

# Which Soft? Decision Support Software

Anabela Tereso, Ricardo Macedo, Rafael Abreu, João Brandão, Henrique Martins

University of Minho, School of Engineering, 4710-057 Braga, Portugal

[anabelat@dps.uminho.pt](mailto:anabelat@dps.uminho.pt); [a54730@alunos.uminho.pt](mailto:a54730@alunos.uminho.pt); [a54776@alunos.uminho.pt](mailto:a54776@alunos.uminho.pt); [a54719@alunos.uminho.pt](mailto:a54719@alunos.uminho.pt); [a54760@alunos.uminho.pt](mailto:a54760@alunos.uminho.pt)

Conference Topic - CT6 - Computer Science

## Abstract

In this project we developed a decision support software that helps to choose the best decision software, but can also be applied to any other problem of selection. It is based on multicriteria methods. With this software we try to give each user the best solution, depending on his preferences.

The entire project was planned and outlined in UML, implemented in C# and the database was built with SQL Server. It was a project divided into three main stages: requirements gathering, UML specification and implementation phase.

**Key Words:** C#, SQL Server, Relational Database, Multicriteria Methods, UML, Software Engineering.

## 1. Introduction

Within the Curricular Unit Labs IV, of the third year of the Computer Engineering Degree, a project was presented to us by Professor Anabela Tereso.

This project was about creating an application for a variety of users in order to allow them to choose the best software, for a particular case, using multicriteria techniques.

In the first phase we dealt with the important task of understanding the problem and requirements analysis. This phase is complex but very important and can determine the success of a project. There are several ways to get requirements. We used interviews and literature review. We registered all aspects that we have drawn from interviews with Professor Anabela Tereso and documents available.

The second phase presents the modeling and the UML specification, supporting the development of software and features relevant to the design (e.g. diagrams) and the implementation phase (e.g. visual development). To model this project we relied on the software tool called Visual Paradigm 8.0. So all the images presented in this report were prepared for the modeling with the help of this tool.

In the third and last phase we implemented the project with Microsoft Visual Studio, the language used was C#. We created the database and inserted data of possible interest to the user when using SQL Server.

## 2. Multicriteria methods

In this section we present the multicriteria methods used in the program.

### 2.1. SMART

SMART is a technique for weighting attributes (von Winterfeldt and Edwards, 1986)(1) (Brugha, 1998)(2).

It consists of assigning 10 points to the least significant attribute. Then assign values greater than 10 to other attributes.

The comparison of the software is based on the values associated to the features. Suppose the selected characteristics are the “cost of the application” and the “interaction with user”. If the “interaction with user” is the characteristic that we consider less important, we give 10 points to it and more than 10 points to the “cost of the application”, for example 30 points, meaning that the second feature is three times more important than the least important one.

## 2.2. AHP

In AHP (Saaty, 1980)(3), the assignment of weights is done using a method of peer-to-peer comparison of the attributes, with a 1-9 scale (table 1) proposed by Saaty (1980)(3) (note: intermediate values can also be use). Table 2 presents an example of the application of this method.

Table 1- The Saaty Rating Scale (Saaty, 1980)(3)

If x is ... than y	So the number of preference to be given is:
equally important	1
a little more important	3
much more important	5
far more important	7
absolutely more important	9

Table 2-Table example of the AHP

Pair-wise comparison matrix				
BEST SOFTWARE	Interaction with user	Cost		
Interaction with user	1	1/4		
Cost	4	1		
Sum	5	1.25		Attribute weights
Normalized matrix				
BEST SOFTWARE	Interaction with user	Cost	Sum	Mean
Interaction with user	0.2	0.2	0.4	0.2
Cost	0.8	0.8	1.6	0.8
Sum				

## 2.3. ValueFn

ValueFn is a method base on Utility Theory (Fishburn, 1970)(4). It is based on the function that maps directly the values of the alternatives to their evaluation.

In the case of maximization, the following formula is used to calculate the priorities:

$$y = \frac{x - Min}{Max - Min}$$

For minimization the following formula is used:

$$y = \frac{Max - x}{Max - Min}$$

For example, the criterion "Interaction with user" should be maximized. Suppose we have three software A, B and C, with values for "Interaction with user" equal to [1,3, 5]. The priorities of each software for this criterion would be [0, 0.5, 1].

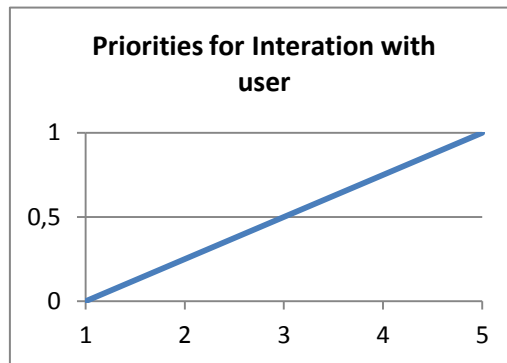


Figure 1 - Graphic example for the method ValueFn (maximization)

And the criterion "Cost" should be minimized. For the three software A, B and C, with values of "Cost" equal to [100, 800, 1000], the values of the priorities each software considering this criterion would be [1, 0.222, 0].

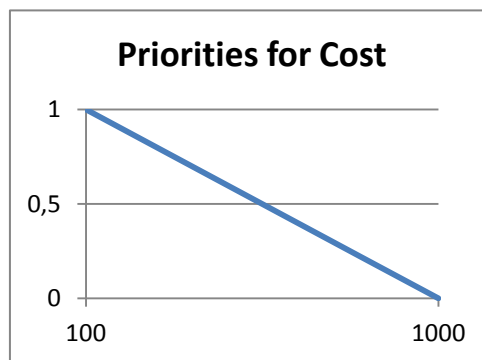


Figure 2 - Graphic example for the method ValueFn (minimization)

Then it is necessary to normalize the values (so that the sum is equal to 1).

Table 3 - Table example for the method ValueFn (maximization)

Software	Interaction with user		Priorities	Normalized Priorities
A	1		0	0.000
B	3		0.5	0.333
C	5		1	0.667
Min	1	Sum	1.5	1
Max	5			

Table 4 - Table example for the method ValueFn (minimization)

Software	Cost		Priorities	Normalized Priorities
A	100		1	0.818
B	800		0.22	0.182
C	1000		0	0.000
Min	100	Sum	1.22	1
Max	1000			

### 3. The application development

#### 3.1. Main algorithm

The main algorithm is the one that makes the comparison of software to choose the best one. The first step is defining the weights of each criterion. It can be done using one of the following methods, AHP or SMART.

For the second step it is necessary to define the priorities or weights, for each software, within each criterion. For each criterion it is necessary to define the method to use to set the priorities, which may be ValueFn or AHP.

#### 3.2. Structure of implementation

We developed the software using software engineering techniques. Initially we tried to understand the problem and did requirements gathering.

Then we proceeded to the UML modeling to create a domain model, sequence diagrams, use case and class diagrams.

Finally we arrived at the implementation phase and developed packages containing three distinct classes: User Interface Layer, Business Logic Layer and Data Access Layer.

In each phase we used the so-called Waterfall Model (figure 3) (Boehm, 1988)(5),(6).

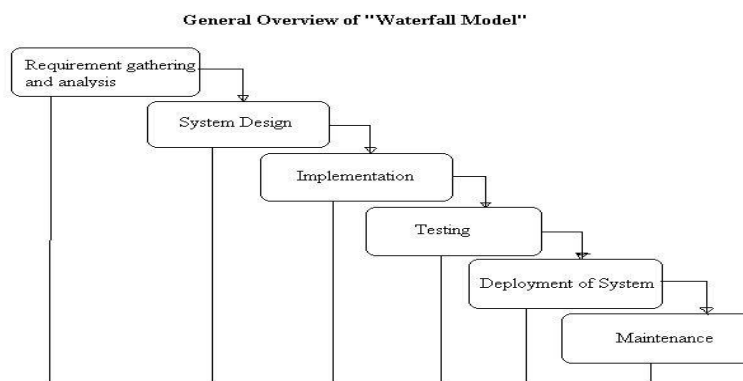


Figure 3 - Waterfall model (6)

The start of a phase should be after the completion of the previous phase. But sometimes it is impossible to solve all the problems of a phase and we have to return to previous stages in order to complete or enhance them.

#### 3.3. Programming tools

Our application was developed exclusively using Microsoft software.

For programming we used Visual C#, for database development SQL Server and Microsoft Office Word and Excel for producing other documents.

### 4. Interface

The interface is very simple. The user can understand easily how the software works.

It is provided a user manual that can be consulted in the case any doubts arise.

In the comparison of software, instructions are given to the user, in the right side of the window, explaining how to proceed, so the user feels to be making conscious decisions.

## 4.1. File management

In this program we work with lists of software that can be saved in a file. We call each list saved in a file a "Save". We can create a new file, open or save it with the same name or a different one, under the file menu or using the buttons shown in the figure bellow.

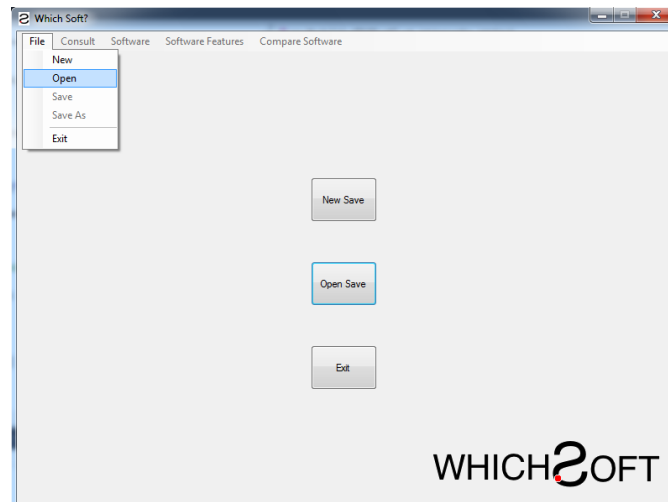


Figure 4 - File Menu

To create a new file just click on "New Save" on the screen of the software, or go to "File" menu and select "New".

To open a file just click on "Open Save" on the screen of the software, or go to "File" menu and select "Open". Then just pick one software list from the ones available. After we changed something, to save the changes, the user should go to the "File" menu and select "Save".

If you want to save but don't want to overwrite on the previous file, just go to "File" menu and select "Save As". Then choose a new name for the file.

## 4.2. Software's management

To each software list saved we can add a new software or even import a previously created one. Just click in the "Software" then "Edit Software List" to find this option.

The user will be asked to fill all the values of the existing features. You can also click the "Import From Other Software Save" to access software to be imported.

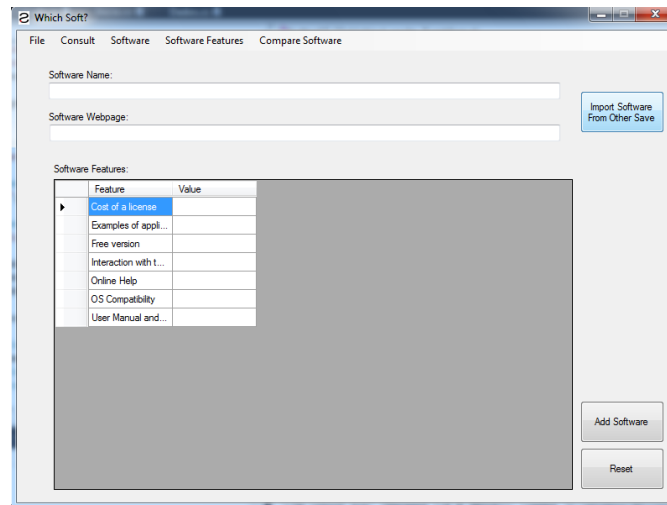


Figure 5 - Add software

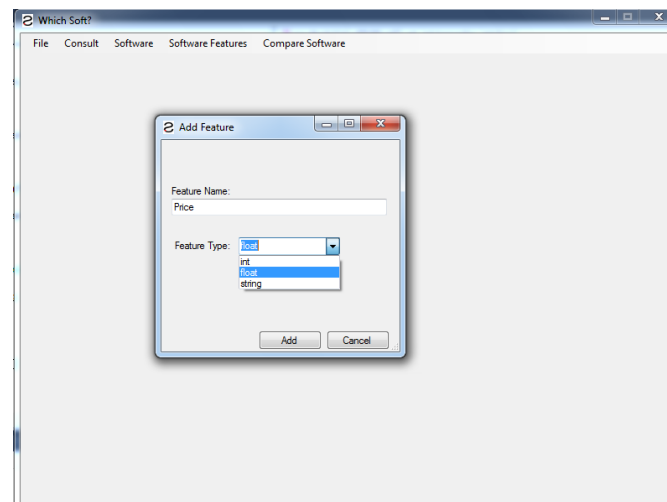


Figure 6 - Add feature

We can add new features or remove them, just by clicking the menu "Software Features" and choosing from the list the ones to be removed.

To add a new feature, we need to indicate not only the name but also the type of the feature.

The type "int" is an integer, "float" is a real number, "string" is a word or any set of characters.

We can also select a basic or extended view, in the menu "Consult". In the basic view it is displayed only the name and website of each software. In the extended view, all the characteristics of each software are shown.

Name	Site	Cost of a license	Examples of app...	Free version	Interaction with L...	Online Help
ELECTRE III-IV	<a href="http://www.lamsade.dauphine.fr/english/software.html#el34">http://www.lamsade.dauphine.fr/english/software.html#el34</a>	519	Yes	Yes	Reasonable	No
HIVIEW	<a href="http://www.catalyse.co.uk/products">http://www.catalyse.co.uk/products</a>	1100	Yes	Yes	Good	Yes
PREFCALC	<a href="http://garage.maemo.org/projects/prefcalc/">http://garage.maemo.org/projects/prefcalc/</a>	0	No	Yes	Reasonable	No
TOPSIS	<a href="http://www.stat-design.com/topsis-edl.php">http://www.stat-design.com/topsis-edl.php</a>	519	No	Yes	Good	No
WINPRE	<a href="http://www.hpre.hut.fi/">http://www.hpre.hut.fi/</a>	2134	Yes	Yes	Very bom	Yes
ELECTRE TRI	<a href="http://www.lamsade.dauphine.fr/english/software.html#TRI">http://www.lamsade.dauphine.fr/english/software.html#TRI</a>	519	Yes	Yes	Very Good	No
LPA VISIRULE	<a href="http://www.lpa.co.uk/">http://www.lpa.co.uk/</a>	0	No	Yes	Good	No

Figure 7 - Extended view

You can also view the software webpage, provided you have an internet connection available. Just go to the "Software", choose "View Software Webpage" and click on the software and then you will see the corresponding webpage.

### 4.3. Compare Software

#### Step one:

In the first step the user has to choose the software and features he wants to compare. Then he has to choose one of two available methods, AHP or SMART.

The screenshot shows the 'Which Soft?' application window with the 'Compare Software' tab selected. It features two columns of checkboxes for selection:

- Choose Softwares:**
  - ELECTRE III-IV
  - HIVIEW
  - PREFCALC
  - TOPSIS
  - WINPRE
  - ELECTRE TRI
  - LPA VISIRULE
- Choose Features:**
  - Cost of a license
  - Examples of applications
  - Free version
  - Interaction with the user
  - Online Help
  - OS Compatibility
  - User Manual and Tutorials

At the bottom right, there is a section for 'Step 1 Algorithm:' with two radio buttons:  SMART and  AHP. Below this are 'Next ->' and 'Reset' buttons.

Figure 8 - Compare software

SMART requires that you enter your preferences for each feature scoring each with values greater than or equal to 10. Then you must choose the method for each feature on step two.

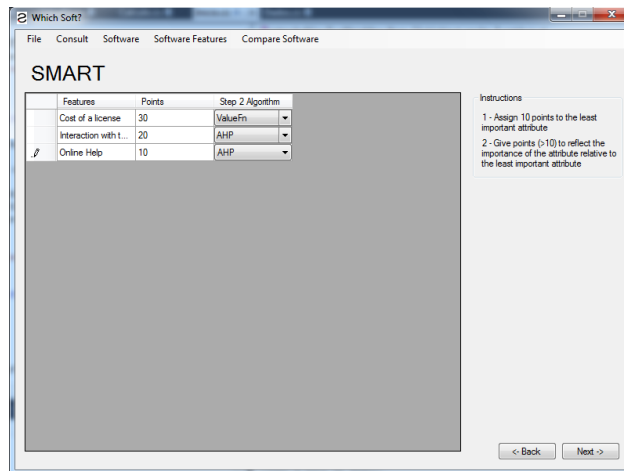


Figure 9 - SMART

With AHP the user has to define the relative importance of each characteristic compared to others, on a scale from 1 to 9. This scale was proposed by Saaty (1980), and 1 means “as important as”, and 9 “absolutely more important than” (see figure 10).

Filling the cell with the importance of the characteristic X compared to Y (say 4), will automatically be filled the cell with the importance of the characteristic Y compared to X (with 1/4). The diagonal is filled with the number 1 meaning a characteristic is “as important as” itself.

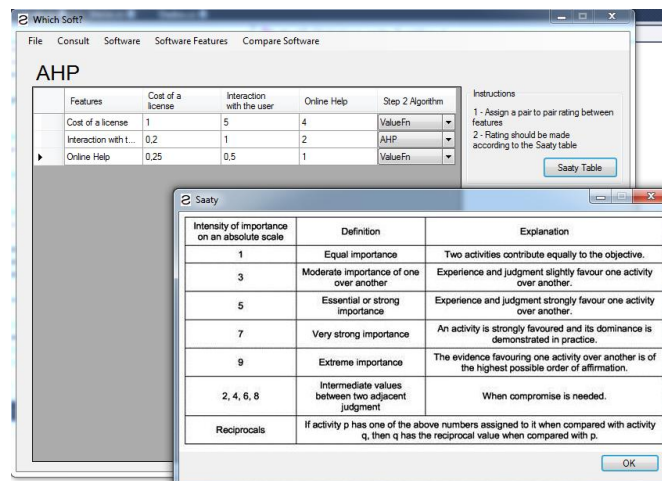


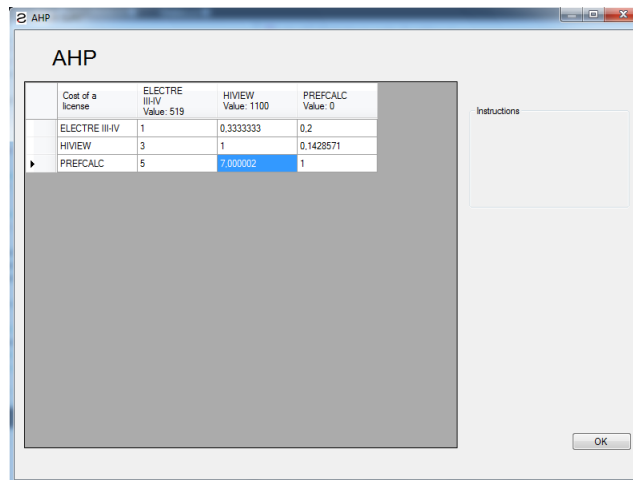
Figure 10 - AHP step one

### Step two:

In step two, each software has to be compared with others, regarding each of the characteristics. The methods available are ValueFn and AHP.

AHP was already explained in step one. An example of its application to step two can be seen in figure 11.





The screenshot shows the AHP software interface. At the top, it says 'AHP'. Below that is a table with the following data:

Cost of a license	ELECTRE III-IV Value: 519	HIVEW Value: 1100	PREFCALC Value: 0
ELECTRE III-IV	1	0.3333333	0.2
HIVEW	3	1	0.1428571
PREFCALC	5	7.000002	1

To the right of the table is an 'Instructions' text area. At the bottom right, there is an 'OK' button.

Figure 11 - AHP step two

The ValueFn method allows to maximize or minimize the values of each feature (see figure 12).

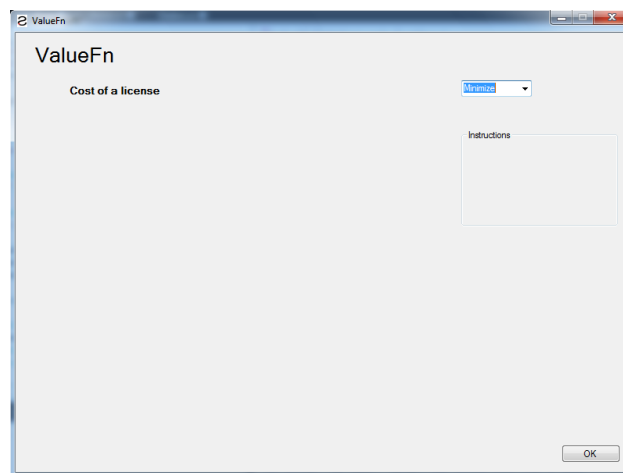
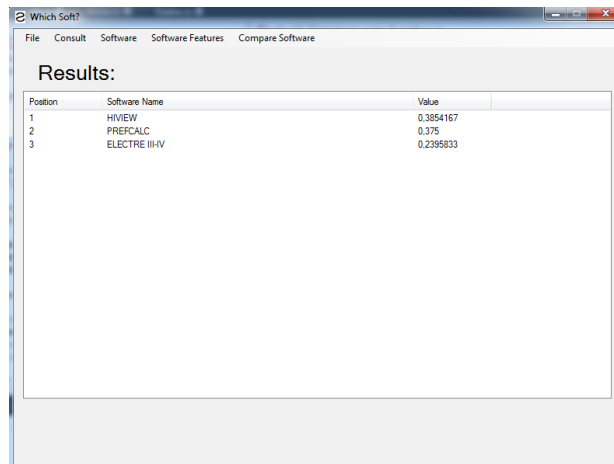


Figure 12-ValueFn

#### 4.4. Results

At the end the results appear as a list with software names and scores, in descending order. The highest scoring software is the best one.



Position	Software Name	Value
1	HIVIEW	0.3854167
2	PREFCALC	0.375
3	ELECTRE III-IV	0.2395833

Figure 12 - Results

## 5. Conclusions and future work

The working group has developed a practical software tool, useful for selecting the best alternative of a set of software alternatives, using multicriteria techniques. From the standpoint of the user, the platform has a set of features that distinguishes each software from others. The fact that features or characteristics reside in a database, allows the use of the Software at any time, to make other choices.

As future work, we have to invest in a more intuitive and attractive interface, in order to attract more users. We also would like to develop the software in order to allow sensitivity analysis.

## 6. References

- (1) von Winterfeldt, D., Edwards, W., 1986. Decision Analysis and Behavioral Research. Cambridge University Press, Cambridge.
- (2) Brugha, C. (1998), "Structuring and Weighting Criteria in Multi Criteria Decision Making (MCDM)", Trends in Multicriteria Decision Making: Proceedings of the 13th International Conference on Multiple Criteria Decision Making, Stewart, T. J. and Van den Honert, R.C. (eds.): Springer-Verlag, p. 229-242. [MCDA]
- (3) Saaty, T (1980), The analytical hierarchy process, McGraw-Hill, New York.
- (4) Fishburn, P.C. (1970) Utility theory for decision making, Research Analysis Corporation, John Wiley & Sons, INC.
- (5) Boehm, B.W. (1988) A spiral model of software development and enhancement, TRW Defense Systems Group, Computer.
- (6) <http://www.onestoptesting.com/sdlc-models/waterfall-model/waterfall-model.asp>