Implementation of utility calculations based on utility difference proportions

Chuye Xiao & Xi Jiang
June 2012

Bachelor Thesis, 15 credits, C
Computer Science

Computer Science program
Examiner: Ulla Ahonen-Jonnarth
Supervisor: Magnus Hjelmbloom
Implementation of utility calculations based on utility difference proportions

By

Chuye Xiao & Xi Jiang

Faculty of Engineering and Sustainable Development
University of Gävle

S-801 76 Gävle, Sweden

Email:
ofk09cxo@student.hig.se
ofk09xjg@student.hig.se

Abstract

This pilot study presents an implementation of utility calculation that is based on utility difference proportions. To conduct a utility calculation for a consequence, one needs to evaluate utility difference proportions and to calculate utility values. The application can help decision makers to compare consequences pairwise and make decisions easier regarding to the final output utility value for each consequence. On the other hand, the application is limited to a specific case with specified preference ordering. The application can be further developed into a more real interactive analytic decision-making support tool in the future.

Keyword: Decision making tool, Multi-criteria decision analysis, Utility difference proportion, Utility value
1 Introduction

We human beings are naturally decision makers who make decisions every day, even every minute. Making a decision can be very easy and it is also possible to get good results only based on one single criterion or only on intuition. The way of dealing with decision problems by only trusting your feelings or intuitions can be regarded as a holistic approach. Another way of dealing with decision problems is the analytic approach. We start with outlining our goal, then we evaluate different alternatives, choose a specific approach and take actions. This is only a simple procedure to make decisions. When we encounter a problem involving multiple criteria and these criteria may even conflict with each other, the analytic approach can help us to properly structure the problem and explicitly evaluate these criteria. For a daily-life example, in order to buy a car that satisfies the customer, not only the price, but also the style, the gas consumption, the reliability of vehicle and many other aspects can be considered. It is not likely to get the car at the lowest price that is also the safest and most comfortable one.

In order to reduce the complexity in decision problems and to solve the problems better, there is a very widely used discipline in the decision making environment: multi-criteria decision analysis. Multi-criteria decision analysis (MCDA) handles two or more conflicting goals [1]. It provides alternatives ordered from the most preferred to the least preferred and it helps us to structure our decision problem clearer and therefore helps us to solve complex problems and make better decisions. One of the most popular MCDA methods is analytic hierarchy process (AHP) which has been developed by Thomas L. Saaty in the 1970s. It has attracted strong interests among a lot of people mainly due to “the nice mathematical properties of the method and the fact that the required input data are rather easy to obtain” [2].

1.1 Study Focus

The aim of this work is to contribute to the foundation for a new interactive analytic decision-making support tool based on proportionality and additivity of utility measures. This pilot study focuses on how to represent proportions between utility differences between specific consequences in a computer program, and how to use these proportions to perform calculations of utility values between consequences. The long term goal of developing this application is to let users make pairwise comparisons of utility differences, to determine the proportions between them and make it a more real interactive analytic decision-making support tool. Figure 1 sketches the idea behind this application.
1.2 Study Problems

1. How should the proportions between pairs of utility differences be represented?
2. How to use utility measure for the calculation of utility values?
3. How can the calculation of utility values be performed in the program?

1.3 Delimitations

Here we state some delimitations of our application before we start to demonstrate scientific backgrounds. First we have made specific assumptions saying that: In the application, we have two aspects and three consequences for the decision problem. Second, we have defined the specific preference ordering between consequences. For instance, we assumed that with respect with aspect 1: consequence a is better than both consequence b and consequence c, and consequence b is better than consequence c. More details will be explained later in section 2.
2 Scientific Background

The background section will present the background theory or related work relevant for this thesis.

2.1 Utility measures

To start with, we give a brief definition for three levels of utility measures [3].

The first is ordinal scales: a measure with ordinal scales is ordered in the way that higher numbers indicate higher values or higher qualities. Note that the intervals between the numbers do not represent the difference of degree. For example, if there is a ten-point rating scale when measuring attitudes on restaurant quality, the difference between rating 5 and rating 6 may not represent the same difference as the difference between rating 8 and rating 9.

The second is interval scales: measures the same way as ordinal scale except that the interval between the numbers represents the difference of degree.

The last one is ratio scales: show different kinds of relative measures of relationship between objects.

2.2 Multiple Criteria Decision Analysis (MCDA)

Multi-criteria decision analysis (MCDA) is a general idea in order to solve complex decision problems that combine different objectives. It subdivides the decision problem into more manageable pieces, which means that these pieces can be easily used to deal with data and judgments. MCDA can provide an approach to reassign these pieces in order to complete the decision making process [4]. Belton and Stewart [5] also give a definition of multi-criteria decision analysis. They summarize that different choices or courses of actions will constitute a MCDA problem, when there are lots of substantial conflicting criteria. The steps to apply MCDA can be generalized as: first of all, the decision problem context needs to be established. Then the options must be identified, as well as the objectives and criteria. The next step is to evaluate the potential consequences (i.e. the possible outcome or result) of each option with respect to each criterion. Closely connected to this step is establishing the relative importance of the different criteria. Then the MCDA process combines the weight and the consequences together in order to calculate the ultimate utility values. Decision makers can perform sensitive analysis[4]. However, MCDA is not only an approach to solve complex decision problems, but also a set of techniques to order the possible options from the most preferred to the least preferred. This order is based on how decision makers prefer the different outcomes. Only one outcome will occur. Here we give a brief clarification about the relationship between an option and an outcome. Options are regarded as a set of potential actions that can be chosen whereas outcomes are the results of the chosen options. For example, buying a lottery can be regarded as one option; winning or failing are the certain outcomes for this option while nothing happening is the outcome for not buying lottery. The reason why MCDA can be used in a broad field, which contains both private and public areas, is because it is a set of techniques; MCDA offers many different ways to resolve complex decision problems.

Let us consider a simple example, when we want to buy a car, involving consequences a and b (a could for instance mean buying Volvo and b could mean buying Saab), as well as two aspects, (aspect 1 could for example mean fuel consumption of the car and aspect 2 could mean safety of the car). Then let us assume that a with respect to aspect 1 is better than b, and b with respect to aspect 2 is better than a. Furthermore, let us assume that the utility of consequence a and
consequence b with respect to aspect1 and aspect 2 are measured on interval scales. Note that this means that the differences in utility from a to b in each aspect are measured on ratio scales. Figure 2 shows the above situation.

![Figure 2](image)

*Figure 2: A simple example involving 2 consequences and 2 aspects in MCDA*

As the figure 2 shows, in this multi-criteria decision analysis problem example, m₀ represents the utility value all aspects considered. (a₁, b₁) denotes difference in utility from consequence a to consequence b with respect to the first aspect; (b₂, a₂) denotes difference in utility from consequence b to consequence a with respect to the second aspect. Figure 2 shows a structure of our decision problem in a clear way.

Note that if a is better than b in aspect 1, then (a₁, b₁) is positive. In that case, a lies above b on the corresponding arrow.

![Figure 3](image)

*Figure 3: A more complex example involving 3 consequences and 2 aspects in MCDA*

Figure 3 illustrates a more complex multi-criteria decision problem with three consequences and two aspects. The notation is the same as in the first example. With these aggregation trees, we can structure the decision problem clearly.
We conclude that the key benefits of MCDA are: first, it emphasizes the importance of establishing values and criteria by the decision-makers; second, it incorporates these values into the decision-making process explicitly; third, it can be used to evaluate the importance of specific values and specific criteria to overall ranks or decisions through analysis. [6]

2.3 Analytic Hierarchy Process (AHP)

When we make decisions, there are often a number of alternatives and multiple criteria to evaluate.

Analytic Hierarchy Process, which was invented by Thomas L. Saaty in the 1970s has been widely used for this kind of problems.

The main procedures of AHP can be summarized as follows [7] (figure 4):
1. "Decomposing", this means to decompose the decision problem into goals, criteria and alternatives in a hierarchy structure.
2. "Weighing", each criterion will be assigned a concerned weight.
3. "Evaluating", in this step, the evaluation of alternatives and the pairwise comparison are performed respectively, here a pairwise comparison means a comparison of two criteria at one time.
4. "Selecting", which means that, after comparison, decision makers can select the best alternative that suits the goal best with the help of this process.

![Figure 4: A simple AHP hierarchy](image)

Saaty [9] states that analytic hierarchy process is a decision-making process which has a natural ability for solving problems based on reasonable analysis and evaluation. Power [10] also gives a definition of AHP, that it is a decision-making approach using hierarchy to structure the multiple criteria by looking at how important they are; comparing the alternatives they have and ranking these alternatives. For the AHP, Saaty [9] says that in order to reveal the primary elements and relationships, which create judgments based on decision makers’ feelings or emotions, the decision problem has to be structured well. To construct the priorities and to show the relative value of the criteria is the next step. For weighing the related priorities of the criteria, the AHP uses 1-9-point scale, which is between 1 (equal importance) and 9 (extreme importance). It is done by asking questions like “How important is criterion A with regard to criterion B?”, for example, if the importance is 1, then it represents equal importance, that the two criteria contribute equally to the goal. If the importance is 9, then it represents extreme importance, that one criterion is strongly preferred over the other. Huw [11] claims that AHP is especially useful for solving complex decision problems and also very suitable for comparing different elements which are difficult to quantify.

Some software has been developed based on AHP, such as Expert Choice, which has been the leading AHP tool for the past two decades. Expert Choice provides an effective way to divide a
decision problem into many small sub-parts, also structure the total goal to sub-objectives, then simplify these sub-objectives into alternative courses of action. It will be easy for the decision makers to compare one alternative another based on the well-structured hierarchy system [12]. Therefore, it will display a suitable solution in order to achieve the goals.

To conclude the popularities of AHP: it uses pairwise comparisons that can provide decision makers a very straightforward and convenient way to input data; it also divides the decision problem into constituent parts and structure hierarchy of criteria which make the importance of each criterion relevant. In addition, AHP has an ability to check consistency of evaluation measures and alternatives.

On the other hand, there exists some criticism on analytic hierarchy process (AHP), such as the limitations of using 1-9-point scale weighing, the way to measure inconsistency, the problem of rank reversal.

2.4 Theoretical Foundation for a new MCDA Tool

This section will present a summary of theoretical foundation written by Odelstad [13] for a new analytic decision-making support tool.

2.4.1 Difference proportions

The difference proportions are based on the proportions [13]. A proportion is a function from a set A×A to Re (the set of real numbers), i.e.

\[ \rho : A \times A \rightarrow \text{Re} \], fulfilling the following requirement that is called proportion condition: For a, b, c \in A,

\[ \rho(a, b) \cdot \rho(b, c) = \rho(a, c) \]  

(1)

If a proportion \( \rho \) is a function into \( \text{Re}^+ \) (the positive real numbers) we say that \( \rho \) is a positive proportion.

Before we come to the formal difference proportion definition, we first give a brief description of ‘diff-set’ by using an example.

Suppose that in one aspect of a decision, we have consequences a, b and c, and the preference ordering for consequences is \( a \succ b \succ c \). Then the diff-set is a set that contains all positive differences, in this case \( \langle a, b \rangle, \langle a, c \rangle, \langle b, c \rangle \), and here, the notation: \( a \succ b \) indicates the relationship that a is better than b.

Now let us come to the definition for difference proportion.

Suppose that D is a diff-set, then we say that \( \rho : D \times D \rightarrow \text{Re}^+ \) is a difference proportion on D if \( \rho \) is a proportion on D and the following holds for all \( \langle a, b \rangle, \langle b, c \rangle, \langle x, y \rangle \in D \),

\[ \rho(\langle a, b \rangle, \langle x, y \rangle) + \rho(\langle b, c \rangle, \langle x, y \rangle) = \rho(\langle a, c \rangle, \langle x, y \rangle) \]  

(2)

2.4.2 From utility difference proportions to utility measure

In this sub-section, we are going to present how we work with an example that we use in this study.

Before we come to the details of our example, let us present utility difference proportion matrices.

Note that we have made an assumption that utility values are measured on interval scale, and the utility difference proportions of aspects between evaluations are measured on ratio scale.

As an illustration, let us return to the first example which has been presented in section 2.2.
First, for simplicity, the consequences are represented by \( a, b \). It means that the number of consequences is 2. Thus \( a_1, b_2 \) represent consequence \( a \) in aspect 1 and consequence \( b \) in aspect 2 respectively. Then we express utility differences as \( \langle a_1, b_1 \rangle, \langle b_2, a_2 \rangle \).

In table 1 and table 2, there are the two intra-factorial proportion matrices for displaying utility difference proportions for different consequences within one aspect. Note that they are all 1x1 matrices where the first column and row is the name for the cells.

**Table 1. Intra-factorial proportion matrix in aspect 1**

<table>
<thead>
<tr>
<th>( \rho_1 )</th>
<th>( \langle a_1, b_1 \rangle )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \langle a_1, b_1 \rangle )</td>
<td>1</td>
</tr>
</tbody>
</table>

**Table 2. Intra-factorial proportion matrix in aspect 2**

<table>
<thead>
<tr>
<th>( \rho_2 )</th>
<th>( \langle b_2, a_2 \rangle )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \langle b_2, a_2 \rangle )</td>
<td>1</td>
</tr>
</tbody>
</table>

In table 3 and table 4, there are two 1x1 inter-factorial proportion matrices for displaying utility difference proportions for different consequences in different aspects. Note that they are the 1x1 matrices.

**Table 3: Inter-factorial proportion matrix for differences \( \langle a_1, b_1 \rangle \) and \( \langle b_2, a_2 \rangle \) from aspect 1 to aspect 2**

<table>
<thead>
<tr>
<th>( \rho )</th>
<th>( \langle b_2, a_2 \rangle )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \langle a_1, b_1 \rangle )</td>
<td>( \rho_1 )</td>
</tr>
</tbody>
</table>

**Table 4: Inter-factorial proportion matrix for differences \( \langle b_2, a_2 \rangle \) and \( \langle a_1, b_1 \rangle \) from aspect 2 to aspect 1**

<table>
<thead>
<tr>
<th>( \rho )</th>
<th>( \langle a_1, b_1 \rangle )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \langle b_2, a_2 \rangle )</td>
<td>( \rho_2 )</td>
</tr>
</tbody>
</table>

The interpretation of matrix in table 3 is \( \rho_1, 2 \langle a_1, b_1 \rangle, \langle a_2, b_2 \rangle \rangle = \rho_1 \)

Here, note that \( \rho_{2,1} = 1/\rho_{1,2} \), and the proof can be seen in [14].
Next let us come to the actual example case. Nowadays, cars have become the most convenient and common traffic tool. Meanwhile, there are a variety of cars that make people too dazzled to choose from.

Here is our case, Tom wants to buy a car and there are three candidate cars that suit his basic preferences; they are Saab, Volvo and Ford. Among these three candidate cars, the two aspects that he focused on most are: safety and fuel consumption. By going through the whole process, it will be easier for him to make a better decision about which car to buy.

The final output matrices contain the intra-factorial matrices (table 5 & 6) that display proportions in one aspect and the inter-factorial matrix (table 7) displays proportions in different aspects. As a result, the different consequences can be compared both in one aspect and different aspects. In this way, it provides a clear and logic representation for all the things considered.

We simplify the notation of each input element. As a result, buying Saab is represented by the letter "a", buying Volvo is represented by "b" and buying Ford is "c". Here a, b and c are the three consequences for the decision problem. On the other side, safety and fuel consumption are two aspects that need to be considered for this decision problem. In aspect safety, the three consequences that are represented as \( a_1, b_1, c_1 \) assuming that their preference order is \( a_1 > b_1 > c_1 \). While in aspect fuel consumption, the consequences are represented as \( a_2, b_2, c_2 \), with the preference ordering: \( b_2 > c_2 > a_2 \). Then \( \{ a_1, b_1 \} \) indicates the utility difference between consequence a and consequence b in aspect safety; and the rest can be done by analogy. Then we define \( D_1 \) as a diff-set that contains all the possible positive differences of the three cars in safety aspect: \( D_1 = \{ \{ a_1, b_1 \}, \{ a_1, c_1 \}, \{ b_1, c_1 \} \} \). \( D_2 \) can be represented in the same way: \( D_2 = \{ \{ b_2, a_2 \}, \{ c_2, a_2 \}, \{ b_2, c_2 \} \} \). Here we assume that \( p \) is the difference proportion in \( D_1 \cup D_2 \rightarrow \mathbb{R}^+ \).

We use particular letters \( (p,s,r) \) to denote utility difference proportions in intra-factorial matrix and inter-factorial matrix. According to this, the matrices can be generated as following in a similar way as in [15]:

**Table 5. Intra-factorial matrix of consequences in aspect 1**

<table>
<thead>
<tr>
<th>( p_1 )</th>
<th>( { a_1, b_1 } )</th>
<th>( { a_1, c_1 } )</th>
<th>( { b_1, c_1 } )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( { a_1, b_1 } )</td>
<td>1</td>
<td>( p_1 )</td>
<td>( p_1s_1 )</td>
</tr>
<tr>
<td>( { a_1, c_1 } )</td>
<td></td>
<td>1</td>
<td>( s_1 )</td>
</tr>
<tr>
<td>( { b_1, c_1 } )</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

**Table 6. Intra-factorial matrix of consequences in aspect 2**

<table>
<thead>
<tr>
<th>( p_2 )</th>
<th>( { b_2, a_2 } )</th>
<th>( { c_2, a_2 } )</th>
<th>( { b_2, c_2 } )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( { b_2, a_2 } )</td>
<td>1</td>
<td>( p_2 )</td>
<td>( p_2s_2 )</td>
</tr>
<tr>
<td>( { c_2, a_2 } )</td>
<td></td>
<td>1</td>
<td>( s_2 )</td>
</tr>
<tr>
<td>( { b_2, c_2 } )</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>
Table 7. Inter-factorial matrix of consequences in aspect 1 and aspect 2

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>$\langle b_2, a_2 \rangle$</th>
<th>$\langle c_2, a_2 \rangle$</th>
<th>$\langle b_2, c_2 \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle a_1, b_1 \rangle$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\langle a_1, c_1 \rangle$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\langle b_1, c_1 \rangle$</td>
<td></td>
<td>$r$</td>
<td></td>
</tr>
</tbody>
</table>

The following calculations are based on and cited from [15]:
Since $\rho$ is a difference-proportion, as mentioned above, we have:

$$\rho (\langle a_1, b_1 \rangle, \langle x,y \rangle) + \rho (\langle b_1, c_1 \rangle, \langle x,y \rangle) = \rho (\langle a_1, c_1 \rangle, \langle x,y \rangle)$$  \hspace{1cm} (3)

In particular, if $\langle x, y \rangle = \langle b_1, c_1 \rangle$, then:

$$\rho (\langle a_1, b_1 \rangle, \langle b_1, c_1 \rangle) + \rho (\langle b_1, c_1 \rangle, \langle b_1, c_1 \rangle) = \rho (\langle a_1, c_1 \rangle, \langle b_1, c_1 \rangle)$$  \hspace{1cm} (4)

From the equation above, we can conclude:

$$p_{1:51} + 1 = s_1$$  \hspace{1cm} (5)

Therefore:

$$s_1 = \frac{1}{1 - p_1}$$  \hspace{1cm} (6)

$$s_2 = \frac{1}{p_2 - 1}$$  \hspace{1cm} (7)

Hence, the intra-factorial matrices can be represented in a new way as shown in table 8 and table 9, where we can see how the difference proportions are related to each other:

Table 8. Intra-factorial matrix of consequences in aspect 1

<table>
<thead>
<tr>
<th>$\rho_1$</th>
<th>$\langle a_1, b_1 \rangle$</th>
<th>$\langle a_1, c_1 \rangle$</th>
<th>$\langle b_1, c_1 \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle a_1, b_1 \rangle$</td>
<td>$1$</td>
<td>$\rho_1$</td>
<td>$\frac{p_1}{1 - p_1}$</td>
</tr>
<tr>
<td>$\langle a_1, c_1 \rangle$</td>
<td></td>
<td>$1$</td>
<td>$\frac{1}{1 - p_1}$</td>
</tr>
<tr>
<td>$\langle b_1, c_1 \rangle$</td>
<td></td>
<td></td>
<td>$1$</td>
</tr>
</tbody>
</table>
Table 9. Intra-factorial matrix of consequences in aspect 2

<table>
<thead>
<tr>
<th>ρ</th>
<th>( \langle b_2, a_2 \rangle )</th>
<th>( \langle c_2, a_2 \rangle )</th>
<th>( \langle b_2, c_2 \rangle )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \langle b_2, a_2 \rangle )</td>
<td>1</td>
<td>ρ</td>
<td>( \frac{p^2}{p^2 - 1} )</td>
</tr>
<tr>
<td>( \langle c_2, a_2 \rangle )</td>
<td>1</td>
<td></td>
<td>( \frac{1}{p^2 - 1} )</td>
</tr>
<tr>
<td>( \langle b_2, c_2 \rangle )</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

The utility difference proportion from \( \langle a_1, b_1 \rangle \) to \( \langle b_2, c_2 \rangle \) can be calculated as following:

\[
\rho (\langle a_1, b_1 \rangle, \langle b_1, c_1 \rangle) \cdot \rho (\langle b_1, c_1 \rangle, \langle b_2, c_2 \rangle) = \rho (\langle a_1, b_1 \rangle, \langle b_2, c_2 \rangle) = \frac{p_1}{1 - p_1} \cdot r \quad (8)
\]

Another utility difference proportion from \( \langle a_1, c_1 \rangle \), \( \langle b_2, c_2 \rangle \) can be calculated in the similar way:

\[
\rho (\langle a_1, c_1 \rangle, \langle b_1, c_1 \rangle) \cdot \rho (\langle b_1, c_1 \rangle, \langle b_2, c_2 \rangle) = \rho (\langle a_1, c_1 \rangle, \langle b_2, c_2 \rangle) = \frac{1}{1 - p_1} \cdot r \quad (9)
\]

Other utility difference proportions can be calculated analogously. Therefore, the inter-factorial matrix are now generated as shown in table 10:

Table 10. Inter-factorial matrix of consequences in aspect 1 and aspect 2

<table>
<thead>
<tr>
<th>ρ</th>
<th>( \langle b_2, a_2 \rangle )</th>
<th>( \langle c_2, a_2 \rangle )</th>
<th>( \langle b_2, c_2 \rangle )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \langle a_1, b_1 \rangle )</td>
<td>( \frac{p_1 r}{1 - p_1} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \langle a_1, c_1 \rangle )</td>
<td></td>
<td>( \frac{r}{1 - p_1} )</td>
<td></td>
</tr>
<tr>
<td>( \langle b_1, c_1 \rangle )</td>
<td></td>
<td></td>
<td>( r )</td>
</tr>
</tbody>
</table>

When we come to this step, we define measure for the aspect safety as \( m_1 : D_1 \to \mathbb{R} \) such that for all \( x_i \in \alpha^1 \) (based on [15]):

\[
m_1(x_i) = \rho (\langle x_i, c_1 \rangle, \langle b_2, c_2 \rangle) \quad (10)
\]

The measure in aspect fuel consumption, \( m_2 \) can be represented in the same way:

\[
m_2(x_i) = \rho (\langle x_2, a_2 \rangle, \langle b_2, c_2 \rangle) \quad (11)
\]

Therefore, the utility value of each consequence in each aspect can be shown as in table 11 and table 12:
Table 11. Values of $m_1$

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$b_1$</th>
<th>$c_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_1$</td>
<td>$r$</td>
<td>$r$</td>
<td>0</td>
</tr>
</tbody>
</table>

$$m_1 = \frac{r}{1 - p_1}$$

Table 12. Values of $m_2$

<table>
<thead>
<tr>
<th></th>
<th>$a_2$</th>
<th>$b_2$</th>
<th>$c_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_2$</td>
<td>0</td>
<td>$\frac{p_2}{p_2 - 1}$</td>
<td>$\frac{1}{p_2 - 1}$</td>
</tr>
</tbody>
</table>

Ultimately, we construct a overall measure for all things considered and , for this reason, we define $m_0$: $\alpha_1 \times \alpha_2 \rightarrow \text{Re}$, such that $(x_1, x_2) \in \alpha_1 \times \alpha_2$:

$$m_0(x_1, x_2) = m_0(x_1) + m_0(x_2). \quad (12)$$

As a result, the final utility value of the consequence $a$ and other consequences can be calculated analogously:

$$m_0(a_1, a_2) = m_0(a_1) + m_0(a_2) = \frac{r}{1 - p_1} \quad (13)$$

Now we continue using the same car-buying example mentioned before. To be more specific, we assign certain numbers in our example in order to show how the theoretical foundation works behind the program.

We assign $1/3$ to $p_1$ for aspect safety, which means that the utility difference proportion $\rho$ (<$a_1$, $b_1$>, <$a_1$, $c_1$>) is $1/3$. We also assign $4/3$ to $p_2$ for aspect fuel consumption which means that the utility difference proportion $\rho$ (<$b_2$, $a_2$>, <$c_2$, $a_2$>) is $4/3$. We also assign 2 to $r$ which is the proportion between the utility difference from $b$ to $c$ in aspect safety and the utility difference from $b$ to $c$ with respect to aspect fuel. Therefore, we can calculate and fulfill the intra-factorial and inter-factorial matrix as shown in table 13 on the basis of tables 8,9 and 10 in the previous part.
Table 13. The combined matrix for the real example

<table>
<thead>
<tr>
<th>ρ</th>
<th>&lt;a₁,b₁&gt;</th>
<th>&lt;a₁,c₁&gt;</th>
<th>&lt;b₁,c₁&gt;</th>
<th>&lt;b₂,a₂&gt;</th>
<th>&lt;c₂,a₂&gt;</th>
<th>&lt;b₂,c₂&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;a₁,b₁&gt;</td>
<td>1</td>
<td>1/3</td>
<td>1/2</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>&lt;a₁,c₁&gt;</td>
<td></td>
<td>1</td>
<td>3/2</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>&lt;b₁,c₁&gt;</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>&lt;b₂,a₂&gt;</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>4/3</td>
<td>4</td>
</tr>
<tr>
<td>&lt;c₂,a₂&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>&lt;b₂,c₂&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Hence, the values of \( m_1 \) and \( m_2 \) can be generated as in tables 14 & 15:

Table 14: Values of \( m_1 \) for the real example

<table>
<thead>
<tr>
<th></th>
<th>( a_1 )</th>
<th>( b_1 )</th>
<th>( c_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_1 )</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 15: Values of \( m_2 \) for the real example

<table>
<thead>
<tr>
<th></th>
<th>( a_2 )</th>
<th>( b_2 )</th>
<th>( c_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_2 )</td>
<td>0</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

The final step is the calculation of the utility values for these three cars all things considered.

\[
m_{0}(a_1, a_2) = m_1(a_1) + m_2(a_2) = 3 + 0 = 3
\]

\[
m_{0}(b_1, b_2) = m_1(b_1) + m_2(b_2) = 2 + 4 = 6
\]

\[
m_{0}(c_1, c_2) = m_1(c_1) + m_2(c_2) = 0 + 3 = 3
\]

As the results show above, we can see that the utility value for consequence \( b \) is the largest one; which means that Volvo is the most suitable car when we considered both safety and fuel consumption. Therefore, Tom can choose Volvo as his car.

3 Method

3.1 Platform and Programming language

First of all, the whole program is developed as a web application. Hence we use Javascript and HTML. We have also used Java. We use Myeclipse as our development platform.

Figure 5 shows a package diagram of this application which illustrates the used languages. First, Javascript is used to compile the overall representation part of the application. To be more specific, Javascript handles the input and output values into the interface. Secondly, Java is used to write the functions or methods for the application, for example for editing the input values and
for performing the calculations. HTML is a markup language, which is used to structure information and content in the application such as titles, paragraphs and so on.

![The Languages of making the application](image)

Figure 5: The package diagram of the languages of making the application

### 3.2 Design issues

First of all, the application should have ability to handle a proportion matrix for each aspect, and one proportion matrix for each combination of aspects. The sizes of these matrices should be generated at the beginning of the application. Implementing them in a proper way is essential of this decision-making application.

Another basic thing is the way of representing the utility difference proportions between consequences. An appropriate representation will make the calculation smoother and provide users more convenience.

The next issue is how the calculations of utility values and utility difference proportions can be conducted automatically. The challenge is how to implement them in the program. Users can assign three values to inter-factorial and intra-factorial matrices, and the application will calculate the rest of utility differences with related values. There are three difference proportions that need to be assigned by the users, the remaining needed proportions will be calculated automatically. In addition, the application should be capable of calculating utility values for consequences automatically.

### 3.3 Design choices

This section will present the choices we made regarding the previous issues and problems in the previous section.

We designed the application so that the utility difference proportions can be represented as fractions as well as integers. The proportion values are stored in a 7x7 array. The combined table for our car example is shown in the table 13 in the section 2.4.2. Note in this table, that the first
column and row of the table indicates the name for the cells, and it is a combination of proportion matrix for consequences in one aspect and proportion matrix for consequences between different aspects.

We have predefined the input utility difference proportions and related proportions at fixed positions in certain cells. Therefore, once these values have been input or calculated, they will be returned to the corresponding cells in the matrix. As shown in section 2.4.2, the utility values for each aspect can be fetched directly from specific cells in the matrix, except for the utility value $a_1$ in measure $1(m_1)$ which has to be calculated as shown in section 2.4.2. Finally, the overall utility values for each consequence are calculated by adding the utility values of each aspect respectively. Note that, the proportions or values can be fetched at corresponding cell in the whole matrix.

The reason for designing the application this way is that we want to simplify the interface and operation for users and help them make decisions. This application is just at the beginning of development, so it is not meant to be complicated in its structure. Hence, we have chosen an easy way to start with.

### 4 Results

In section 2.4.2, we introduced the basic theoretical background behind the application; in this section, we will present the overall result of applying these theories. Important code will be explained.

First of all, figure 6 shows a screenshot of the user interface after all the values have been input and calculated.
Consequence1: a
Consequence2: b
Consequence3: c

Proportion from a1b1 to a1c1: 1/3
Proportion from b2a2 to c2a2: 4/3
Proportion from b1c1 to b2c2: 2

<table>
<thead>
<tr>
<th>ρ</th>
<th>&lt;a1,b1&gt;</th>
<th>&lt;a1,c1&gt;</th>
<th>&lt;b1,c1&gt;</th>
<th>&lt;b2,a2&gt;</th>
<th>&lt;c2,a2&gt;</th>
<th>&lt;b2,c2&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;a1,b1&gt;</td>
<td>1</td>
<td>1/3</td>
<td>1/2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;a1,c1&gt;</td>
<td></td>
<td>1</td>
<td>3/2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;b1,c1&gt;</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;b2,a2&gt;</td>
<td></td>
<td></td>
<td>1</td>
<td>4/3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>&lt;c2,a2&gt;</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>&lt;b2,c2&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>a1</th>
<th>b1</th>
<th>c1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility values of consequences: aspect 1</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>a2</th>
<th>b2</th>
<th>c2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility values of consequences: aspect 2</td>
<td>0</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility values of consequences: aspects 1&amp;2</td>
<td>3</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

Figure 6: A screen shortcut of the results
In order to make the running process clearer, we illustrate an activity diagram in figure 7.

![Activity diagram for the application](Image)

*Figure 7: Activity diagram for the application*

When the user starts to run the application, the first thing is to define the names of the user's consequences and press submit button, then the names will be updated in the inter-factorial and intra-factional matrices. Secondly, the user needs to define the proportion from \(\langle a_1, b_1 \rangle\) to \(\langle a_1, c_1 \rangle\), which is \(p_1\), and press calculate button, and after this the inter-factorial matrix for aspect 1 will be fulfilled. Similarly, the users enter the proportion from \(\langle b_2, a_2 \rangle\) to \(\langle c_2, a_2 \rangle\), which is \(p_2\), and press calculate button, and then the inter-factorial matrix for aspect 2 will be fulfilled. The last step is to define the value of \(r\), in the screenshot, it is “difference proportion: \(\langle b_1, c_1 \rangle\) to \(\langle b_2, c_2 \rangle\)” . Once the value of \(r\) is defined and the calculate button has been pressed, the matrices of utility values of consequences in aspect 1 and aspect 2 and the utility values of consequences in both aspects will be generated. In section 2.4.2, we already introduced the theoretical background for how these utility values can be calculated. Therefore, according to the values in the matrix of utility values for all things considered, decision makers can choose the most suitable consequence.
The whole application has been edited mainly in three files to perform the program: “Test.java”, “index.jsp” and “Fraction.java”. We also contained these three files in appendix 1, 2 and 3.

The groundwork of file “index” is to store values and to present the application interface. It stores input values, all difference proportions and utility values in specific positions through the program. For instance: “tb.rows[1].cells[3].innerHTML” (line 212, appendix 2) indicates that the value, that has been calculated will be sent to row 1 cell 3 in the HTML table. The “Fraction” file contains functions for dealing with fraction calculations. The “Test” file is doing the calculations for the utility difference proportions and utility values.

Note that we have considered two situations for the input difference proportions; we assume that the input values can be a fraction or an integer.

Now, let us demonstrate how the code works.

In appendix 2, code through line 30 to 134 shows how the default table is generated. The values will be stored in a 7x7 array. Lines from 135 to 165 show how the default table gets the names for each cell. It means that when the users input the names of their consequences, each name is displayed onto the table through the predefined position. For example, if the users type name for consequence1, the name will be placed in to row 0, cell 1 in the table. The other tables work in the same way.

Lines 174 to 185 in the appendix 2 show how the first inter-fatorial proportion matrix is fulfilled. Firstly, the difference proportion between two consequences that are the same will be equal to 1. For example, the difference proportion from \( \langle a_1, b_1 \rangle \) to \( \langle a_1, b_1 \rangle \) will equal to 1. Secondly, the difference proportion from \( \langle a_1, b_1 \rangle \) to \( \langle a_1, c_1 \rangle \) is assigned by the users and will be placed in the table through line 178 into the specified position. The last step is to calculate the difference proportions from \( \langle a_1, b_1 \rangle \) to \( \langle b_1, c_1 \rangle \) and from \( \langle a_1, c_1 \rangle \) to \( \langle b_1, c_1 \rangle \) by using method P1 (see line 180 in appendix 2) in class Test (Test.java). Code for P1 is shown from line 1 to 48 in appendix 1. The calculations are based on the difference proportion from \( \langle a_1, b_1 \rangle \) to \( \langle a_1, c_1 \rangle \) (see table 8). Finally, the calculated difference proportions are displayed in the interface (line 180 in appendix 2). The calculated values are returned as a message to a callback function. The code for the callback function is shown in lines between 210 and 214 in appendix 2. To be more specific, the calculated difference proportions will be returned to row 1, cell 3 and row 2, cell 3 in the table respectively. The second inter-fatorial proportion matrix is fulfilled in a similar way as the first inter-fatorial proportion matrix. The calculations for that are shown in the appendix 1, lines between 49 and 90, and they are based on the difference proportion from \( \langle b_2, a_2 \rangle \) to \( \langle c_2, a_2 \rangle \) (see table 9). We have made some comments in the code in order to make it easier to understand.

All difference proportions have been calculated and stored in the table and we now explain the calculations of the utility values.

We start from the utility value of \( a_1 \) (consequence a in aspect 1) where the calculation is different from the other utility values according to section 2.4.2. In appendix 1, lines 119 to 148, the code shows how \( a_1 \) is calculated.

After the calculation, \( a_1 \) will be returned to the predefined position: “tb2.rows[1].cells[1].innerHTML”, which means returning the values to the first row and first column in the table 2 (here table 2 is the table containing utility values of consequences in aspect 1).

The other utility values can be fetched directly from the intra-fatorial proportion matrices in the way as the code in appendix 2, lines 244 to 252, shows. For example, the value in row 1, cell 1 in table 4 is fetched from the value in row 1, cell 1 in table 2.

The overall utility values of consequences in both aspects are calculated by doing an addition: the utility value of each consequence in aspect 1 plus the utility value of each consequence in aspect 2. It is shown in appendix 2, lines 253 to 256. Here we explain line 256 in more detail. The utility value is generated by adding value “r” and value in row 1, cell 2 in table 2. Then the utility value will be returned through the function callbackAdd to row 1, cell 2 in table 4, the code is shown in line 256 and line 242 in appendix 2.
Ultimately, the application will generate three new tables: the table of utility values for aspect 1, the table of utility values for aspect 2 and the table of utility values for all things considered, which means that the three tables will display the utility values of the consequences in each aspect and the overall utility value for each consequence in both aspects.

The results can be used as an aid for decision making if the decision maker has the same preference structure as the one that has been used in our example.

5 Discussion

This thesis presents a pilot test about implementation of utility calculation based on utility difference proportions. It allows decision makers to compare different consequences in different aspects pairwise.

People usually make decisions based on their intuition. When the decision problems become complex, it is better to structure the problem in a proper way. Our application can provide a simple way by implementing and representing the difference proportion matrices and utility calculations. It first requires users to assign utility difference proportions between two consequences at time, then the application will calculate the related proportions automatically; at last, based on the existing difference proportions, the final result will generate the utility values for the different consequences respectively.

The application will display the result with all things considered, corresponding utility difference proportions. Every difference proportion and utility value in the table has a predefined position and user can get help for his/her decision making by just looking at final utility values.

However, there are no perfect studies and the space for improvement and development always exist. This application deals with decision problems which contains three consequences and two aspects in a specific case. It can only do one pairwise comparison at time, thus to compare more than two consequences simultaneously can lead a new direction to be developed. In addition, the ordering of the utility difference proportions have been predefined, so it lacks of flexibility. Furthermore, if we would not have predefined the preference orderings, another calculation should be done: consistency checking of difference proportions. For instance: the application should check whether the values satisfy the ordering that the users assume in the beginning.

6 Conclusion and Future work

This report describes a pilot study for a new decision making support tool, including its theoretical foundation, design idea, working process and motivation. The pilot application has been implemented successfully and the original goal was accomplished.

There is a variety of decision-making support tools which have been widely used, such as Expert Choice: very popular software that focuses on helping people to make choices. Comparing the existing decision-making support tools with the software that we want to develop in the future, they share some common features in some aspects, for example, they are both concerned with multi-criteria decision analysis (MCDA). Implementing pairwise comparisons is also one similarity. What makes them different from each other is: the software we want to develop establishes utility difference proportions for consequences.

Our application can be developed further in the future.

First, this application works with decision problems that only contain two aspects and three consequences. For this reason, we suggest that further improvement work can focus on adding more consequences and aspects, so as to broaden its applicability and improve the efficiency. On the other hand, this “three consequences and two aspects” is a specific case as a test for the application. The application should also handle more general cases even if there are many aspects and consequences. Besides, the specific preference ordering of the consequences is also a delimitation. We believe that the application can be improved so that the preference ordering is
not predefined which makes the application more flexible. In our application, modifications are not allowed during the running process, which means lack of interactivity. As a result, another improvement that can be done in the future is to make the application more interactive and more user-friendly so that the decision makers can update new aspects or consequences during the runtime. The application could then handle more complicated problems with more flexibility and convenience. In other words, it would be more like a decision-making support tool which is also the ultimate target with this project. These meliorations would make the application a real decision-making tool that is capable of dealing with multiple aspects and consequences, where also the users can update new requirements during the process in order to make their decision problems complete and be able to more likely achieve the best result.

Acknowledgements

We would like to thank our supervisor Magnus. Without the help and support of him, we could have not accomplished this work successfully. We also would like to thank our examiner Ulla, who helped us to make an improvement for this thesis.

References


Appendix 1: Test.java

```java
package se.hig.decision;

import se.hig.decision.Fraction;

public class Test {

    public String P1(String p1) {
        String a;
        if (p1.contains("/")) {
            int x1 = Integer.parseInt(p1.split("/")[0]);
            int y1 = Integer.parseInt(p1.split("/")[1]);
            Fraction fraction1 = new Fraction(1, 1);
            Fraction fraction2 = new Fraction(x1, y1);
            Fraction fraction3 = fraction1.minus(fraction2);
            int x2 = Integer.parseInt(fraction3.toString().split("/")[0]);
            int y2 = Integer.parseInt(fraction3.toString().split("/")[1]);
            fraction1 = new Fraction(1, 1);
            fraction2 = new Fraction(x2, y2);
            fraction3 = fraction1.divide(fraction2);
            a = fraction3.toString();
            if (a.contains("/1")) {
                a = a.substring(0, a.length() - 2);
            }
            fraction2 = new Fraction(x1, y1);
            Fraction fraction4 = fraction2.multiply(fraction3);
            String b = fraction4.toString(); //p1/(1-p1)
            if (b.contains("/1")) {
                b = b.substring(0, b.length() - 2);
            }
            a = a + "," + b;
        } else {
            int n = Integer.parseInt(p1);
            Fraction fraction1 = new Fraction(1, 1 - n); //1/(1-p1)
            a = fraction1.toString();
            if (a.contains("/1")) {
                a = a.substring(0, a.length() - 2);
            }
            Fraction fraction2 = new Fraction(n, 1 - n); //p1/(1-p1)
            String b = fraction2.toString();
            if (b.contains("/1")) {
                b = b.substring(0, b.length() - 2);
            }
            a = a + "," + b;
        }
        return a;
    }

    public String P2(String p2) {
        String a;
    }
```
if (p2.contains("/")) {
    int x1 = Integer.parseInt(p2.split("/")[0]);
    int y1 = Integer.parseInt(p2.split("/")[1]);
    Fraction fraction1 = new Fraction(1, 1);
    Fraction fraction2 = new Fraction(x1, y1);
    Fraction fraction3 = fraction2.minus(fraction1); // p2 - 1
    int x2 = Integer.parseInt(fraction3.toString().split("/")[0]);
    int y2 = Integer.parseInt(fraction3.toString().split("/")[1]);
    fraction1 = new Fraction(1, 1);
    fraction2 = new Fraction(x2, y2);
    fraction3 = fraction1.divide(fraction2); // 1/(p2 - 1)
    a = fraction3.toString();
    if (a.contains("/1")) {
        a = a.substring(0, a.length() - 2);
    }
    fraction2 = new Fraction(x1, y1);
    Fraction fraction4 = fraction2.multiply(fraction3);
    String b = fraction4.toString(); // p2/(p2 - 1)
    if (b.contains("/1")) {
        b = b.substring(0, b.length() - 2);
    }
    a = a + "," + b;
} else {
    int n = Integer.parseInt(p2);
    Fraction fraction1 = new Fraction(1, n - 1); // 1/(n-1)
    a = fraction1.toString();
    if (a.contains("/1")) {
        a = a.substring(0, a.length() - 2);
    }
    Fraction fraction2 = new Fraction(n, n - 1); // n/(n-1)
    String b = fraction2.toString();
    if (b.contains("/1")) {
        b = b.substring(0, b.length() - 2);
    }
    a = a + "," + b;
}
return a;
}

public String R1(String R, String P) {
    Fraction fraction1 = new Fraction(1, 1);
    Fraction fraction2 = new Fraction(1, 1);
    if (R.contains("/")) {
        int x1 = Integer.parseInt(R.split("/")[0]);
        int y1 = Integer.parseInt(R.split("/")[1]);
        fraction1 = new Fraction(x1, y1);
    } else {
        fraction1 = new Fraction(Integer.parseInt(R), 1);
    }
    if (P.contains("/")) {
int x1 = Integer.parseInt(P.split("/")[0]);
int y1 = Integer.parseInt(P.split("/")[1]);
fraction2 = new Fraction(x1, y1);

} else {
    fraction2 = new Fraction(Integer.parseInt(P), 1);
}

Fraction fracation3 = fraction1.add(fraction2);
String a = fraction3.toString();
if (a.contains("/1")) {
    a = a.substring(0, a.length() - 2);
}
return a;
}

public String M1(String R, String P) {
    Fraction fraction1 = new Fraction(1, 1);
    Fraction fraction2 = new Fraction(1, 1);
    if (R.contains("/")) {
        int x1 = Integer.parseInt(R.split("/")[0]);
        int y1 = Integer.parseInt(R.split("/")[1]);
        fraction1 = new Fraction(x1, y1);
    }
    else {
        fraction1 = new Fraction(Integer.parseInt(R), 1);
    }
    if (P.contains("/")) {
        int x1 = Integer.parseInt(P.split("/")[0]);
        int y1 = Integer.parseInt(P.split("/")[1]);
        fraction2 = new Fraction(x1, y1);
    }
    else {
        fraction2 = new Fraction(Integer.parseInt(P), 1);
    }
    Fraction fraction3 = fraction1.multiply(fraction2); //r/(1-p1)
    String a = fraction3.toString();
    if (a.contains("/1")) {
        a = a.substring(0, a.length() - 2);
    }
    if(a.contains("0/"))
        a="0";
    return a;
}
Appendix 2: index.jsp

```html
1. < %@ page language="java" pageEncoding="utf-8" %>
2. <%
3. String path = request.getContextPath();
4. String basePath =
   request.getScheme() + "://" + request.getServerName() + "." + request.getServerPort() + path + "/";
5. %>
6. <!
7. --
8. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
9. <html>
10. <head>
11.   <base href="<%=basePath%>">
12.   <title>My JSP 'index.jsp' starting page</title>
13.   <meta http-equiv="pragma" content="no-cache">
14.   <meta http-equiv="cache-control" content="no-cache">
15.   <meta http-equiv="expires" content="0">
16.   <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
17.   <meta http-equiv="description" content="This is my page">
18.   <script type='text/javascript' src='/demo/dwr/interface/Test.js'></script>
19.   <script type='text/javascript' src='/demo/dwr/engine.js'></script>
20.   <script type='text/javascript' src='/demo/dwr/util.js'></script>
21.   <link rel="stylesheet" type="text/css" href="styles.css">
22. -->
23.   <script language="javascript">
24.     function add(){
25.       var tb = document.getElementById("testTbl");
26.       var rowNum=tb.rows.length;
27.       for (i=0;i<rowNum;i++)
28.         {
29.           tb.deleteRow(i);
30.           rowNum=rowNum-1;
31.           i=i-1;
32.       }
33.     //add new array
34.     var newTr=new Array();
35.     var name=new Array("a1b1","a1c1","b1c1","a2b2","a2c2","b2c2")
36.     var num=7;
37.     for(var i=0;i<num;i++)
38.       {
39.         if(i==0)
40.           {
41.             newTr[0]=tb.insertRow();
42.             newTr[0].height=50;
43.             var newTd=new Array();
44.             newTd[0]= newTr[0].insertCell();
45.             newTd[0].innerText="ρ";
```
newTd[0].setAttribute("align","center");
newTd[0].width=50;
for(var j=1;j<num;j++)
{
    newTd[j]= newTr[0].insertCell();
    newTd[j].innerText=name[j-1];
    newTd[j].width=50;
    newTd[j].setAttribute("align","center");
}

else
{
    newTr[i]=tb.insertRow();
    newTr[i].height=50;
    var newTd=new Array();
    newTd[0]= newTr[i].insertCell();
    newTd[0].innerText=" ";
    newTd[0].setAttribute("align","center");
    for(var j=1;j<num;j++)
    {
        newTd[j]= newTr[i].insertCell();
        newTd[j].innerText=" ";
        newTd[j].setAttribute("align","center");
    }
}
}

function add1(tablename,name,trname)
{
    var tb = document.getElementById(tablename);
    var rowNum=tb.rows.length;
    for (i=0;i<rowNum;i++)
    {
        tb.deleteRow(i);
        rowNum=rowNum-1;
        i=i-1;
    }
    //add new array
    var newTr=new Array();
    for(var i=0;i<2;i++)
    {
        if(i==0)
        {
            newTr[0]=tb.insertRow();
            newTr[0].height=50;
            var newTd=new Array();
            newTd[0]= newTr[0].insertCell();
            newTd[0].innerText=" ";
            newTd[0].setAttribute("align","center");
            newTd[0].width=50;
            for(var j=1;j<4;j++)
            {

            }
        }
    }
}
newTd[j].innerText=name[j-1];
newTd[j].width=50;
newTd[j].setAttribute("align","center");
}
else
{
newTr[i]=tb.insertRow();
newTr[i].height=50;
var newTd = new Array();
newTd[0]= newTr[i].insertCell();
newTd[0].innerText=trname;
newTd[0].setAttribute("align","center");
for(var j=1;j<4;j++)
{
newTd[j]= newTr[i].insertCell();
newTd[j].innerText=" ";
newTd[j].setAttribute("align","center");
}
}

function commit(object)
{
if(document.getElementById("textName1").value=="")
{
alert("Please enter the name of consequence 1");
return;
}
if(document.getElementById("textName2").value=="")
{
alert("Please enter the name of consequence 2");
return;
}
if(document.getElementById("textName3").value=="")
{
alert("Please enter the name of consequence 3");
return;
}
tb = document.getElementById("testTbl");
tb.rows[0].cells[1].innerHTML=document.getElementById("textName1").value+"1"+document.getElementById("textName2").value+"1";
tb.rows[0].cells[2].innerHTML=document.getElementById("textName1").value+"1"+document.getElementById("textName3").value+"1";
tb.rows[0].cells[3].innerHTML=document.getElementById("textName2").value+"1"+document.getElementById("textName3").value+"1";
function commitP1()
{
    if(document.getElementById("textP1").value=="")
    {
        alert("please enter the value of proportion from a1b1 to a1c1");
        return;
    }

    var tb = document.getElementById("testTbl");
    tb.rows[1].cells[1].innerHTML="1";
    tb.rows[2].cells[2].innerHTML="1";
    tb.rows[3].cells[3].innerHTML="1";
    tb.rows[1].cells[2].innerHTML=document.getElementById("textP1").value;
    var a=document.getElementById("textP1").value;
    Test.P1(a,callBackAddp1);
}

function commitP2()
{
    if(document.getElementById("textP2").value=="")
    {
        alert("please enter the value of proportion from b2a2 to c2a2");
        return;
    }
}
else
{
    var tb = document.getElementById("testTbl");
    tb.rows[4].cells[4].innerHTML="1";
    tb.rows[5].cells[5].innerHTML="1";
    tb.rows[6].cells[6].innerHTML="1";
    tb.rows[4].cells[5].innerHTML=document.getElementById("textP2").value;
    var a=document.getElementById("textP2").value;
    Test.P2(a,callBackAddp2);
}

function callBackAddp2(message){
    var tb = document.getElementById("testTbl");
    tb.rows[4].cells[6].innerHTML=message.split(",")[1];
    tb.rows[5].cells[6].innerHTML=message.split(",")[0];
}

function callBackAddp1(message){
    var tb = document.getElementById("testTbl");
    tb.rows[1].cells[3].innerHTML=message.split(",")[1];
    tb.rows[2].cells[3].innerHTML=message.split(",")[0];
}

function commitR()
{
    if(document.getElementById("textR").value=="")
    {
        alert("please enter the value of proportion from b1c1 to b2c2");
        return;
    }

    var name=new Array(document.getElementById("textName1").value+"1",document.getElementById("textName2").value+"1",document.getElementById("textName3").value+"1");
    add1("testTb2",name,"Utility values of consequences:aspect 1");

    var name=new Array(document.getElementById("textName1").value+"2",document.getElementById("textName2").value+"2",document.getElementById("textName3").value+"2");
    add1("testTb3",name,"Utility values of consequences:aspect 2");

    var name=new Array(document.getElementById("textName1").value,document.getElementById("textName2").value,document.getElementById("textName3").value);
    add1("testTb4",name,"Utility values of consequences:aspect 1 & 2");

    var tb1 = document.getElementById("testTbl");
    var tb2 = document.getElementById("testTb2");
    tb2.rows[1].cells[2].innerHTML=document.getElementById("textR").value;
    tb2.rows[1].cells[3].innerHTML="0";

    Test.M1(document.getElementById("textR").value,tb1.rows[2].cells[3].innerHTML,callBackAddm1);
}

function callBackAdd(message)
```javascript
{ var tb4 = document.getElementById("testTb4");
  tb4.rows[1].cells[2].innerHTML = message;
}

function callBackAddm1(message) {
  var tb1 = document.getElementById("testTbl");
  var tb2 = document.getElementById("testTb2");
  tb2.rows[1].cells[1].innerHTML = message;
  var tb3 = document.getElementById("testTb3");
  tb3.rows[1].cells[1].innerHTML = "0";
  var tb4 = document.getElementById("testTb4");

  Test.R1(document.getElementById("textR").value, tb3.rows[1].cells[2].innerHTML, callBackAddm1);
}

</script>

</head>

<body onload="add();">

<table>
  <tr>
    <td>
      Consequence 1:
    </td>
    <td>
    </td>
  </tr>
  <tr>
    <td>
      <input type="text" id="textName1" style="width: 100%" value="a"/>
    </td>
    <td>
      <input type="button" id="ButtonCommit" onclick="commit(1);" value="Submit" />
    </td>
  </tr>
  <tr>
    <td>
      Consequence 2:
    </td>
    <td>
    </td>
  </tr>
  <tr>
    <td>
      <input type="text" id="textName2" style="width: 100%" value="b"/>
    </td>
    <td>
      <input type="button" id="ButtonCommit" onclick="commit(2);" value="Submit" />
    </td>
  </tr>
</table>
```
<table>
<thead>
<tr>
<th>Consequence 3:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion from $a_1b_1$ to $a_1c_1$:</td>
<td></td>
</tr>
<tr>
<td>Proportion from $b_2a_2$ to $c_2a_2$:</td>
<td></td>
</tr>
<tr>
<td>Proportion from $b_1c_1$ to $b_2c_2$:</td>
<td></td>
</tr>
</tbody>
</table>

```html
<input type="text" id="textName3" style="width: 100%" value="c" />
<input type="button" id="ButtonCommit" onclick="commit(3);" value="Submit" />
<input id="textP1" style="width: 100%" type="text" />
<input type="button" id="ButtonCommit" onclick="commitP1();" value="Calculate" />
<input id="textP2" style="width: 100%" type="text" />
<input type="button" id="ButtonCommit" onclick="commitP2();" value="Calculate" />
<input id="textR" style="width: 100%" type="text" />
<input type="button" id="ButtonCommit" onclick="commitR();" value="Calculate" />
```
<table id="testTbl" border=1 >
  &lt;/table&gt;

<table id="testTb2" border=1 >
  &lt;/table&gt;

<table id="testTb3" border=1 >
  &lt;/table&gt;

<table id="testTb4" border=1 >
  &lt;/table&gt;

&lt;/body&gt;
&lt;/html&gt;
Appendix 3: fraction.java

1. package se.hig.decision;
2.
3. public class Fraction implements Comparable {
4.     private int numerator;
5.     private int denominator;
6. 
7.     public Fraction() {
8.         
9.     }
10. 
11.     public Fraction(int n, int d) {
12.         setFraction(n, d);
13.     }
14.     public Fraction(double d) {
15.         convertToFraction(d);
16.     }
17. 
18.     private void convertToFraction(double d) {
19.         int decimalCount = 1;
20.         int dLen = String.valueOf(d).length();
21.         for (int i = 0; i < dLen; i++) {
22.             d = d * 10;
23.             decimalCount *= 10;
24.         }
25.         numerator = (int) d;
26.         denominator = decimalCount;
27.         reduct();
28.     }
29. 
30.     public void setNumerator(int n) {
31.         numerator = n;
32.         reduct();
33.     }
34. 
35.     public int getNumerator() {
36.         return numerator;
37.     }
38. 
39.     public void setDenominator(int d) {
40.         if (d == 0)
41.             denominator = 1;
42.         else
43.             denominator = d;
44.         reduct();
45.     }
public int getDenominator() {
    return denominator;
}

public void setFraction(int n, int d) {
    setNumerator(n);
    setDenominator(d);
    reduct();
}

public String toString() {
    if (numerator * denominator < 0)
        return "-" + Math.abs(numerator) + "/" + Math.abs(denominator);
    else
        return Math.abs(numerator) + "/" + Math.abs(denominator);
}

public boolean equals(Fraction f) {
    if (numerator == f.numerator && denominator == f.denominator)
        return true;
    else
        return false;
}

public boolean isGreater(Fraction f) {
    if ((double) numerator / denominator > (double) f.numerator / f.denominator)
        return true;
    else
        return false;
}

private void reduct() {
    int minValue = Math.min(Math.abs(numerator), Math.abs(denominator));
    for (int i = minValue; i >= 1; i--)
    {
        if (numerator % i == 0 && denominator % i == 0)
        {
            numerator = numerator / i;
            denominator = denominator / i;
            break;
        }
    }
}

public Fraction add(Fraction f) {
    Fraction fraction = new Fraction();
    fraction.numerator = numerator * f.denominator + f.numerator;
    fraction.denominator = denominator * f.denominator;
    fraction.reduct();
    return fraction;
}

public Fraction minus(Fraction f) {

}
Fraction fraction = new Fraction();
fraction.numerator = numerator * f.denominator - f.numerator * denominator;
fraction.denominator = denominator * f.denominator;
fraction.reduct();
return fraction;
}

public Fraction multiply(Fraction f) {
    Fraction fraction = new Fraction();
    fraction.numerator = numerator * f.numerator;
    fraction.denominator = denominator * f.denominator;
    fraction.reduct();
    return fraction;
}

public Fraction divide(Fraction f) {
    Fraction fraction = new Fraction();
    fraction.numerator = numerator * f.denominator;
    fraction.denominator = denominator * f.numerator;
    fraction.reduct();
    return fraction;
}

public int compareTo(Object o) {
    Fraction f = (Fraction) o;
    if ((double) numerator / denominator > (double) f.numerator / f.denominator)
        return 1;
    else if ((double) numerator / denominator < (double) f.numerator / f.denominator)
        return -1;
    else
        return 0;
}

public boolean equals(Object obj) {
    Fraction f = (Fraction) obj;
    if (Math.abs((double) numerator / denominator - (double) f.numerator / f.denominator) < 0.00001)
        return true;
    return false;
}

public int hashCode() {
    String str = String.valueOf((double) numerator / denominator);
    return str.hashCode();
}