 1994). They also suggested the utilisation of AIP in GDM. In AIP, each DM has right to assign value and get a personal resulting preference (Forman and Peniwati, 1998). All final preference vectors are concerned and then computed using either geometric mean or arithmetic mean method.

After using one prioritisation method, assume the individual priority vector $w^{(k)} = (w_1^{(k)}, w_2^{(k)}, \ldots, w_n^{(k)})^T$. Then the group priority vector can be denoted as $w^{(c)} = (w_1^{(c)}, w_2^{(c)}, \ldots, w_n^{(c)})^T$, where

$$w_i^{(c)} = \frac{\prod_{k=1}^{m} (w_i^{(k)})^{\lambda_k}}{\sum_{i=1}^{n} \prod_{k=1}^{m} (w_i^{(k)})^{\lambda_k}} \quad (Dong \ et \ al., \ 2010)$$

**2.8. Software Products**

This chapter introduces an assessment model to evaluate the SA and GDM feature based on their general functional requirements. It initially gives the marking schemes retrieving comparing criteria from previous background research. Then, the marking schemes are used to score four different educational and commercial tools: PriEsT 2, V.I.S.A, Web-HIRPE and IDS.

**2.8.1. Marking Scheme**

Based on the assessment model by Leonelli (2012), ideally a SA enabled DSS tool is able to deal with three forms of SA problems, to highlight the most sensitive criterion and to indicate the changes simultaneously occur to a number of criteria. In
addition, the capability of performing SA in different number of criteria (or alternatives) is also an important measurement. Hence, the requirements for assessing SA are summarised in the upper half of Table 3, from R1 to R6.

By considering GDM issues in section 2.7, the properties for assessing the GDM features are collected in Table 3, from R7 to R11.

**Table 3: Marking scheme for the evaluation of SA and GDM feature**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Properties Name</th>
<th>Value Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Numerical incremental analysis</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>Probabilistic Simulation</td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>Mathematical Models</td>
<td></td>
</tr>
<tr>
<td>R4</td>
<td>Most sensitive criteria</td>
<td></td>
</tr>
<tr>
<td>R5</td>
<td>Simultaneous changes</td>
<td>0: Feature unavailable</td>
</tr>
<tr>
<td>R6</td>
<td>The number of comparing elements is changeable</td>
<td>1: Partial available</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3: Almost acceptable</td>
</tr>
<tr>
<td>R7</td>
<td>Different selection methods</td>
<td>5: Excellent</td>
</tr>
<tr>
<td>R8</td>
<td>A range of prioritisation or weighting methods</td>
<td></td>
</tr>
<tr>
<td>R9</td>
<td>Various environment settings</td>
<td></td>
</tr>
<tr>
<td>R10</td>
<td>Group consensus support</td>
<td></td>
</tr>
<tr>
<td>R11</td>
<td>Portability</td>
<td></td>
</tr>
</tbody>
</table>

### 2.8.2. Evaluation of Software

The previous section presents the marking scheme with which is used to score each software product in the following by matching their features.

**PriEsT 2**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>It has been realised in PriEsT 2. Due to algorithmic simplicity, this feature always operates fast. The performance of graphs produced is perfect. However, the graph might become a mess if a large number of alternatives are considered.</td>
<td>3</td>
</tr>
<tr>
<td>R2</td>
<td>All decision elements, from any level of the hierarchy, can be considered. The decision problems are simplified since the simulation indicates the alternatives that never perform well. Overall, it is fairly nice although the response time is longer than numeric one: one-at-a-time</td>
<td>5</td>
</tr>
<tr>
<td>R3</td>
<td>The interface is excellent. Low dimensional (one to three) transformations are available with good models. However, multi-dimensional transformation has not been implemented.</td>
<td>3</td>
</tr>
<tr>
<td>R4</td>
<td>The most sensitive element can be identified once the lowest OPSC and TSC are obtained.</td>
<td>5</td>
</tr>
<tr>
<td>R5</td>
<td>Every time the weight of a criterion changes, the scores and final rankings of all alternatives changes accordingly.</td>
<td>3</td>
</tr>
<tr>
<td>R6</td>
<td>The user can choose to compare quite a large number of criteria or alternatives every time.</td>
<td>5</td>
</tr>
</tbody>
</table>
R7, R9-R11 | Not implemented in PriEsT 2 | 0
R8 | This feature is individually marked from other GDM ones. PriEsT has more than 10 elicitation methods available. They are usable after the realisation of the GDM features. | 5

**Total: 29**

The descriptions of R1 to R5 in this table are mainly adapted from Leonelli (2012).

**V.I.S.A**

V.I.S.A is a commercial DSS providing stand-alone and web-based application (SIMUL8 Corporation, 2013). Compared to other three, the layout of V.I.S.A. is more straightway and the learning curve is flatter.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Explanation</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>V.I.S.A well supports this feature. It has a <em>profile across tree</em> representing all criteria as bars where each is surrounded by a number of cycles as alternatives.</td>
<td>5</td>
</tr>
<tr>
<td>R2</td>
<td>V.I.S.A does not offer this function</td>
<td>0</td>
</tr>
<tr>
<td>R3</td>
<td>It only depicts points in a XY performance chart where only two criteria are compared each time</td>
<td>1</td>
</tr>
<tr>
<td>R4</td>
<td>It does not indicate which one is the most sensitive</td>
<td>0</td>
</tr>
<tr>
<td>R5</td>
<td>Every time the weight of a criteria changes, the overall profile with strings and bars and scores in bar chart form will instantly change. The changes are very concise.</td>
<td>5</td>
</tr>
<tr>
<td>R6</td>
<td>V.I.S.A is able to compare any finite number of criteria or alternatives within a finite time.</td>
<td>5</td>
</tr>
<tr>
<td>R7</td>
<td>Limited description.</td>
<td>3+3</td>
</tr>
<tr>
<td>R8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R11</td>
<td>V.I.S.A is easily accessible online. It also has offline version. The result or data generated from one version is also usable in another.</td>
<td>5</td>
</tr>
</tbody>
</table>

**Total: 27**

**Web-HIPRE**

It is a Java applet version of a well-known tool, HIPRE 3+. As the first online decision tool (System Analysis Laboratory, 2007), critically, doing sensitivity analysis is not the strength of Web-HIPRE. In contrast, it provides a convenient way to link any web pages together, which improves the efficiency enormously. People can easily find references and more information about some elements (goal, criteria and alternatives) in Web-HIPRE simply by clicking any node in the decision tree.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Explanation</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Web-HIPRE offers this feature by a bar chart or the way similar to the one in Figure 2. While the weight is changing, the graph records the original one. The relative importance of DMs can also be analysed.</td>
<td>5</td>
</tr>
</tbody>
</table>
IDS

Intelligent Decision Support (IDS) is developed for educational use. It is a powerful DSS with respect to a decision problem modelling and prioritisation method. It offers the SA feature whereas it does not support GDM.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Explanation</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>It has a wide range of methods (partial value function and utility curve, and additive value function approach, etc.). Detail settings are available, although the graphs are a little small.</td>
<td>5</td>
</tr>
<tr>
<td>R2-R4</td>
<td>Not support for these features.</td>
<td>0</td>
</tr>
<tr>
<td>R5</td>
<td>Simultaneous changes are shown by the increase or decrease of the heights of bars, each of which can represent a criterion or alternative.</td>
<td>5</td>
</tr>
<tr>
<td>R6</td>
<td>It is simple to choose no matter how many criteria to be compared, depends on the number of inputs. There is a 'select all' option. Ticking one each time is also available.</td>
<td>5</td>
</tr>
<tr>
<td>R7, R9-R11</td>
<td>No available in IDS</td>
<td>0</td>
</tr>
<tr>
<td>R8</td>
<td>There are some prioritisation methods. It also provides a few weighting methods such as swing weighting, direct assignment, and PC.</td>
<td>3</td>
</tr>
</tbody>
</table>

**Total: 18**
2.9. **Summary**

A series of findings from background research shows that the performance of GDM feature associates with other techniques. First, solving a GDM problem ultimately returns to the computation of a single PCM. For this reason, a low consistency index is a fatal contributor. In addition, using AHP (PC) to address GDM problems has been proved both effective and efficient with the help of a large number of existing algorithms and models. In addition, a GDM-DSS with a number of prioritisation methods provides more choices for DMs to elicit a group preference vector. Furthermore, a well-developed SA enabled system can also offer valuable suggestions in terms of weights setting to all DMs.

Currently, although PriEsT does not support GDM, it performs at a reasonable level among the DSSs selected, due to its well support for different forms of SA problems. It is obvious that it has huge potential to be a comprehensive and powerful tool after the integration of GDM. It is because it provides more than 10 types of elicitation methods including two new approaches.
3. Analysis and Requirements
As stated in Section 1.1, the primary aim of this project is to add GDM feature into PriEsT. Based on the objectives and deliverables defined, this chapter describes specifies project requirements in terms of realisation of new GDM features. This chapter begins with a review of PriEsT and identifies the need to add GDM. Finally, the definition of requirements for this enhancement is given.

3.1. Current PriEsT

3.1.1. Functions and Architecture
PriEsT 2 currently offers a variety of functions. The features can be summarised as follows:

- Detection of inconsistent judgements and decision aid for users
- Different prioritisation method available to choose from, namely GM, EV, NCS, EAST, CRO, DLS, WLS, LLAV, LLS, FPP, TOP and PrInT
- Provision of solution list
- Gantt chart and equaliser view of weight or ranking
- Two-objective space
- XML support and Swing based UI
- Support of the three SA algorithms

(Leonelli, 2012) and (Siraj, 2012d)

In addition, the software has some fundamental operations such as ‘Import’, ‘Export’ and ‘About’ on the toolbar.

3.1.2. Analysis of Dependency and Quality of Code
To adapt the current system into a web service system, a dependency analysis has been conducted to analyse the compile-time class dependency that might cause cyclic dependencies between server classes and client classes.

Current PriEsT contains 60 logic classes (excluding testing and UI classes). All the classes in util, random, geom and algorithm package rely on only fundamental Java packages. In addition, 5 objective classes, 2 io classes, 2 sensitivity classes and 1 core class are classes with high cohesion. Therefore, they are directly reusable. The dependency analysis of the other 35 classes having potential risks is given in Figure 6.

The analysis found two problems. One is the mutual dependency. For example, PCMatrix class and five classes in Property package are in the relationship of mutual dependency. ResultList class in Core package depends on AbstractMethod class, and the reverse is true. The other is the disordered dependency between the higher and lower layer. The XML adapter class, locating in the foundation layer, relies on Method package.

---

1 More information about PriEsT 2 is in (Leonelli, 2012)
In terms of the code, although there are many warnings (see Figure 7 to Figure 10), it is well structured and highly modularised. In addition to that, the detailed annotations make reuse work easily. The findings provided by two analysis tools (CheckStyle and PMD) show that the main issues of the code include ShortVariable, LocalVariableCouldBeFinal, LineLengthTooLong etc. They are not the significant issues.
3.2. Functional Requirements

Based on the previous analysis, PriEsT 2 is a complete product and a new GDM module is the only module required to add after the resigning. The specific functional and non-functional requirements for building GDM are described after the concerning factors in the following sections.

3.2.1. Concerning Factors

Through an analysis of background research, a variety of concerning factors with regard to GDM applications are concluded as follows:

- The group member characteristics – People who are more persuasive, powerful or confident heavily influence decision teams. The worst situation is that they are not domain experts or specialists.
- Group size – large group tends to have a greater amount of knowledge and less personal biases and accordingly to make effective decisions. However, large group size results in higher complexity and larger consumption of computation time. In addition, some individuals are more likely to contribute less.
- Variety of methods available - whether the software has multiple techniques to assign values for all alternatives, such as direct ranking, direct voting, pairwise comparison etc.; whether the software supports a variety of prioritisation methods.
- Time to submit decision – It is given to each member to create an individual PCM before submitting it. For each DM, when the time is up, what action should the system carries out: 1) the DM is treated as giving up his right of voting in this session and all the data submitted will be cancelled; or 2) the system automatically retrieves the data that has been already filled in.
- View group result statistics – After each round of voting/ direct value assigning, each member within the group is able to view the outcome generated according to group contribution.

3.2.2. Functional Requirements

GDM is able to assist teams to achieve their goals by encouraging interactions among team members. That means the two most important functional requirements of a GDM enabled system are allowing multiple user access and providing group interaction mechanism. In addition, more detailed requirements are extracted from the concerning factors. All the initial functional requirements are summarised as follows:

1. Multiple user access – The system is able to tackle the access requests from a certain number of users.
2. Group consensus support – The software enables DMs to resubmit their judgements until they are said to be consistent by the facilitator.
3. The system has different selection methods – AIJ and AIP are two possible options, and the system should have both of them to aggregate individual PCMs from all DMs.
4. The use of a range of prioritisation or weighting methods – GDM module is able to use the existing prioritisation and weighting methods, and allows the system to satisfy different user demands and work in various conditions.
5. Portability – The software can be accessed by universal platforms including Web browsers.
6. Various environment settings – The facilitator can set the environment parameters for the group meetings, e.g., group size, thinking time.

3.2.3. Non-functional Requirements

In addition to functional requirements, the main non-functional requirements are defined as follows:

1. Robustness - The system is tolerant of errors or expected format data that are made by the user.
2. Usability – 1) the interaction is simple; 2) the interface is understandable: the structure and flow are in a logical and consistent manner, including icons, procedures etc.; 3) the user is always reminded the status of the system, therefore, and knows what to do next. 4) The text on the interfaces should be concise and navigatable (Adapted from Pressman (2010)).
3. Security – The system is equipped with the basic security mechanisms. The user who intends to join the group must show group ID and password defined by the facilitator.

3.3. Use Case Diagram

From the perspective of the software, all the actions of users that the system should offer are given in the user case diagram. The black frame represents the system boundary, while two types of users are outsiders who use the system.

![Use case diagram](image)
4. Progress
In software engineering, a typical software lifecycle contains several phases, including requirement, design, implementation, testing and evaluation. As the project requirements have been stated in the chapter 3, this chapter firstly describes the software redesign, architecture of GDM and the typical flow of GDM module. It then gives the UI prototype. Implementation and testing methodology is provided after that. Next, the project evaluation methods are identified. Finally, the progresses so far are concluded and the project plan is described.

4.1. Architecture of GDM
PriEsT 2 is structured as an MVC framework. Consequently, model building and UI design are two subjects for the project design and implementation. This section mainly covers the model part and architecture of GDM while the UI design is shown in section 4.3.

4.1.1. Redesign
An analysis of PriEsT in section 3.1 has reflected that although the current software is well structured, some structural changes are required to make it safer to be a web services system. A façade pattern (Gamma et al, 1994) is used to better satisfy the need to be a web service, and protect the current software from many potential architectural modifications. In other word, the current system will perform as a new unit communicating with the new module, and all the requests between them will go through a façade object. Visitor pattern might also be involved if an object needs new operations but the structure of the object should be stable.

In addition, some parts of the system should have further modularisations. Certain classes or even the methods are being separated into smaller sections so that they are more coherent and easily called by a central control deck. A prototype of a central control deck has been created to be a ‘facilitator’ leading the meetings. Thanks to the fact that PriEsT is in MVC structure, changes in the model do not influence the correspond interface. The current PriEsT system is the main part of a facilitator object or a participant object.

Oracle (2013) suggests two ways to create web services starting from Java code or starting from a WSDL file. Thanks to the existing Java classes, the first method is chosen, simply starting by coding the endpoints in the source code. Decision Deck (2013) also provide plenty of available XMCDA web services on the website.

4.1.2. Architecture of New Module
The new module includes three components: 1) Users’ Authentication and Access, 2) Controller and Meeting Management, and 3) Data Receiver System, see Figure 12. An analysis of PriEsT in section 3.1 has reflected that although the current software is well structured, some structural changes are required to make it safer to be a web services system. A façade pattern (Gamma et al, 1994) is used to better satisfy the need to be a web service, and protect the current software from many potential architectural modifications. In other word, the current system will perform as a new unit communicating with the new module, and all the requests between them will go through a façade object. Visitor pattern might also be involved if an object needs new operations but the structure of the object should be stable.

In addition, some parts of the system should have further modularisations. Certain classes or even the methods are being separated into smaller sections so that they are more coherent and easily called by a central control deck. A prototype of a central control deck has been created to be a ‘facilitator’ leading the meetings. Thanks to the fact that PriEsT is in MVC structure, changes in the model do not influence the correspond interface. The current PriEsT system is the main part of a facilitator object or a participant object.

Oracle (2013) suggests two ways to create web services starting from Java code or starting from a WSDL file. Thanks to the existing Java classes, the first method is chosen, simply starting by coding the endpoints in the source code. Decision Deck (2013) also provide plenty of available XMCDA web services on the website.

4.1.2. Architecture of New Module
The new module includes three components: 1) Users’ Authentication and Access, 2) Controller and Meeting Management, and 3) Data Receiver System, see Figure 12. The most difficult part could be the design and implementation of the controller panel. Normally there are two means to realise a facilitator, or a control panel from the software perspective. The first method centralises the facilitator (C), executing all operations including the computation of individual consistencies, group consensus, aggregating the overall PCM, and generating the final preference vector. This method is more suitable for a small scale GDM. In the second method, there will be a series of distributed sub-controller (C1, C2, ..., Cn). In this case, the sub-controllers are pre-processors posting processed data to the central controller (C). In this project, the first method has been chosen.
The reason is that in many numerical examples from case studies (Dong et al., 2010; Wu and Xu, 2012) five DMs are assumed to participate in making group decisions.

Adapting existing application to support XMCDA is also an important redesign task. The completeness of that will allow easy interactions among different MCDM algorithms. The clients (participants) send SOAP requests, with the help of a browser (or mobile phone), to use the services provided by the server (the facilitator). Data from DMs will be packed into a XMCDA-formatted file before being sent to the facilitator. The data receiver system includes an XMCDA adapter (Figure 12). The formatted file generated is exportable and storable into a document. It can be treated as a reference for future use.

![Architecture of the enhanced PriEsT](image)

**Figure 12: Architecture of the enhanced PriEsT**

(*PriEsT 2 is the main part of a facilitator or a participant object*)

### 4.2. Logic of GDM

This section gives the main logic of GDM. Figure 15 displays the standard process for the system to make a group decision, based on the framework mentioned in 2.7. In the group meeting, a facilitator is responsible for one round, i.e. the generation of one group PCM and the storage issue followed. The facilitator can keep in charge in the next meeting unless he cancels the meeting or one third of joiners quit the meeting.

As mentioned in 3.2.2, group rules setting is indispensable for running a team. Once a group is created, the facilitator defines the rules for the system to follow in the next one round. The facilitator can store and export rules for future use. The rules include:

- **Equality or in-balance mode**: the facilitator has to define whether all the DM's PCM are equally considered with the same weight. For example, should the chairperson in a group have higher weight?
• **The maximum size of the group**: the size of the group is defined before each meeting to keep the number of participants stable during the entire decision making procedure.

• **GDM algorithm to use**: System will allow two methods: AIJ and AIP. The default setting would be the latter.

• **Prioritisation method to use**: PriEsT has already supported more than 10 types of prioritisation methods. The facilitator could choose one or multiple ones.

• **Longest waiting time for each DM**: This setting comes after an algorithm selection. The waiting time should be longer under the AIP mode than AIJ one as each DM needs more time to have final personal rankings.

• **Consistency index value**: The facilitator sets the threshold value to control the degree of consistency. Normally the value is 0.1.

• **Discard mode of recommendation mode**: Sometimes people do not submit their judgements on time or the judgements are uncompleted, the system will automatically choose how to deal with their PCMs. If discard mode is selected, the system will ignore all the data sent and will be sent from the DMs. Otherwise, the system fills all the vacancies itself.

4.3. **Interface Design**

The interface of GDM module uses the template of the existing software, instead of creating an isolated frame. Similar to SA, the new module shares a tabbed pane with ‘Decision Aid’ and ‘Elicitation’. The user is able to switch between a group of functions by clicking the titled tabs. Once selecting the GDM tab, the user then requires choosing a role, whether to be a facilitator (Figure 13) or participant (Figure 14). The meanings of main components are as follows:

• **Start**: Click the button to start the group meeting with participants joined.

• **Progress**: Show the entire progress of the group meeting

• **Participant IP**: Record the IP address of a joiner

• **Status**: The possible status of a participant can be: ‘invitation not sent yet’, ‘invitation sent’, ‘waiting to join’, ‘submitting data’, ‘finished’ and ‘disconnected’.

• **Group setting panel**: See the detailed explanations in 4.2.

• **Function Panel**: Provide other functions, including ‘continue to the next round of meeting’, ‘check the consistency of PCM generated’, ‘send the result to all participants’ and ‘quit the group’.

As mentioned previously, the project requirements might be revised. The UIs of the GDM aspect of PriEsT will be accordingly evolved, following the steps below:

1) Extract information from the revised project requirements (both functional and non-functional) and optimise it if necessary.

2) Update the UI with layout and Java Swing elements, e.g. panel, images.

3) Map all interface actions to potential user actions, and bear the processing logic in mind.

4) Identify objects of interface elements that associate with the models behind.

5) Rename all the variables to be meaningful and refine interface layout.
4.4. Implementation Environment and Tools
Software environment during the project includes the following aspects:

- **Language and platform**: PriEsT 2 is written in Java using Netbeans IDE. Therefore, NetBeans IDE 7.0.1 or 7.2 is used as the main platform for this
enhancement project. Extra plugin tools, such as Android SDK, the lightweight user interface toolkit (LWUIT), and Sun Wireless Toolkit will be integrated into the platform to facilitate the implementation process.

- **Server:** GlassFish open source edition 3.1 has been chosen as it is the built-in server of NetBeans.
- **XMCDA library:** J-XMCDA library is necessary and it allows MCDA objects to be stored into or read from XMCDA files. The library also offer XMCDA web services support. In addition, XMLBeans and JDOM will be used to manipulate XML objects.
- **Web services support:** WSDL and SOAP are two basic elements for web services. Thus, JAX-WS 2.0 (JSR 224) and SOAP AIP will be supportive.
- **Mobile support:** Java ME SDK 3.3 and Android SDK are necessary to make the enhanced PriEsT available for Java ME and Android devices. LUWIT toolkit will be helpful for the creation of the swing-like interfaces on the mobile PriEsT.

### 4.5. Implementation and Testing Methodology

The aim of this project is to enable PriEsT to conduct GDM. The current specific requirements are defined in section 3.2. However, those requirements might change as the project progresses, e.g. Modifying the interface on mobile version due to the limit of screen size. A more significant reason is the characteristics of the project that it should be completed in sequence. The GDM feature must be implemented on the desktop version before being transplanted to mobile platform, while the redesign of PriEsT is a prerequisite.

Because of these reasons, the entire enhancement process follows the incremental model (see Figure 16). For example, the realisation of AIP comes earlier than that of AIJ due to its relative simplicity. Three iterations are needed and each one flows in Waterfall method, covering design, implementation and test phase. Section 1.2 defines the deliverables in each interaction. Some best practice from agile methodology will be involved, such as continuous integration and test-driven development (TDD).

![Figure 16: The incremental process model (Pressman, 2010)](image-url)
In the testing process, each unit test is conducted after every time a section of component-level code. In the OO context, instead of testing operations/methods in isolation, each operation will also be tested in other subclasses where it runs.

Since GDM module will have a number of conditional statements, every branch of the module should be fully tested. For instance, in the context of inviting participants, the system should check whether the IP address is eligible, whether the invitation is sent, whether the request is accepted and whether the user has joined the group. Due to this reason, basic Path testing, invented by Tom McCabe in 1976, conducts to ensure every statement in a test case has been executed at least once, based on any possible branches.

In addition, a series of architecture tests are necessary to test interactivity between the older version and new modules. A example could be an attempt of a participant object to visit a prioritisation method object in the legacy system.

4.6. Project Evaluation

The project will perform three activities to realise GDM features: the redesign process of the PriEsT 2, implementation of the GDM module and implementation of mobile PriEST. Two main tasks are required to evaluate these activities: evaluation of the web version system and evaluation of the mobile version system.

4.6.1. Evaluation of Web Version

Before the evaluation, the two GDM algorithms (section 2.7.3) and other requirements (section 3.2.2) must be implemented. The concerning factors in section 3.2.1 will also be taken into account. The first step of this evaluation task is to verify the fulfilment of the requirements, with the help of case studies from literature. Then a performance test is conducted.

In order to evaluate the correctness of AIJ, the numerical data in (Done et al, 2010) will be used to verify the output of the algorithms. The case study uses RGMM (LLS) prioritisation method. The data model in (Wu and Xu, 2012) can be used to evaluate both AIJ and AIP algorithms. For this case, the option of prioritisation method will be EV. Both of the case studies assume there are five DMs engaged in making a group decision. One computer might runs multiple instances of the system in different Web browsers (e.g. IE/ Chrome, Firefox/Safari). Every instance is differentiated in terms of name and IP address.

For the performance test of the GDM module, two Monte-Carlo simulations will be executed to generate a range of random decision objects with ranging hierarchy structures. The number of DMs is fixed at 5, suggested by two authors before. The first simulation creates n criteria (3<n<9) and 9 alternatives in each iteration. The maximum value of n is 9 as Satty (1977) stated that comparing more than 7 (plus or minus 2) options is difficult for human beings due to brain limitation. The second simulation will use the reverse setting: 9 criteria and n alternatives (3<n<9). The findings from the simulations will be used to identify the system performance (e.g. the average execution time and precision ratio of the algorithms) versus problem scale.

4.6.2. Evaluation of Mobile Version

Once the mobile PriEST is realised, the second evaluation will start. It verifies the completion of the requirements in the mobile platform. Again, the requirements are
given in section 3.2.2 and the testing data is retrieved from the case studies mentioned. The results from this evaluation will be compared with those from Web version. Due to the limitation of screen size, the interfaces or some functions on mobile platform will be constrained.

Similar in the performance test, the system computation time and algorithm error ratio will be evaluated using the identical simulation settings. The hardware difference might contribute to the diversity of performance.

4.7. Progress
Progress on the project has been summarised in the table as follows:

Table 4: Progress

<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A literature research has completed. The subjects contained include MCDM, XMCDA, PC, AHP, SA, prioritisation methods, GDM.</td>
</tr>
<tr>
<td>2</td>
<td>An evaluation framework for SA and GDM functional features has created.</td>
</tr>
<tr>
<td>3</td>
<td>Several similar DSSs have been analysed. Three of them are selected to compare with PriESt 2.0 using the evaluation framework.</td>
</tr>
<tr>
<td>4</td>
<td>A full analysis of the source code of PriESt 2.0 has been completed.</td>
</tr>
<tr>
<td>5</td>
<td>The initial functional and non-functional requirements for the project have been defined.</td>
</tr>
<tr>
<td>6</td>
<td>A prototype of control panel with main group interactive logics has been developed.</td>
</tr>
<tr>
<td>7</td>
<td>The architecture is being modified to fit the need of XMCDA standard and GDM feature.</td>
</tr>
</tbody>
</table>

4.8. Project Plan
The Gantt chart displays the plan for the entire project. The task in the blank frame means the exam period. However, the project will still be in progress from 17th May to 28th May.

![Figure 17: Gantt chart for project plan](image-url)
5. Reference


Gamma, E., Helm, R., Johnson, R. & Vlissides, J. *Design patterns: Elements of Reusable Object-Oriented Software*: Addison-Wesley.


