Towards Web-based Decision Modeling Software based on DEX Methodology

Adem Kikaj
Jožef Stefan International Postgraduate School
Department of Knowledge Technologies, Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
adem.kikaj@ijs.si

Marko Bohanec
Department of Knowledge Technologies, Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
marko.bohanec@ijs.si

Abstract. *Decision making is a challenging human activity, which often requires support from appropriate methods and tools. One such approach is provided by multiple criteria decision analysis (MCDA), which typically proceeds by developing a multi-criteria decision model, which is in turn used for the evaluation and analysis of decision alternatives. Currently, there are many diverse MCDA methods, and some of them are supported by software. DEX (Decision EXpert) method is in our focus and it belongs to the class of qualitative hierarchical multi-criteria models. The aim is to design a new web-based architecture that would facilitate in developing and using DEX models. We do this through an investigation of already developed MCDA software and their architectures, formulating the requirements for the DEX software and developing appropriate system architecture. The first implementation of this work-in-progress development is expected towards the end of 2019.*

Keywords. Multiple criteria decision analysis software, Decision support, Web application, Software, DEX method

1 Introduction

Decision making is the process of identifying and choosing alternatives based on the values, preferences and beliefs of the decision-maker. Some decision problems are inherently difficult because of various obstacles, such as missing information, uncertainty, conflicting goals, and opposing views of multiple decision-makers. In such situations, decision making may substantially benefit from using decision-aiding methods and tools.

Multiple Criteria Decision Analysis (MCDA) (Greco et al., 2016), is an approach aimed to support decision makers faced with difficult decision problems that involve multiple and possibly conflicting criteria. MCDA is a systematic, step-wise approach, which typically begins with the identification of decision alternatives and criteria, and is followed by the development of a multi-criteria model, which is in turn used to assess and analyse the alternatives. On this basis, the alternatives can be ranked and the best one chosen.

In order to support the use of MCDA methods in practice, various multi-criteria decision-modeling software (DMS) has been developed (Mustajoki & Marttunen, 2013; Ishizaka & Nemery, 2013). DMS usually provides ways to build a decision model and analyse the results. DMS consists of different forms of computer programs and is focused on offering an environment that is designed to enable a decision-maker to structure and better understand a decision problem. Most of the DMS try to offer a user-friendly graphic user interface to ease the process of decision-making by (i) visualizing the process and the results of a decision problem and (ii) providing analytical decision tools. DMS can take different forms, such as Decision-tree, Linear-programming, Statistical, Spreadsheet-based, Rule-based, etc. (Nagel, 1991).

DEX (Decision Expert) is one of the MCDA methods (Bohanec et al., 2013; Trdin & Bohanec, 2018). It is a qualitative decision support method used to evaluate and analyse decision alternatives. Currently, DEX is implemented in a freely available software called DEXi (Bohanec, 2015). DEXi is implemented in Delphi and runs on the Windows platform. DEXi supports two basic tasks: (i) development of qualitative multi-attribute models and (ii) application of these models for the evaluation and analysis of decision options. The DEX method is also distributed/implemented partly in other computing platforms (Bohanec, 2019). DEXi has been introduced 20 years ago and it has reached the state that calls for a new implementation.

In this work, we are focused on designing a new software architecture for a DMS that integrates the DEX method. In order to provide it, we do it by investigating already implemented DMS with the aim on their implementation concepts. We address specifically the platform distribution knowing that the

latest technologies offer a variety of different functions on different platforms, which can provide novel and useful features for the decision maker and decision analyst. The new architecture is expected to support the DEX approach in substantially different projects and problem areas, such as economy, agriculture, tourism etc.

The main purpose of this paper is to investigate decision-making software that implements MCDM methods, and propose and develop a modern software architecture for DEX decision-making methodology.

The structure of this paper is as follows. The next section describes the DEX method and its application together with its implementation as a decision-making software through the years. Section 3 presents a survey of platform distribution of DMS. Section 4 describes the identified functional and non–functional requirements of the DEX2Web software, and section 5 proposes the design of the architecture that will implement the DEX method. The last section summarizes and concludes this paper.

2 DEX method and DEXi software

Decision EXpert (DEX) (Trdin & Bohanec, 2018; Bohanec, 2017; Bohanec et al., 2013) is a qualitative, hierarchical, multi-criteria decision-support methodology for the evaluation and analysis of decision alternatives. DEX decision models have a hierarchical structure, which represents a decomposition of some decision problem into smaller, less complex sub-problems.

DEX method originates in the decision-making software called *DECMAK* (Bohanec et al., 1983), which has been first implemented about 40 years ago. DECMAK was implemented in Pascal targeting RT-11 or RSX-11 operating system. Later on, it was implemented as an interactive expert system shell for the IBM PC/XT/AT/PS (Bohanec & Rajkovič, 1990).

To date, the DEX method has been used to support complex decision processes in various problem domains, including health care, project management, quality and risk assessment, environmental management, data mining and many more. There exists a database of DEX models that contains 582 models developed in 140 decision-making projects in the period 1979 – 2015 (Bohanec, 2017).

The current software DEXi is the third generation of DEX software. DEXi is implemented in Delphi and runs on Windows platforms. DEXi supports an interactive user interface to construct decision models, evaluate and analyse them.

To illustrate the DEX approach, Fig. 1 shows a DEX model for evaluation of smart phones. The Smartphone DEX model is based on attributes such as Front and Back camera, Screen Size, Screen Resolution, Fast charge feature of battery and Battery Size. The Front and Back camera attributes are aggregated to the overall attribute Camera. Screen Size and Screen Resolution are mapped to the attribute screen. Attributes Screen, Fast charge and battery Size are aggregated to the Battery attribute. The top level (root) of the DEX model is the Smartphone attribute that maps the Camera and Battery attribute, and represents the overall evaluation of a smart phone

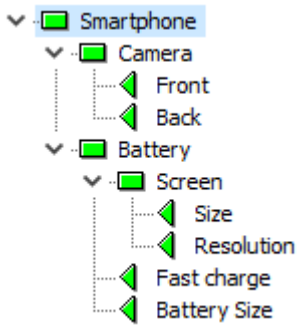

Figure 1. Tree View of the Smartphone DEX model represented in DEXi

In DEXi, decision rules (Kikaj & Bohanec, 2018) represent the mapping of subordinate attributes to the values of the aggregated attribute in terms of a decision table. Decision rules can be defined through an interactive user interface that checks their completeness and consistency at all times. Decision rules for the Smartphone DEX model, more exactly for the Smartphone aggregated attribute (root), are fully defined and shown in Table 1.

Table 1. Decision table with elementary rules of Smartphone DEX model

	Camera	*Battery*	**Smartphone**
1	unacc	short last	**unacc**
2	unacc	med last	**unacc**
3	unacc	long last	**unacc**
4	acc	short last	**unacc**
5	acc	med last	**acc**
6	acc	long last	**acc**
7	exc	short last	**unacc**
8	exc	med last	**acc**
9	exc	long last	**exc**

Attribute	**A**	**B**	**C**
Smartphone	***exc***	acc	**unacc**
Camera	***exc***	acc	***exc***
Front	***yes***	**no**	***yes***
Back	***yes***	***yes***	***yes***
Battery	***long last***	med last	**short last**
Screen	***exc***	***exc***	***exc***
Size	***big***	medium	medium
Resolution	***high***	***high***	***high***
Fast charge	***yes***	**no**	***yes***
Battery Size	***big***	med	**small**

Figure 2. Evaluation of three smart phones

Using this model, different smart phones types can be evaluated, as shown in Fig. 2. The row labelled Smartphone shows overall qualitative evaluations of phones A, B and C, which are excellent, acceptable and unacceptable, respectively. The other rows show the disaggregation of the final values according to the hierarchical structure of attributes. From there, it can be seen, for instance, that the phone C is unacceptable primarily due to the short battery life.

DEXi software includes four different analysis procedures for the evaluated alternatives (Bohanec, 2015): (1) "Plus-minus-1 analysis", (2) "Selective explanation", (3) "Compare options" and (4) "Charts".

The first version of DEXi software has been released 20 years ago. DEXi has been perpetually updated and extended with new features. However, DEXi has reached a point where it calls for a new software implementation.

There are three fundamental problems with the current implementation of DEXi. First, it has been designed to run only as a Windows-based desktop application, which was appropriate when Windows was a prevailing operating system, but now, with multiple competing operating systems as well as the web-based platforms, DEXi presents a serious limitation regarding the distribution over various platforms. Second, the software architecture of DEXi is too limited to introduce radical new features and extensions towards extended DEX, which was recently proposed by (Trdin & Bohanec, 2018). The proposal suggests new features such as numeric attributes, rich value scales (intervals, fuzzy and probability value distribution, and samples) and multiple types of utility functions. The third concern about DEXi are its current limitations regarding group and collaborative decision making, which can largely benefit from using web-based approaches.

3 Platform Distribution of Multiple Criteria Decision Making Software

Many different methods and approaches have been developed to handle multiple criteria decision analysis. There is an abundance of software that implements these approaches.

A decision-making software (DMS) is a computer program that helps a single or a group of decision makers to make choices and take decisions, most typically by ranking, prioritizing or choosing from a number of options. More specifically, a DMS covers various stages of the decision-making process, from problem exploration and structuring to ascertaining the decision maker preferences and identifying a most preferred compromise solution (Weistroffer & Li, 2016).

Problems that can be aided by multiple criteria decision making (MCDM) can be divided into two main types, (i) multiple attribute decision analysis (MADA) and (ii) multiple objective optimizations (MOO). In MADA problems, the decision-maker must choose from among a finite number of explicitly identified alternatives, characterized by multiple attributes, where these attributes define the decision criteria. In contrast, MOO deals with problems where the alternatives are only implicitly known and the decision criteria are expressed in the form of mathematical objective functions that are to be optimized.

In our work, we are going to be focused on the software that implements MADA methods and their software platform. The rationale is that DEX belongs to the MADA group and it needs a modern implementation, and the software platform is one of the first steps that need to be chosen in software development. We make a survey over already implemented MADA methods.

MADA methods are implemented in different software platforms, such as Windows, Apple Mac, Web-based, Excel Add-on, Unix/Linux, Matlab Solver, R, etc. The methods implemented as decision-making software are taken from different sources such as (INFORMS, 2019; MCDM, 2019; Bohanec, 2019; Weistroffer, 2019; Weistroffer & Li, 2016). This survey includes 90 different DMS, where 58 of them are implemented on the Windows platform, followed by 23 targeting the web as an environment to run, see Fig. 3. Even though that Windows is still the preferred software platform, the web-based solutions are gaining in importance.

Decision-making software started to be implemented in the web-based since the mid-1990s (Köksalan, et al., 2011). Since decision-support is increasingly integrated into business processes and is used for ad hoc analyses, the web-based technologies provide a suitable development platform to facilitate those integrations. The power of the web-based DMS is that it can help at retrieving, analysing, and displaying structured data from large databases, as well as facilitating the communication and decision-making in a group.

In Table 2 we show Head-to-Head comparison between desktop-based and web-based software. This comparison is done with the purpose to evaluate the environment that we are looking for to implement the DEX method.

Since we are looking for a new modern software implementation that is not platform oriented, such as DEXi that currently runs only on the Windows platform, the web-based implementations look like the most promising environment, since it is able to run in all software platforms that have a web browser installed. On the other hand, the web-based implementations are generally more demanding than desktop ones, because they are split between the client

and server side, which need to communicate with each other.

Table 2. Head-to-Head comparison of Desktop-based software and Web-based software

Desktop-based software	Web-based software
Runs locally – Personal Computer	Runs online – Server
Hard to maintain – Manual updates/upgrades	Easy to maintain – Centralized
High usage of hardware recourses	Low usage of hardware recourses
Standalone – works fast if it meets the system requirements	Working speed depends on the Internet connectivity
Data saved only locally	Data saved online and/or locally
Single user – Does not support group work	Multiple users – Supports group work
Access only in installed machines	Access from any computing device that has a web browser
Can have very low-security issues	Security issues can be very high

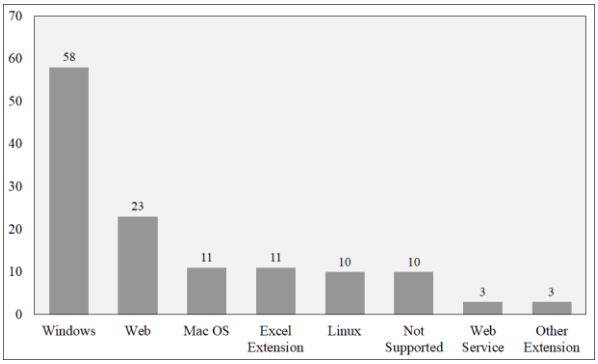

Figure 3. Platform Distribution of the Multiple Attribute Decision Making Software

4 Software Requirements for DEX2Web

The forthcoming implementation of the DEX method, which we call DEX2Web, will provide multiple functions for its users. In this section, we provide the identified functional and non-functional requirements (Sommerville, 2011) for the software. The functional requirements target the type of User requirements. The User requirements are statements written in natural language that explain the services and operational constraints of the system. On the other hand, the non-functional requirements target the type of Product requirements. The Product requirements specify the system behaviour such as: execution speed, memory required, reliability, portability, usability, etc.

4.1 Functional Requirements

- **User registration** – This function will allow the users of DEX2Web to register/create an account through a web browser. There will generally be two types of users: decision makers and decision analysts.
- **User login** – This function will allow the user to login after they provide the credentials.
- **User account editor** – This function will allow the user to edit their account information such as: name, surname, email, phone number, username, and password.
- **Project Editor** – This function will allow the registered users to create, update, delete, import, export, and share with other users a DEX project. A DEX project will consist of several DEX models together with associated data about decision alternatives. Additionally, a project will contain other components, such as reports and charts.
- **Model Editor** – This function will allow the user to create, modify, delete, import, export and share with other users a DEX model. A DEX model will consist of hierarchically structured attributes, their value scales and aggregation functions.
- **DEX Model - Attribute** – This function will allow the decision maker to modify – add, update, delete and link an attribute of a DEX model. Attributes will form a tree or hierarchy.
- **DEX Model - Scales –** This function will allow the decision maker to modify – add/declare, update, and delete scales of aggregated attributes of a DEX model. In perspective, both qualitative (as in DEXi) and quantitative (as proposed in extended DEX) scales will be supported.
- **DEX Model - Functions** – This function will allow the decision maker to modify/define aggregation functions in DEX model. In addition to functions represented by decision tables (current DEXi), other forms of functions are sought for, particularly those based on weights to be used for the aggregation of numeric attributes.
- **DEX Model - Options –** This function will allow users to add, update, and delete decision alternatives of an already built DEX model. In addition to single qualitative values, as currently in DEXi, other forms of input data will be supported: value intervals, probability and fuzzy distributions, and samples.
- **DEX Model - Evaluation** – This function will allow the users to evaluate already declared decision alternatives of a DEX Model. Both qualitative and quantitative evaluation will be supported.
- **DEX Model - Analysis** – This function will allow creating, and export/print analysis as: "what-if", "plus-minus-1", "Selective explanation", and "Compare alternatives".

- **DEX Model - Reporting** – This function will allow creating a graphical and textual presentation of DEX models, alternatives and evaluations results.
- **Interactive UI** – This function will allow advanced users to have a different representation of model attributes, scales, and functions.
- **Settings** – This function will allow each user to save different settings such as: report style, UI style, and saving models (local or online).
- **Search** – This function will allow the users to search anytime inside their account, their DEX Projects, and DEX Models.
- **User types** – This function will specify inside a DEX Model one of two user types: (i) Decision Maker or (ii) Decision Analyst.
- **Decision Maker –** This user type will have the ability to use the functions specified above: *DEX Model – Attribute, DEX Model – Scales, and DEX Model – Functions.*
- **Decision Analyst –** This function will allow users of type Decision Analyst within a DEX Model to create polls for voting online in real time for new changes in a decision model and share the results with other users that are part of that DEX Project.

4.1 Non-Functional Requirements of DEX2Web

- The user interface shall be implemented using HTML5 and CSS.
- The framework used to add style to the HTML5 is the latest Bootstrap Framework for UI.
- For each new style of the user interface, the Bootstrap Framework needs to be recompiled including that new style.
- Java EE 8 is the technology used to integrate *extended DEX* library and develop the business logic of the DEX2Web software.
- The server needs to be secured using network restriction when needed to access for new changes.
- The loading time of a static HTML page should be less than 250 ms.
- The code in all architecture layers needs to be commented on properly.
- For scalability purposes, the whole system shall be deployed in the Server using Docker technology.
- AJAX calls are allowed for small requests from client-side.
- All processes that have to do with DEX library needs to be finished in the Server-side and then to be transferred in the user-side.
- Java Logging API is needed for all processes.
- Database processes need to be in the form of transactions.
- Session timeout is set to 10 minutes.
- The HTTPS protocol is mandatory.
- The validation of importing DEX files is in two sides, user and server side.
- Nothing in Database is permanently deleted.
- Database backup needs to be done automatically every day by running server-side scripts.
- Database backup needs to be located on a different Virtual Private Server.

5 DEX2Web Software Architecture

The aim of this work is to design and implement a new-generation web-based software for DEX decision modeling using the *extended DEX* library. The idea is to employ a multi-layer architecture that will keep the web-based software updated, centralized and distributed for different software platforms that have a web browser.

DEX2Web is going to implement a multitier architecture, more specifically a client-server architecture (Rosen & Shklar, 2009). A client/server is a two-layer architecture where we have to do with a client that represents a computer and it has a software that runs on it and with a server that serves to the client specific information or services based on the client requests usually through the intranet. In a web-based software we have to do usually with a three or *n*-layered architecture that usually runs through the internet.

The communication between the server and client is done based on the request of the client to the server. The web client will communicate with application server by using HTTP (HyperText Transfer Protocol) which is based on TCP (Transmission Control Protocol) (Rosen & Shklar, 2009). To support transmitting sensitive data, we will employ the HTTPS protocol, the secure version of the HTTP protocol (Rosen & Shklar, 2009).

DEX2Web is going to implement a 4-layer architecture (Fig. 4) consisting of:

- An object-oriented *extended DEX library* that provides means and methods for the creation and modification of DEX models, and evaluation and analysis of decision alternatives.
- *DEX services*: Accessing the DEX library using the principles of Software as a Service (SaaS). DEX services provide a web-based connection between the DEX2Web user interface and the functionality of the DEX library.
- *Database*: A relational database used to store data in order to facilitate effective information sharing between users and library to provide full functionality of the DEX2Web.
- *DEX2Web user interface*: Web pages for handling the dialog with the user (decision maker) and invoking the necessary DEX services.

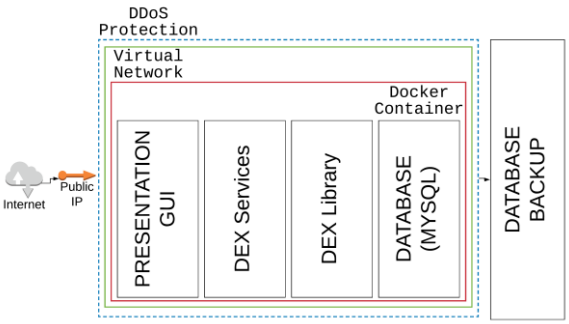

Figure 4. 4 - Tier Architecture of DEX2Web

From these components, the former three are going to be executed in the server-side (Object-oriented DEX Library, DEX services, and Database), and the latter (DEX2Web user interface) on the client side.

The *extended DEX* library is more advanced comparing to the last implementation of DEX method in DEXi software. This library is being implemented in Elements platform using Oxygene programming language *(The Oxygene Language, 2019)*. The *extended DEX* library will serve to the web-based software all the needed functions/tools to develop and manipulate DEX Models, as well as to analyse them. The communication between the library and the web-based software itself are going to be handled by using the Model-View-ViewModel (MVVM) pattern.

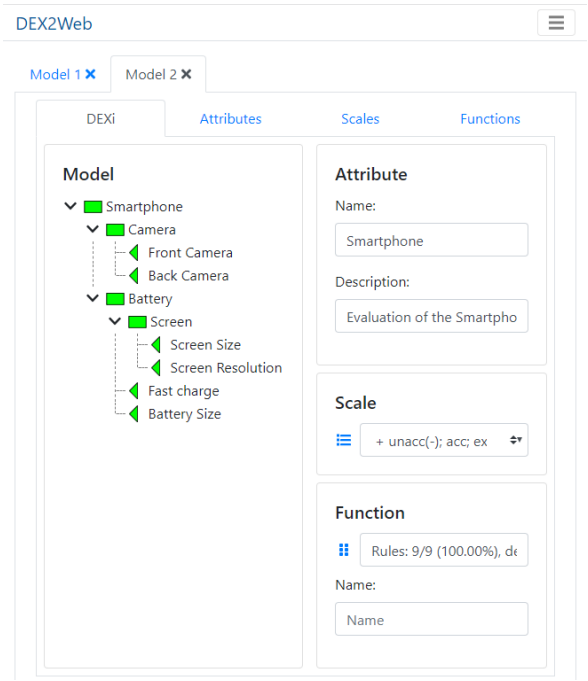

Figure 5. Main View of the DEX2Web – Smartphone Model

DEX Services component will accept the requests/inputs from the users (user-side) such as Decision Analyst and Decision Makers. This component's task is to connect the web technologies with the library and to process and validate the inputs through secure instructions and steps using authorization protocols and from there to provide back to the users the feedback from the *extended DEX* library component.

The *DEX2Web user interface* components rely on HTML5 pages where their content is updated dynamically based on the DEX Services output. This component is rendered on the user-side and it is what the users will be interacting with. The component itself includes also technologies that will be transferred to the user which their task is to validate user inputs. Fig. 5 shows an example where the Smartphone Model is rendered on the user-side after it has been requested and it is ready for interaction.

The *Database* component will store and manipulate all the data of the software. The four functions used in the database are: creating, reading, updating and deleting data. An outline of the information to be stored in database is shown in Fig. 6.

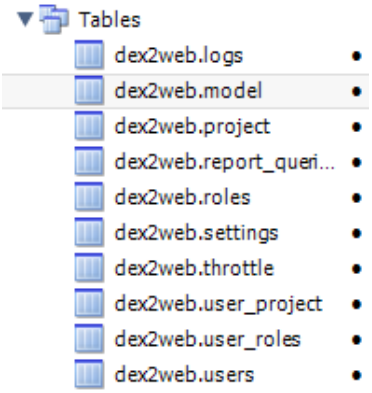

Figure 6. Database Schema of DEX2Web

For the DEX2Web database, we shall use MySQL (Mysql, 2019), the world's most popular open source database management system with enhanced attributes such as reliability, observability, manageability, security, and performance (Why MySQL?, 2019).

In our development process, we choose Java technology for developing the web-based software, more exactly the Java EE platform (Java™ EE, 2019).

The application server used in this development is GlassFish which is a complete open source Java EE application that supports dynamic and scalable HTML5 applications.

Conclusion

The purpose of this work was to pinpoint the importance of implementing decision-modeling software and to propose a novel web-based architecture for implementing the extended DEX method.

From the technological viewpoint, based on an overview of 90 MCDM software implementations, it is clear that traditional desktop applications are giving

way to modern web-based architectures. These offer many advantages, particularly an easy access and no need for perpetual updating for the user, and platform independence for the developer. There are also many possible threats, such as more difficult development and integration of multiple components and multiple technologies, user-identity and security issues, etc. But the direction is clear and the future DEX implementation is going to be web-based.

From the functional viewpoint, based on decades of experience with previous generations of DEX software, the new implementation is necessary to facilitate adding new "extended" features. These include using and combining qualitative and quantitative attributes, extended values (intervals, fuzzy and probability value distribution, and samples), multiple types of utility functions, and support for group decision-making. All these are becoming possible thorough an implementation of the novel extended DEX library.

At the time of writing of this paper, the extended DEX library has been almost completed, and the remaining three components of DEX2Web are being developed. A first prototype version of the software is expected to become publicly available towards the end of 2019. Overall, the idea is to first support the functions that are available in the current DEXi software, and then to gradually add additional extended features. With this, DEX2Web is expected to become a general decision-modeling platform facilitating both qualitative and quantitative MCDM methods.

Acknowledgments

The Public Scholarship, Development, Disability and Maintenance Fund of the Republic of Slovenia (Contract no. 11011-44/2017-14) financially supported this research.

References

Bohanec, M. (2015). DEXi: Program for Multi-Attribute Decision Making User's Manual. *Ljubljana, Slovenia: Institut Jozef Stefan*.

Bohanec, M. (2017). Multi-criteria DEX models: An overview and analysis. *SOR-2017: 14th International Symposium on Operational Research in Slovenia*, 155-160.

Bohanec, M. (2019, June 6). *Decision Support Resources*. Retrieved from IJS Decision Support Resources: http://kt.ijs.si/MarkoBohanec/dss.html

Bohanec, M. (2019, June 24). *DEXi*. Retrieved from http://kt.ijs.si/MarkoBohanec/dexi.html

Bohanec, M., & Rajkovič, V. (1990). DEX: An expert system shell for decision support. *Sistemica*, 145-157.

Bohanec, M., Bratko, I., & Rajkovič, V. (1983). An expert system for decision making. Processes and. *North-Holland*, 235-248.

Bohanec, M., Žnidaršič, M., Rajkovič, V., Bratko, I., & Zupan, B. (2013). DEX methodology: three decades of qualitative multi-attribute modeling. *Informatica*, 37(1).

Greco, S., Ehrgott, M., & Figueira, J. R. (2016). *Multiple Criteria Decision Analysis: State of the Art Surveys*. New York: Springer-Verlag.

INFORMS. (2019, June 9). Retrieved from The Institute for Operations Research and the Management Sciences: https://www.informs.org/ORMS-Today/OR-MS-Today-Software-Surveys/Decision-Analysis-Software-Survey/Page-1

Ishizaka, A., & Nemery, P. (2013). *Multi-criteria decision analysis: methods and software*. John Wiley & Sons.

Java™ EE. (2019, June 24). Retrieved from Oracle: https://www.oracle.com/technetwork/java/javaee/overview/index.html

Kikaj, A., & Bohanec, M. (2018). Complex Decision Rules in DEX Methodology: jRule Algorithm and Performance Analysis. *Proceedings of the 21th International Conference Information Society IS 2018, Volume A*, 17-20.

Köksalan, M. M., Wallenius, J., & Zionts, S. (2011). *Multiple criteria decision making: from early history to the 21st century*. World Scientific. Retrieved from (2011). . World Scientific.

MCDM. (2019, June 6). Retrieved from Software Related to MCDM | Multiple Criteria Decision Making: https://www.mcdmsociety.org/content/software-related-mcdm

Mustajoki, J., & Marttunen, M. (2013). *Comparison of Multi-Criteria Decision Analytical Software-Searching for ideas for developing a new EIA-specific multi-criteria software*. Finnish Environment Institute.

Mysql. (2019, June 18). Retrieved from https://www.mysql.com/

Nagel, S. S. (1991). *Decision-aiding software: Skills, obstacles and applications*. Springer.

Rosen, L., & Shklar, L. (2009). *Web Application Architecture: Principles, Protocol and Practices*. John Wiley & Sons Ltd.: West Sussex, England.

Sommerville, I. (2011). *Software engineering 9th Edition*. ISBN-10, 137035152.

The Oxygene Language. (2019, June 15). Retrieved from https://www.elementscompiler.com/elements/oxygene/

Trdin, N., & Bohanec, M. (2018). Extending the multi-criteria decision making method DEX with numeric attributes, value distributions and relational models. *Central European Journal of Operations Research*, 1--41.

Weistroffer, D. R. (2019, June 9). *Multiple Criteria Decision Support Software*. Retrieved from Multiple Criteria Decision Support Software: http://www.isy.vcu.edu/~hweistro/mcdmchapter.htm

Weistroffer, H. R., & Li, Y. (2016). Multiple criteria decision analysis software. In *Multiple Criteria Decision Analysis* (pp. 1301-1341). Springer.

Why MySQL? (2019, June 18). Retrieved from https://www.mysql.com/why-mysql/