Decision Support in a Telecommunications Engineering E-Learning Platform

B. Rodič
Faculty of Information Studies, Novo mesto, Slovenia
Blaz.Rodic@fis.unm.si

Abstract - This contribution presents the development of a decision support system to be used by telecommunications engineering students in selecting a suitable frequency filter design tool. System design follows the expert system paradigm and implements multi-criteria decision modelling using the Decision Matrix method, and combines set theory based option filtering with the MAUT method for option ranking. The expert system inference engine is rule based, while the knowledge base is designed as a Decision Matrix and implemented in a relational database. The user interface is to offer two types of user experience for beginners and advanced users. The system is developed as an integrated part of a telecommunications engineering online learning system, and is designed to be open-ended to incorporate additional filter design tools and decision criteria.

Keywords – telecommunications; e-learning; filter design; decision support; multi criteria decision analysis; expert system

I. INTRODUCTION: FILTER TOOL SELECTION AS A DECISION PROBLEM

The selection of an appropriate filter design tool in telecommunications engineering projects requires considerable knowledge of the range of available tools, their characteristics and relevant criteria for the required filter type. Such expert knowledge is typically acquired through engineering experience.

In order to facilitate the learning process and the selection of the filter design tool for experienced and inexperienced users, we have decided to construct an expert system (ES) - a decision support tool that would incorporate the knowledge base of experienced telecommunications engineers and researchers and could be integrated within the online learning system under development by the Faculty of Telecommunications, Technical University of Sofia (TUS) and described in [1], [2], [3] and [4]. The goals of this system’s development is to integrate available online tools solving different computer-aided design tasks and combining tools for complex task solving. Filter design tools are integrated in the online learning system after verification, estimation, classification and characterization.

A survey of the filter design tools was performed to gather the potential tool selection criteria, and the relevant selection criteria were then selected by the engineering team and collected in a table containing the criteria types, their names, description, and possible values. The list of approximately sixty filter design tools characteristics was reduced into a set of eighteen selection criteria, e.g. frequency range, approximations model, roll-off slope, sensitivity, noise, etc. and grouped into criteria types [4].

As there are several criteria present in the filter design tool selection problem, we have examined several Multi Criteria Modelling Analysis (MCDA) methodologies in the development process and implemented a combination of a Decision Matrix (DM) and MAUT method for option filtering and ranking.

In this contribution we present the original decision problem, the decision support tool development, and the decision modelling methodology used within our collaboration project with TUS. Our joint development goal is a working, interactive web-based filter design tool selection system.

II. METHODOLOGY: MULTICRITERIA DECISION MODELLING AND EXPERT SYSTEMS

Several decision modelling methods were examined in the development of the presented solution. A good overview of MCDA methodologies and tools is presented in (Bohanec, 2006) and (Kolios, et al., 2016). According to (Kahraman & Kaya, 2010), successful selection of the most appropriate multi-criteria methodology should consider a range of different perspectives in order to comprehend all sides of the problem and, when necessary, consider interconnections among the criteria. MCDA methods need to structure the decision procedure, to demonstrate the trade-off among the criteria, to assist decision-makers to reflect upon, articulate and apply worthy judgments related to satisfactory trade-offs, resulting in suggestions when considering alternatives, to estimate risk and uncertainty more consistently and reasonably, to simplify negotiation and to keep a record of how decisions are made.

According to [5], an ES can be defined as: a program that attempts to mimic human expertise by applying inference methods to a specific body of knowledge. Bohanec and Rajković [6] describe ESs as intelligent information system that behave, in a certain sense, as a human expert in the application domain. According to [6], ESs are typically composed of two modules: (1) a knowledge base and (2) an inference engine. The knowledge base contains the knowledge about a particular problem domain. The most common methods of knowledge [6] are production (if-then) rules, semantic nets and frames. In addition, these formalisms are usually
capable of dealing with imprecision, uncertainty, and qualitative (nonnumeric) nature of the expert knowledge.

Grosan and Abraham [7] describe rule-based systems (also known as production systems or ESs) as the simplest form of artificial intelligence. A rule based system uses rules as the knowledge representation for knowledge coded into the system. Instead of representing knowledge in a declarative, static way as a set of things which are true, rule-based system represent knowledge in terms of a set of rules that tells what to do or what to conclude in different situations. Rule-based ESs [8] are used as a way to store and manipulate knowledge to interpret information in a useful way. They are often used in artificial intelligence applications and research. A classic example of a rule-based system is the domain-specific ES that uses rules to make deductions or choices. Rule-based systems have been used to automate problem-solving know-how, provide a means for capturing and refining human expertise since the 1980s [8].

The decision-matrix method, also known as the Pugh method and Pugh Concept Selection [12], is a quantitative technique used to rank the multidimensional options of an option set, frequently used in engineering, and applicable to other multi-criteria ranking or selection decision. The DM consists of a set of criteria, usually displayed in rows, and a set of options, usually displayed in columns. The relation between a criterion and an option is entered in the intersecting cells, and can represent the presence or absence of a quality described by the criterion, or a qualitative or quantitative value of the option for the criterion. In case of quantitative values, the DM can be used as a quantitative MCDA model, however its structure imposes the limitation to a single level of criteria, i.e. hierarchical MCDA models cannot be formulated as a DM.

The solution presented in this contribution is related to the system described by [13], as a prototype ES that helps software project managers and software engineers in selecting the appropriate software development methodology, and to the system described by [14], who present a rule based recommendation system that can be helpful to software developers in selecting the most appropriate SDLC model to be used for the development of a software product and to the solution presented by [15] whose paper presents an ES based rapid prototyping (RP) system selection program incorporating the data on 39 commercially available RP systems, and finally [16], who describe a decision support system using qualitative and quantitative criteria in multicriteria decision tree for project planning support.

### A. Multi-Attribute Decision Analysis

One of the widely used approaches to decision support is multi-attribute decision analysis (MCDA) ([17] and [18] in [6]). The main concept of MCDA is the decomposition of a decision problem into smaller, simpler sub problems with the decomposition repeated until each elementary sub problem can be represented as a single value, which are referred to as attributes, parameters or criteria.

The resulting decision problem structure can be represented as the tree of attributes $X$ as shown in Fig. 1. Each option $O$ is modelled as a vector of values $x$ of the corresponding attributes. The vectors are then evaluated by a utility function $F$. This function is defined by the decision maker(s) and represents their goals. When applied upon a particular option $O_k$, the function $F$ yields a utility $Y_k = F(O_k)$. Options can be ranked according to $Y$ and usually the best option is chosen (decision is completed). In the multi-attribute paradigm, the decision makers’ knowledge about a particular decision problem is therefore described by attributes $X$ and a utility function $F$ while the options are represented as vectors (where the $i$-th component is represented by a variable $x_i$). [6]

The utility function for option $O_k$ for the multi criteria decision model shown in Fig. 1 would therefore be written as $F(O_k) = F(x_{k1}, x_{k2}, x_{k3}, \ldots, x_{kn})$, with option $k$ represented as $O_k = (x_{k1}, x_{k2}, x_{k3}, \ldots, x_{kn})$.

As the attributes often represent different measures (e.g. weight, price), the attributes’ values are projected on a unified scale (e.g. [0, 100]) via attribute utility functions $u$. The utility function structure depends on the structure of the tree of attributes. Utility function for a single-level tree could be sum of individual attribute utilities $u_k$ e.g.:

$$F(O_k) = \sum_{j=1}^{n} u_j(x_{ij}) .$$

MCDA models often include weights to model the varying significance of different attributes. The value function for a single-level (non-hierarchical) MCDA model could then be written as the MAU (multi attribute utility) function as it is known from the MAUT method [17]:

$$F(O_k) = \text{MAU}(O_k) = \sum_{j=1}^{n} w_j u_j(x_{ij})$$

where $w_j$ describes the significance (weight) of attribute $j$.

A typical decision problem in the application of MCDA involves a static decision problem, i.e. decision makers follow a static set of goals, which are translated into a hierarchical structure of criteria and their weights, and then implemented as a utility function.

![Figure 1: Tree of attributes in a MCDA model [6]](image)

The resulting multicriteria model has a static structure, a static set of quantitative or qualitative criteria and static criteria weights. In the application of such a model, each options is evaluated by entering the value of its individual
criteria into the model, and a singular value representing the option utility is calculated. After analysis of results, the option with the highest utility value is then typically selected. The filter design tool selection problem (and engineering tool selection problems in general) however importantly differs because:

- the decision goals differ depending on the engineering problem that the tool is to be used on;
- due to variable goals, the criteria weights vary as well;
- some of the criteria can be exclusive due mandatory requirements for the engineering tool. These criteria have binary values (feature is present/not present); and
- weights have a discrete set of values: mandatory, desired, and irrelevant.

In such a problem, the evaluation of the exclusive criteria produces a set of viable options, which have to be then ranked according to the utility function calculated from the non-exclusive criteria. These specific qualities of the examined decision problem make the use of methods such as AHP [19], MAUT [17], or DEXi [6] impractical, as each use of the model requires changes to the multicriteria model as opposed to entering new options and their criteria values. Alternative methods, previously used in engineering, such as the DM or Pugh method [12] are better suited to this type of problem, and were used as a basis for the development of the solution – a rule-based ES. Adaptations of the DM system were however required in order to allow the development of a responsive, web-based ES.

B. Decision Modelling

Fig. 2 shows the original filter design tool selection problem as an influence diagram [9], [11]. We start with the engineering problem which requires the use of a filter design tool, and from which we can assemble a list of requirements. Based on the available tools (software packages), which represent the options, and our requirements we can conduct the tool selection process. The decision process requires explicit definition of the decision criteria and their selection. The results of the tool selection process is a ranked list of feasible options. In order to model the decision process shown in Fig. 2, we had to define the selection criteria. A survey of filter design tools was performed to gather the potential tool selection criteria. The relevant criteria were then selected by the engineering team and collected in a table containing the criteria types, their names, description, and possible values. The process is displayed in the influence diagram in Fig. 3. By consulting with the engineering experts and researchers which have provided the list of available options and their characteristics and defined the filter design goals, we were able to reduce the list of approximately 60 filter design tools characteristics into a set of 18 selection criteria, grouped into criteria types.

III. RESULTS: EXPERT SYSTEM DESIGN

A. Criteria and Weights

Due to problem specifics described in Chapter 2 we have decided to define a discrete, qualitative set of values for the weights: mandatory, desired (but optional), and irrelevant, that the user selects from when using the system. A mandatory value for a criteria will have the system to evaluate only the options where the feature is present. A desired value for a criteria will have the system add a number of points, depending on criteria weight, to the score of an evaluation option with the feature. Finally, irrelevant value for a criteria will have the system not evaluate the options for this criteria at all.

To shortly describe the operation of the proposed system, the evaluation of the mandatory criteria produces a set of viable options, which then have to be ranked according to the utility function calculated from the desired criteria.

As the importance of criteria depends on user goals, the criteria weights \( w_i \) are not constants, but depend on user input, and can be therefore described as \( w_i = f(\text{input}) \).

User does not input weights directly, but chooses each criterion to be either:
suitable options per criteria, and the SQL intersect operation is used to combine these sets and thus generate the composite set of suitable options.

In the foreseen use case, the user is presented with a sequence of questions aiming at filtering the set of viable filter design tools. Every question allows the user to define a feature (criteria) as either mandatory, desired, or irrelevant, and selecting the criteria priority, thus setting the criteria weight.

The criteria set as mandatory by the user are used to construct the set of viable options, i.e. tools that offer all mandatory features. To accelerate the filtering of viable options we have decided to move the filtering processing to the database side.

First the set of viable options VO is generated by producing an intersection of the sets of options O that satisfy individual mandatory criteria, written using relational algebra and set theory as:

\[
VO = \bigcap_{i=1}^{n} \prod_{\text{options} \sigma_{X_{i}=True}}(O)
\]

with VO representing the set of viable options, \(X_i\) with indexes \(i=1\ldots n\) representing the mandatory criteria specified by the user, and \(\sigma_{X_i} = True\) representing the selection operation in relational algebra.

The options \(O_i\) in the viable set VO are then evaluated and ranked according to the desired criteria using MAUT [20] equation for the utility function \(F(O_k)\):

\[
F(O_k) = \sum_{j=1}^{n} w_j u_j(x_{kj})
\]

where \(w_j = 1\) and \(u_j(x_{kj}) = \begin{cases} 1, \text{if } x_{kj} = True \\ 0, \text{if } x_{kj} = False \end{cases}\).

Consequently the viable options are to be displayed to the user in descending order according to their final score, i.e. utility, facilitating the selection of the Analogue Filter design tool in online learning system.

**C. Expert System Implementation**

ESs are usually developed using specialized ES software packages known as ES shells, e.g. DEXi [21]. As this ES is to be integrated within an Online assisted platform for Computer-aided design in communications being developed at the Faculty of Telecommunications, Technical University of Sofia, we have decided to implement it as a web based application using PHP and a MySQL relational database.

**TABLE I. DECISION MATRIX STRUCTURE**

<table>
<thead>
<tr>
<th>Criteria Type</th>
<th>Criteria Priority</th>
<th>Evaluation Criteria</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>AADE</td>
<td>1</td>
<td>ADE</td>
<td>0</td>
</tr>
<tr>
<td>LCFD</td>
<td>2</td>
<td>LCFD</td>
<td>0</td>
</tr>
<tr>
<td>FilterCAD</td>
<td>3</td>
<td>FilterCAD</td>
<td>1</td>
</tr>
<tr>
<td>FilterLab</td>
<td>1</td>
<td>FilterLab</td>
<td>1</td>
</tr>
<tr>
<td>PAC Designer</td>
<td>0</td>
<td>PAC Designer</td>
<td>0</td>
</tr>
<tr>
<td>PAC Designer</td>
<td>1</td>
<td>PAC Designer</td>
<td>1</td>
</tr>
<tr>
<td>PAC Designer</td>
<td>2</td>
<td>PAC Designer</td>
<td>0</td>
</tr>
</tbody>
</table>

The decision matrix is implemented as table within a relational database, containing the data on options with each row/entry representing a single option, and columns/attributes containing the criteria values. The implementation of the DM in a relational database simplifies decision model maintenance by avoiding code-level changes and accelerates system operation via implementation of a part of the option evaluation via database server-side execution in SQL code.

Option evaluation is conducted using set theory and relational algebra based instructions to execute a set of rules for filter design tool selection. User requirement entries/replies to questions are used to generate the sets of mandatory, or
desired, with priority levels 1 (highest), 2 (medium), and 3 (lowest).

Priority levels are converted into weight values using the equation:

\[
w_j = \begin{cases} \frac{3}{6}, & \text{if input}(j) = 1 \\ \frac{2}{6}, & \text{if input}(j) = 2 \\ \frac{1}{6}, & \text{if input}(j) = 3 \end{cases}
\]

, thus ensuring the conformity with the MCDA convention [20] of \(\sum_{j=1}^{n} w_j = 1\).

The evaluation criteria are grouped into criteria types, with the following types currently used: Filter types offered; Frequency range offered; Sharp roll-off slope possible; Cascading possible; Phase compensation offered; Ripple; Group Delay; Power consumption; Sensitivity; Noise level; Cost; Size. For example, the "Filter types offered" criteria type contains the criteria Low-pass; High-Pass; Band-Pass; Band-Stop; All-pass.

Value domains for the criteria range from Boolean (e.g. the Low-pass criteria per option can be true or false), discrete (with separate tables defining allowed criteria values, allowing the use of qualitative criteria) and quantitative (integer or real, e.g. the Cost criteria).

**B. Decision Matrix and the MAUT Utility Function**

Using the defined set of criteria and the filter design tool data we have designed a table to be used as the Decision Matrix (DM). Table 1 shows a fraction of the DM, displaying its structure. We chose not to include the whole table due to space restrictions. Rows are used to list the criteria types and individual criteria, while columns are used to indicate the presence or absence of filter design criteria types and individual criteria, while columns are used to list the rows are used to list the criteria weight.

The DM is implemented as table within a relational database server-side execution in SQL code.
The ES has the three usual [6] components:

- a knowledge base, implemented as a set of tables in MySQL database containing the data on the available filter design tools as a set of criteria and their discrete values;
- an inference engine, implemented in PHP and a MySQL relational database code containing the rules for filter design tool selection based on user input;
- a user interface with two alternative user experiences: the classic ES interface implemented as a set of web pages containing the set of questions and options for the user and an alternative display for advanced users, containing all criteria on a single screen, allowing the selection of features via checkboxes and radio buttons.

The knowledge base contains the following tables:

- the options table: a table containing the data on options with each row/entry representing a single option, and columns/attributes containing the criteria values (thus implementing the DM);
- criteria value tables: a set of tables containing discrete values for each criteria, where necessary. Quantitative and Boolean type criteria do not require criteria value tables;
- criteria weight tables: a set of tables containing the weight for each criteria. As mentioned in chapter 4.2, we have decided to set all weights to the integer value of 1 (one).

The implementation of the knowledge base in a relational database simplifies knowledge base maintenance by avoiding code-level changes and accelerates system operation via implementation of a part of the inference engine via database server-side execution in SQL code.

D. Inference Engine Operation

After determining the list of viable options, each option (tool) is given a final score by adding all its points. Consequently the suggested options are ranked and displayed to the user in descending order according to their final score. Scoring detail is displayed to the user as a part of the explanation subsystem (justifier) of the system.

First the set of viable options is generating by producing an intersection of the sets of options that satisfy individual mandatory criteria using relational algebra (see Equation 4).

The options in the viable set are then evaluated and ranked according to the desired criteria using Equation 5.

To illustrate the operation of the inference engine we present a simplified example, where the user has specified that the sought tool should offer the following options (listed criteria values should be True):

- **Mandatory:**
  - Filter types offered:
    - “Low-pass”
    - “High-pass”

- **Optional:**
  - Cascading possible
  - Cost: <100 EUR

In implementation the list of specified mandatory criteria is traversed in PHP code and a set of SQL queries producing the sets as SQL views is generated. SQL code is passed to the database management system and a dataset containing the viable options and their criteria values is returned.

The generated SQL code for the given example would be:

```
CREATE VIEW ViableOptions
AS
SELECT OptionId
FROM Options
WHERE LowPass = True
INTERSECT
SELECT OptionId
FROM Options
WHERE HighPass = True
```

The evaluation of desired criteria as per equation (8) would be then implemented as a set of rules as in the following pseudo-code example:

```
IF: UserInput(CascadingPossible) = "Desired"
THEN
  IF Option(CascadingPossible) = True
  THEN: OptionScore = OptionScore + 1
```

IV. CONCLUSION

Based on the study performed in the paper a MCDA based decision support system is to be developed to support the Analogue Filter design tool in online learning system. The approach will be extended to design in different categories of circuits and systems where a set of online CAD tools with high degree of equivalence is available. The boom of online tools led to development of portals or pools of online tools as for example Martindale’s Center [22], but in those pools, tools are neither estimated nor characterized and there is not an interactive mode proposed for users. The novelty in the approach proposed in the paper is based on the estimates and characterization passports of online tools in the online learning system which are implemented further for automated support of designers in tool selection and development of interactive mode for users.

The following step is the development of a rule-based ES to be integrated in the telecommunications engineering online learning system, and used as a decision aid selecting the appropriate filter design tool. It is to be open-ended and able to incorporate additional filter design tools and decision criteria. The planned development will include design of the data model for the knowledge base, the design...
of user interface and coding of the user interface and inference engine using web technologies (PHP, JavaScript).

REFERENCES


