Examining a new Decision Making Technique and its application to the Software Procurement Process

Wayne S. Goodridge

***Abstract*— The process of ranking management information systems with the goal of finding the appropriate system to buy is complex. Measuring the importance of conflicting criteria by different stakeholders having varying interests can be challenging. Multi-criteria decision making (MCDM) offers promising solutions to this problem. The Analytic Hierarchy Process (AHP) and some of the ELECTRE methods are among the leading methods used in multi-criteria decision making. However, it is shown that these techniques suffer from rank irregularities which ultimately lead to inconsistent and unreliable decisions. This paper discusses the Routing Decision Support (RDS) method as a possible alternative to these techniques and the implementation of the method by way of a new software decision support system called Aki-Decisions. We examine the step by step process of how the Aki-Decisions software was used to make a recommendation for the selection of a Student Management System for the University College of Barbados.**

***Index Terms*— Analytic Hierarchy Process, ELECTRE, Multi-criteria decision making, Rank irregularities**

I. INTRODUCTION

The process of ranking management information systems with the goal of finding the appropriate system to buy is complex. In practice, managers may resort to simple heuristics to decide on alternatives that should be evaluated against competing criteria. In such complex situations structured approaches, such as multi-criteria decision making (MCDM), are sometimes used. The AHP [1] and some of the ELECTRE methods [2, 3, 4] are among the leading methods used in multi-criteria decision making. However, it was shown in [5] that these techniques suffer from rank irregularities. This means that the order of the alternatives of the decision process can change when one of the non-optimal alternatives is replaced by a worse, less desirable, one.

The Routing Decision Support (RDS) algorithm introduced in [6] is a MCDM technique that was originally used to find the best paths to route packets in a computer network based on a set of network constraints and user goals. This paper uses the RDS method as a basis for implementing a new software package called Aki-Decisions to be used to find the most appropriate alternative given a set of user criteria and constraints. It is believed that the RDS method does not suffer from the rank irregularity problem because of the type of scale employed in the decision process. A simple proof will be given in this paper to demonstrate that the RDS does not suffer from rank irregularities. Hence Aki-Decisions software implements the RDS method.

Any good software package that implements a MCDM model should free the decision maker from the technical implementation details of the model. The decision maker's focus should be on the fundamental value judgments and choices related to the assessment of each criterion**.** Visual and interactive techniques built into the software can help the decision maker communicate the value judgments of the criteria against each alternative and evaluate the results. A unique contribution of the Aki-Decisions software is the way it allows the user to communicate his/her value judgments and choices.

The paper is organized as follows: Section 2 describes the RDS algorithm and illustrates, by way of a simple example, how it works. Section 3 discusses how the Aki-Decisions software system implements the RDS algorithm. In section 4 we present a Case Study in which we highlight the use of the Aki-Decisions software in the procurement process of a student management system, and we provide an analysis of the decision process. The paper concludes with some directions for future research.

II. THE RDS ALGORITHM

In this section a decision problem will be defined and the RDS algorithm will be explained. An example will also be given to illustrate how the RDS can be used to assist with a basic decision problem.

A. Problem definition

Multi-criteria decision making (MCDM) methods can help to improve the quality of decisions by ensuring that the decision-making process is more explicit, rational, and efficient [7]. As a consequence MCDM methods are used widely in engineering [8], manufacturing [9], and business environments [10]. Most MCDM techniques have been shown to have irregularities in terms of the reliability of the ranking

Manuscript received January 10, 2011.

W. S. Goodridge is with the University of the West Indies, St. Augustine, Trinidad in the Department of Computing and Information Technology. Wayne.goodridge@sta.uwi.edu

of the best alternatives [5].

A MCDM problem comprises a row vector $\vec{w}$, of size m, which contains a set of weights, and an $n \times m$ matrix X. The matrix consists of values x_{ij}, where $i \in [1,n]$, $j \in [1,m]$, n is the number of alternatives, and m is the number of criteria.

Alternatives

Weights $\begin{bmatrix} w_1 & w_2 & \dots & w_m \end{bmatrix}$

$$\begin{matrix} A_1 \\ A_2 \\ \\ A_n \end{matrix} \quad \begin{Vmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ - & - & - & - \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{Vmatrix}$$

Criteria $C_1 \quad C_2 \quad \dots\dots C_m$

Fig 1: Structure of Decision Problem

As shown in Fig 1, element x_{ij} of the matrix in the decision problem represents the performance value of the *i-th* alternative, A_i, in terms of the *j-th* criterion, C_j.

Now suppose that a decision algorithm Z determines that the list of alternatives should be ranked as follows: $A_1 \geq A_2 \geq A_3 \geq \dots\dots \geq A_n$.

In the above list A_1 is the best alternative followed by A_2 and so on. Now suppose A_3 is replaced by A_e where A_e is worse than A_3 in all respects. If algorithm Z is executed again and the new list looks like $A_2 \geq A_1 \geq A_e \geq \dots\dots \geq A_n$ or $A_1 \geq A_e \geq A_2 \geq \dots\dots \geq A_n$ then algorithm Z is said to have irregularities and cannot be considered to be reliable.

B. Introduction to the RDS

The RDS algorithm [11] uses the structure of the decision problem in Fig 1. The algorithm uses a scale s_j to convert values of x_{ij} for each criterion $j \in [1,m]$ into a dimensionless value in the range $[-1,1]$. The scale s_j can be any monotonic continuous increasing or decreasing function whose domain exists in $[-1,1]$.

Equation (1) shows the expression used to calculate the linear preference value for a given criterion against an alternative. The s_j function behaves differently for benefit and cost criteria [11]. A benefit criterion means that the higher the score of an alternative in terms of that criterion, the better the alternative is. The opposite is considered true for the cost criteria. For benefit criteria the best value for a given criterion is the value with the greatest magnitude and the worst value has the lowest magnitude.

In Fig 2 we see that the RDS algorithm is organized into three (3) major identifiable components during the decision process. The first component is the formulation of a decision matrix and the scaling of each element in this matrix based on scale s_j (defined in equation (1)). In the second component, the RDS algorithm allows a given decision maker to express a set of tradeoffs d_j (in the form of a vector $\vec{d}$) for each criterion. The tradeoffs are then multiplied term by term by elements w_j of the weight vector $\vec{w}$. Weights w_j are assigned to each criterion such that $\sum w_j = 1$. It should be noted that a decision maker's set of tradeoffs is different from the weightings of the criterion for a given decision problem. Finally, the third component is the generation of an output column vector, $\overline{y}$, of size m. Each position in this vector is associated with an alternative, and the position that carries the largest value is the best alternative. An ordering of the elements of $\overline{y}$ will result in a ranking of the decision alternatives.

$$s_j(x) = \begin{Bmatrix} 2\dfrac{(x_{ij} - b_j)}{(a_j - b_j)} - 1, & a_j \neq b_j \\ 1, & a_j = b_j \\ 1, & x_{ij} > a_j \\ -1, & x_{ij} < b_j \end{Bmatrix} \qquad (1)$$

Illustrative example - Buying a House

Suppose David wishes to buy a house from the set of houses in Table I, and location, distance to main city and number of rooms, are important criteria in helping him make his choice. He wishes to minimize cost, minimize location to city and maximize the number of rooms in the house. He is very concerned about the price (weighting = 0.4) and distance to city (weighting = 0.4) and cares less about the number of rooms (weighting = 0.2). However, he wishes all three factors to be considered in the decision process and so his tradeoff vector $d_j = (1,1,1)$. The values in Table I are scaled by using equation 1 to produce the 3 x 5 matrix shown in Fig 3. Based on the weightings of the sorted output vector in Fig 3, the RDS output would be:

$$House_5 \geq House_2 \geq House_1 \geq House_3 \geq House_4$$

House 5 is indeed a suitable choice since it has the lowest price, shortest distance to city and most rooms. The intent of this simple example is just to show how the RDS works.

C. RDS does not suffer from Rank Irregularities

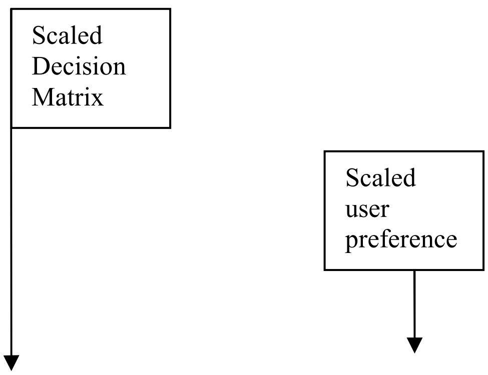
In this Section we will show that the RDS method does not change the indication of the best alternative when a non-optimal alternative is replaced by another worse alternative. Suppose that the RDS method produces the following ranking for a set of alternations:

$A_1 \geq A_2 \geq A_3 \geq \dots\dots \geq A_n$. Based on equation 1:

$$s_j(x) = 2\frac{(x - b_j)}{(a_j - b_j)} - 1$$

Table I: Set of Houses and their properties (criteria)

	Price	**Distance to City (km)**	**No. of Rooms**
House 1	150000	10	2
House 2	120000	6	3
House 3	180000	12	4
House 4	230000	7	2
House 5	130000	5	5



$$\begin{pmatrix} s_1(x_{11}) & s_2(x_{12}) & .... & s_m(x_{1m}) \\ s_1(x_{21} & s_2(x_{22}) & .... & s_m(x_{2m}) \\ - & - & - & - \\ s_1(x_{n1} & s_2(x_{n2}) & ... & s_m(x_{nm}) \end{pmatrix} \begin{pmatrix} d_1 \bullet w_1 \\ d_2 \bullet w_2 \\ - \\ d_m \bullet w_m \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ - \\ y_m \end{pmatrix} \Rightarrow \begin{pmatrix} pos(y_1) \\ pos(y_2) \\ - \\ pos(y_m) \end{pmatrix}$$

Fig 2: Pictorial view of concepts that make up the RDS decision process

$$\begin{pmatrix} 0.454 & 0.429 & -1.000 \\ 1.000 & -0.714 & -0.333 \\ -0.091 & -1.000 & 0.333 \\ -1.000 & 0.429 & -1.000 \\ 0.818 & 1.000 & 1.000 \end{pmatrix} \begin{pmatrix} (1)(0.40) \\ (1)(0.40) \\ (1)(0.20) \end{pmatrix} = \begin{pmatrix} -0.190 \\ 0.619 \\ -0.367 \\ -0.429 \\ 0.927 \end{pmatrix} \Rightarrow \begin{pmatrix} 0.927 \\ 0.619 \\ -0.190 \\ -0.367 \\ -0.429 \end{pmatrix}$$

Fig 3: Pictorial view of RDS algorithm for finding the best House to buy

Now suppose alternative $A_p, p \in [2, n]$ is selected from the list of alternatives. If $E = A_p$ is replaced by a dominant (non-optimal) alternative, say F, then by definition of "dominance":

$$E_1 \geq F_j, \forall j \in [1, m]$$

For the new set of alternatives let

$$p_j(x) = 2\frac{(x - b_j)}{(a_j - b_j)} - 1 = 1 + \frac{2(x - a_j)}{(a_j - b_j)}$$

The worse values for criteria are represented by b_j. Since introducing a new non-optimal alternative will cause b_j to decrease for $\forall j \in [1, m]$, then because of equation (1), a_j - b_j increases.

Therefore, $\frac{2(x - a_j)}{(a_j - b_j)}$ decreases.

So that $s_j(x) \geq p_j(x)$ and hence

$\sum_{j=1}^{m} s_j(x_{ij}) s_j(d_j) w_j \geq \sum_{j=1}^{m} p_j(x_{ij}) p_j(d_j) w_j$ due to the fact that the user preferences $s_j(d_j)$ and w_j are the same for both the new and old list of alternatives.

Since A_1 had the highest value of $\sum_{j=1}^{m} s_j(x_{ij}) s_j(d_j) w_j$ and was not replaced, and every real point in the new alternative list $\sum_{j=1}^{m} p_j(x_{ij}) p_j(d_j) w_j$ is less than the original alternative list, it follows that A_1 would retain the same position in the output vector of the RDS, with the highest value.

D. Comparing the RDS and AHP Approaches

The AHP and RDS techniques are both examples of weighted approaches to the MCDM problem. Both the AHP and the RDS methods can combine qualitative and quantitative criteria and turn them into a standardized numerical scale. For qualitative criteria the AHP uses a scale between 1 and 9 and normalizes values to the interval [0, 1]. On the other hand, for qualitative criteria, the RDS does not use numbers directly to represent user preferences but instead asks the user to rank the value for each criterion against all the alternatives. A number between 1 and 100 is assigned to each alternative for a given criterion based on the relative positions of each alternative. The highest and lowest values are then used with equation (1) to generate the normalized values for the criterion for each alternative in the interval of [-1, 1]. The use of the highest and lowest values to generate the normalized values is the key difference between the AHP and the RDS approaches. The RDS uses an interval/ratio scale which results in a system that allows the operations of multiplication and addition to be define on [-1, 1]. The scale used in the AHP is ordinal at best. A ranking of 10 does not mean that the preference for an item is twice that of an item rated 5.

The AHP approach uses pair comparisons for determining the tradeoffs between alternatives for each criterion. However, this approach can lead a decision maker making inconsistent decisions since only two alternatives are involved in the tradeoffs process at any given time. However, in the RDS approach the decision makers compares a given criterion against all alternatives, making if difficulty for inconsistencies to be introduced.

III. AKI-DECISIONS IMPLEMENTATION OF THE RDS ALGORITHM

Software is an important element in the application of MCDM methodologies because of the considerable amount of computation involved. This does not mean that to have a good software tool is sufficient to apply an MCDM methodology correctly. Software is just a tool and before using the software a sound knowledge of the decision methodology must be adopted. In this section the basic architecture of the Aki-Decisions software will be examined, how it interfaces with users who evaluate criteria against alternatives, and finally a look at the user roles that are typically involved in the decision making process.

A. AKI-DECISIONS Architecture

Aki-Decisions software is written in Delphi 6 and uses an Access database for storage of decision alternatives and criteria. The software was created primarily to implement and evaluate the RDS method. Fig 4 shows the architecture of the software which facilitates different MCDM algorithms. The Common Object Platform (COP) layer allows different MCDM methods to be used while providing common storage interface for the database. Later versions of the software will allow decision makers to use different MCDM methods to enable them to compare the methods.

Aki-Decisions software has five major functional goals. These are:

1. To facilitate input of criteria and alternatives - the software allows users to enter a list of alternatives and their criteria. For each criterion the user has to specify whether or not it is quantitative or qualitative, and cost or benefit.
2. To allow decision facilitator (see next section) to management contribution of stakeholders.
3. To facilitate the entry of decision makers' qualitative and quantitative evaluations of criteria.
4. To allow decision makers to visualize relationships between criteria.
5. To allow workflow management of tasks relating to the setup of the decision problem.

The discussion of how Aki-Decisions software meets the functional goals 1, 2, 3 and 4, outlined above, is provided in the Case Study. Goal 5 deals with workflow management. This goal is achieved in the Aki-Decisions software by structuring the steps the decision maker goes through in the decision process. The software has a wizard that firstly allows the users to enter a list of alternatives. The steps of the decision process that the wizard goes through are shown in Fig 5. The next step in the decision making process is to define the criteria used to evaluate the alternatives followed by setting up of the decision matrix which basically contains criteria values for each alternative. The setup of the expert weightings option allows the decision facilitator (see next section) to assign weights to different criteria for a given expert based on the expert's experience on the given criterion. The next step is for the facilitator, based on the advice of evaluators (see next section), to arrange criteria into groups and sub-groups and assign weightings for each criterion. This step seems like it should logically come after setting up the decision criteria and will be changed in a later version of the software: arranging the criteria into groups and sub-groups needs to come after defining of decision criteria and this is what is important. The final step is for the software to allow the decision maker to calculate the best option.

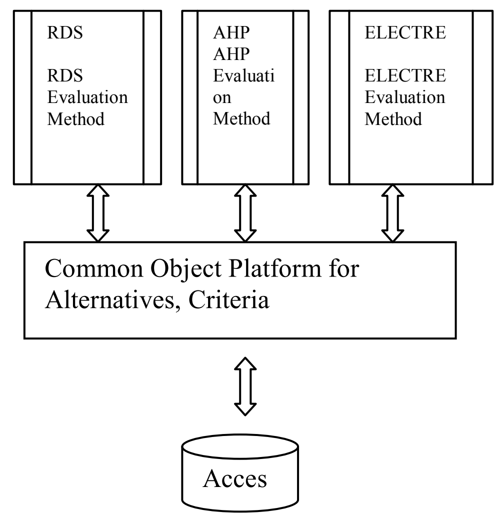


Fig 4: Architecture of the Aki-Decisions Software

B. AKI-DECISIONS Software Interface

The decision maker's focus should be on the fundamental value judgments and choices. To help the decision maker achieve this, the Aki-Decisions software does not use a numeric scale to describe the values for qualitative criteria because doing so would force the use of an improper scale [11]. For example, suppose in the example given earlier we included a qualitative criterion such as car colour. Assigning numeric values, say between 1 and 9, to each car colour would be confusing since a car with a value of 8 is not necessarily "twice" preferred to a car with a value of 4. Fig 7 shows a sample screen shot for assigning values to qualitative scales. Here the user selects the best and worst alternatives and uses a visual comparative scale to measure the importance of the criterion with respect to each alternative.

C. User roles in Computerized Decision Making

In a computer assisted decision making environment participants can play different roles. Some of the roles that exist are:

1. Facilitator: manages the interaction between stakeholders of the decision process.
2. Alternative designer: proposes some alternatives/actions
3. Criterion designer: defines criteria on which alternatives

are to be evaluated
4. Evaluator: evaluates alternatives on criteria
5. Decision Maker: expresses preferences on alternatives and criteria

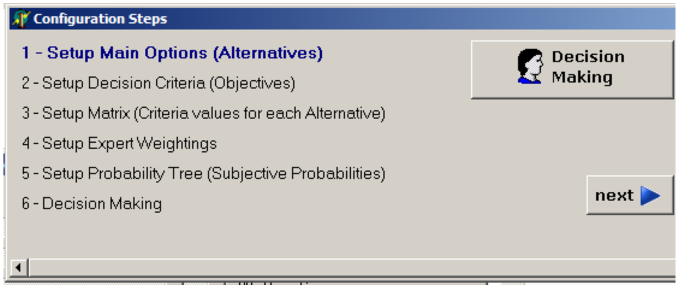


Fig 5: Steps involved in the decision making process with the Aki-Decisions Software

A Stakeholder can possess several of the above roles simultaneously. To use the Aki-Decisions software successfully clear roles must be set for all participants in the decision making process. Usually the organization requiring a solution to a problem would propose a set of alternatives or hire a consultant to come up with a set of alternatives. Hence the consultant or organization concerned would be taking the Alternative Designer role. Summarily, Criterion Designer can be distinct from the Alternative Designer and is responsible for creating the criteria on which the decision will be based. The Facilitator of the decision process acts like a project manager and coordinates all the roles that stakeholders play. In addition, the Facilitator is most likely the person that will be the administrator of the software in terms of configuration management.

IV. CASE STUDY – PROCUREMENT PROCESS FOR STUDENT MANAGEMENT SYSTEM (SMS)

The Government of Barbados recently decided to amalgamate three tertiary institutions under the banner of the University College of Barbados (UCB). The acquisition of computerized integrated student management systems was one of the main aspects of the amalgamation. The process of selecting a management information system that can meet the diverse needs of faculty, administration, and students systems is complex. In practice, managers may resort to simple heuristics or rules of thumb to decide on alternatives that should be evaluated against competing criteria. However, in such complex situations structured approaches, such as multi-criteria decision making (MCDM), can be used. We now examine how the Aki-Decisions software is used in the decision process.

A. The Decision Process

A sub-committee comprising members from the Board of Management of each institution was responsible for deciding which selection criteria would be used in the decision problem. In this section the decision process used by the UCB is presented. The decision making process can roughly be divided into four (4) steps.

Step 1

- *Determine the set of alternatives.*

The University College of Barbados (UCB's) project management office examined proposals from the following three Student Management Systems:

1. Comprehensive Academic Management System (CAMS) – produced by Three Rivers Systems and used by 300 medium sized colleges in the USA.
2. PowerCAMPUS – produced by SunGard Higher Education and used by 150 medium sized colleges in USA. SunGard is also responsible for the Banner SMS which is used by the University of the West Indies.
3. Colleague – produced by Datatel and used by over 700 colleges in the USA. Colleague runs on different operating systems platforms including Unix and Windows. The windows version of Colleague is used by 110 colleges.

Step 2

- *Determine the criteria (objectives) under which the systems would be assessed.*

The following criteria were decided upon by the UCB's project manager and the IT staff at the Barbados Community College (BCC) and Samuel Jackman Prescod Polytechnic (SJPP):

1. **Annual Maintenance** This is a percentage of the cost of the software that is paid to the vendor every year for updates and licensing of the software. This can vary between 15% and 22% or be fixed.
2. **Automatic Course Evaluation** This is the system's ability to allow a student to evaluate a course. The system is then able to produce statistics which can then be used by administration for further decision making.
3. **Connection Strategy** Whether or not the connection of workstations to the server is browser based or client server. Client server connections are usually more difficult to manage than browser based connections.
4. **Cost of Software** This is the cost of the basic package of the Student Management System with regard to the modules that the University College of Barbados (UCB) will be using.
5. **Implementation Cost** This is money spent training users, configuring system to environment and converting existing data into new system.
6. **Course Management** This allows the institution to offer courses online. Typically, students are able to participate in discussion forums, chat and do online tests.
7. **Ease of ad hoc Reporting** This is a measure of how easy it is to produce an 'on-the-fly' report within the system.
8. **Ease of IT Support** This is a measurement of how much IT support will be required for the system. e.g.

Is a database administrator required?

9. **Implementation Cost** This includes the cost of bringing in the vendor or his representative, accommodation, travel and consultancy fees.
10. **Impression of Web Interface** This is the look and feel of the vendor's portal.
11. **Interface with Active Directory** This is the system's ability to authenticate with the operating system's security system.
12. **Software Support Quality** This deals with the quality and efficiency of support when the client seeks assistance.
13. **Software Support Model** Deals with the plan that the vendor has for supporting software, hours allocated per month whether or not a person is available at the next end to support any questions. For example, in the case of Sunguard, the first line of support is not a person but a reference to historic information on problems and solution, whereas CAMS has a person at the other end of the line when you call.
14. **Scalability** This is a measure of the software ability to adapt to increase of student numbers without experiencing increase in response times.
15. **Portal Individual Customization** This is the ability of the user of a portal to customize it to his/her needs.
16. **Adoption of 151 user requirements** This is the ability of the vendor to meet the 151 user requirements outlined by the IT staff of BCC and SJPP.
17. **Functionality of Student Management** This is the perceived functionality of the vendor's student management system.
18. **Functionality of Accounting System** This is the perceived functionality of the vendor's accounting module.
19. **Functionality of Fund Raising Module** This is the perceived functionality of the Fund Raising Module.
20. **Functionality of Scheduling System** This is the perceived functionality of the vendor's Scheduling Module.
21. **Technical Capability** This is a subjective measurement of the vendor's technical capability.
22. **Usability** This is a measure of the user friendliness of the system.

Each criterion was then categorized into qualitative and quantitative criteria. For quantitative criteria such as cost of software and implementation cost, a linear preference function (see table ii) was used to minimize the cost. For qualitative criteria graphical scales were used with summarized data from the UCB project office and IT staff. For example, each qualitative criterion was measured using a window similar to that shown in Fig 6. Each decision maker will repeat this for each criterion.

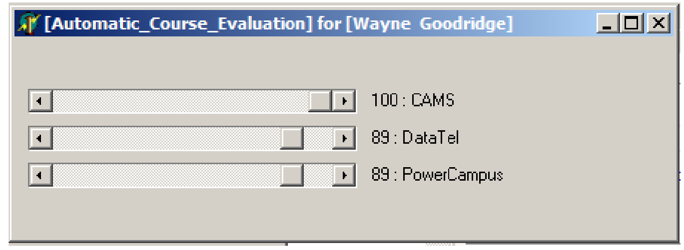


Fig 6: Example of how qualitative criteria are measured (In this example, the accounting function is measured based on feedback from user Wayne Goodridge.)

Table II: Properties and their possible values

Properties	**Possible values**
Name	(Name of decision criterion)
Scale Model	Linear, Exponential, Negative Exponential
Type	Qualitative, Quantitative
Direction	Cost, Benefit

Step 3

- *Assign values to the decision matrix and user in the decision group.*

The columns of the decision matrix are the criteria used in the decision making process and the rows represent each alternative and corresponding value for the criterion as shown in Fig 7. It is important to note that for some qualitative criteria like Software Support and Ease of Use that no actual values are given in the decision matrix since a screen like the one in Fig 6 is used for measuring the impact of the criteria for each alternative.

Step 4

- *Assign weightings to each criterion.*

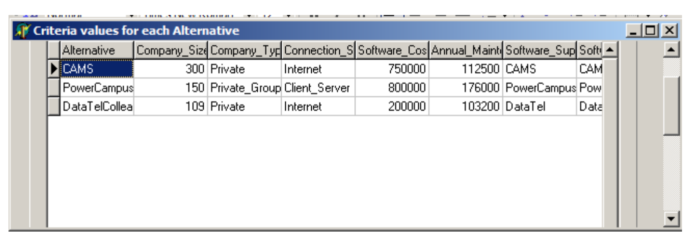


Fig 7: Decision Matrix for the UCB's procurement problem.

Fig 8 shows the weightings given to the system by the UCB's project office. The current version of the Aki-Decisions software only allows for one set of weightings to be expressed in the system. Hence it was necessary for the decision makers on the UCB committee to agree on the collective weightings for each criterion.

In addition to assigning weightings to each criterion it is important to measure the level of confidence that should be used by each member in the decision making team with respect to each criterion. For example, the Registrar of the

UCB may have a good understanding of the student management functions required by the UCB's student services department. However, in relation to what is involved in assessing a good course management system, the Registrar may be at a loss.

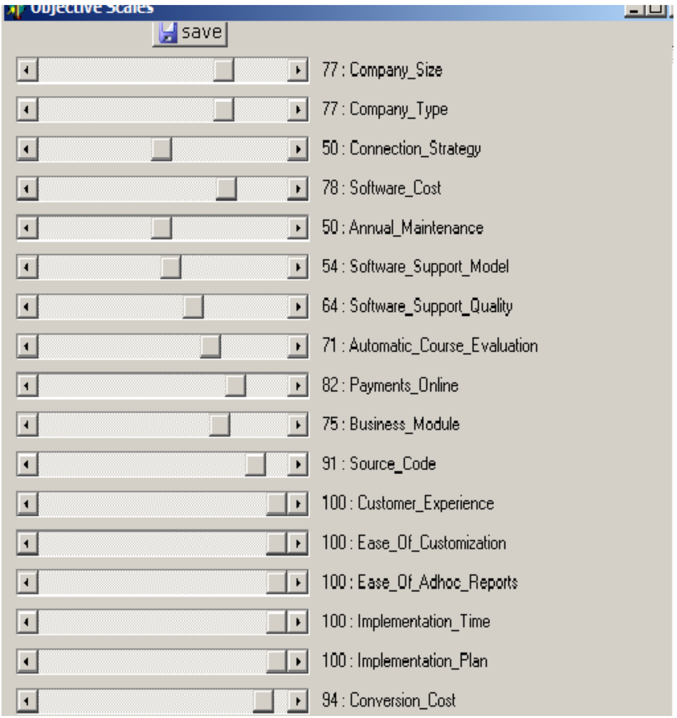


Fig 8: Weightings for the criteria.

The Aki-Decisions software addresses this issue by allowing the Facilitator of the system, after consultations with the decision makers, to assign to each criterion, for a given member of the decision group, a value of average, above average and expert. The system uses a weighting system defined by the Facilitator acting on the advice of the decision makers to bias the contribution by a user for a given criterion. The result is that the measurement for the subjective scale (see Fig 6) is an aggregate of all the decision makers rather than a single decision maker. Fig 9 illustrates how this is achieved by the Aki-Decisions software.

B. Results of the Decision Process

The RDS system uses the concept of a preference scale [13] to measure the effect of each criterion on a given alternative. Table III shows the measured preference for each criterion across the alternatives. The preference scale range for a given criterion j is $-w_j \le s_j \le w_j$. That is, every criterion is measured on a separate scale relative to its importance in the decision problem. For each row in the table, if the maximum value is w_j, the minimum value is $-w_j$. This is consistent with the notion of using the best and worst values of each criterion to be the basis for the preference scale. The DataTel Colleague alternative has the highest preference value overall. This means that this is the preferred alternative.

The sub-committee of the Board of Management responsible for selecting a student information system for the new institution was very happy with the results of the Aki-Decisions software. This was because the use of preference ranking charts helped them to easily understand the relative strengths and weaknesses of the three systems. The decision makers were happy with the fact that the decision making process was scientific and therefore trusted the systems recommendations.

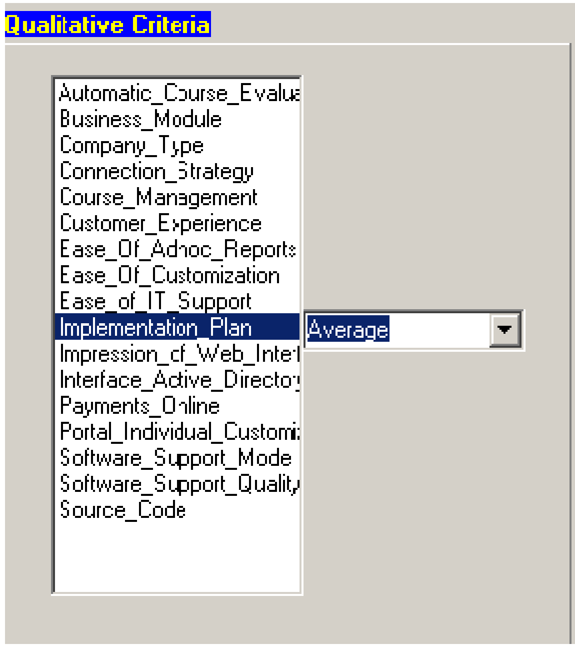


Fig 9: Bias assigned to a user with respect to a given criterion.

V. CONCLUSION

This paper demonstrates how the nontrivial task of selecting an expensive student management system can be simplified by using a structured decision making approach. Measuring the importance of conflicting criteria by different stakeholders having varying interests can be challenging. However, the decision wizard provided in the Aki-Decisions software helped with a systematic approach to the problem. The mathematical approach used to rank the alternatives is the key to the Aki-Decisions software. The authors of this paper believe that the results of many of the existing MCDM approaches are unreliable due to the fact that they suffer from rank irregularities. Therefore, the relatively new technique of the RDS method is used to implement the Aki-Decisions software. In future work we will show that the RDS method does not suffer from rank irregularities. In future work we will also show how other techniques like AHP and ELECTRE can be implemented in the Aki-Decisions software by using the COPs layer. We hope that the technique is adopted and applied to as many domains as possible.

Table III: Preference values for each criterion across the three alternatives

Criteria Name	DataTel Colleague	Power Campus	CAMS
Connection Strategy	0.03	-0.03	0.03
Software Cost	0.05	0.04	-0.05
Annual Maintenance	-0.01	0.05	-0.05
Software Support Model	0.03	0.03	-0.03
Quality of Software Support	0.03	0.02	-0.03
Automatic Course Evaluation	-0.03	-0.03	0.03
Ease of Customization	0.03	0.03	-0.03
Ease of Adhoc Reports	0.03	0.02	-0.03
Impression of Web Interface	-0.05	0.05	0.01
Interfacing with Active Directory	0.02	0.03	-0.03
Portal Individual Customization	-0.02	0.03	-0.03
Course Management	-0.03	-0.03	0.03
Ease of IT Support	0.02	-0.04	0.04
Implementation Cost	-0.03	-0.05	0.05
Additional Software Cost	0.03	-0.03	-0.03
Scalability	0.03	0.02	-0.03
Adoption of 151 UCB requirements	0.06	0.05	-0.06
Functionality Student Management	0.05	0.07	-0.07
Functionality of Accounting Module	0.07	0.07	-0.07
Functionality of Fund Raising Module	0.06	0.02	-0.06
Functionality of Scheduling Module	0.03	-0.07	0.07
Technical Capability	0.04	0.03	-0.04
Vendor Stability	0.03	0.03	-0.03
Totals	0.47	0.31	-0.41

REFERENCES

[1] Saaty, T.L. (1994) *Fundamentals of Decision Making and Priority Theory with the AHP*. RWS Publications, Pittsburgh, PA.

[2] Roy, B. (1968) ‘Classement et choix en présence de points de vue multiples: La méthode ELECTRE’, *Revue Francaise d'Informatique et de Recherche Opérationnelle*, Vol. 8, pp.57-75.

[3] Roy, B. (1978) ‘ELECTRE III: Un algorithme de classements fonde sur une representation floue des preference en presence de criteres multiples’, *Cahiers de CERO*, Vol. 20, No. 1, pp.3-24.

[4] Roy, B. and Bertier, P. (1971) ‘La methode ELECTRE II: Une methode de classement en presence de critteres multiples’, SEMA (Metra International), Direction Scientifique, Note de Travail No. 142, Paris, 25p.

[5] Wang, X. and Triantaphyllou, E. (2008) ‘Ranking Irregularities When Evaluating Alternatives by using Some ELECTRE Methods’, *Omega*, Vol. 36, pp.45-63.

[6] Goodridge, W., Robertson, W., Phillips, W. and Sivakumar, S. (2005b) ‘Heuristic Constraint-Path Routing Decision System’, *Proceedings of the 3rd Annual Conference on Communication Networks and Services Research (CNSR)*. Halifax, Nova Scotia, pp.3- 8.

[7] Accorsi, R., Apostolakis, G. E. and Zio, E. (1999) ‘Prioritizing stakeholder concerns in environmental risk management’, *Journal of Risk Research*, Vol. 2, No. 1, pp.11-29.

[8] Bonano, E. J., Apostolakis, G. E., Salter, P. F., Ghassemi, A., and Jennings, S. (2000) ‘Application of risk assessment and decision analysis to the evaluation, ranking and selection of environmental remediation alternatives’, *Journal of Hazardous Materials*, Vol. 71, pp.35-57.

[9] Wabalickis, R.N. (1988) ‘Justification of FMS with the Analytic Hierarchy Process’, *Journal of Manufacturing Systems*, Vol. 17, pp.175-182.

[10] Boucher, T.O. and Mcstravic, E.L. (1991) ‘Multi-attribute Evaluation within a Present Framework and its Relation to the Analytic Hierarchy Process’, *The Engineering Economist*, Vol. 37, pp.55-71.

[11] Goodridge W., Nikiov, A., Sahai, A. (2008) ‘Study of the effectiveness of the routing decision support algorithm’, International Journal of Electronics, Circuits and Systems, Vol. 1, No. 4, pp.223-228.

[12] Triantaphyllou, E. and Baig, K. (2005) ‘The Impact of Aggregating Benefit and Cost Criteria in Four MCDA Methods’, *IEEE Trans. on Engineering Management*, Vol. 25, No. 2, pp.213-226.

[13] Barzila, J. (1997) ‘A new methodology for dealing with contradicting engineering design criteria’, *Proceedings of the 18th Annual Meeting of American Society for Engineering Management*, 73-79.