Exploring MCDA methods with DecSpace

João Amador, INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal, joao.amador@ist.utl.pt

Abstract

Multiple Criteria Decision Aiding (MCDA) is a domain of Operations Research and Management Science mainly devoted to the development of methods, techniques and tools aiming to help decision makers in complex decision processes. This paper describes DecSpace, a framework to explore a wide range of methods of that kind. DecSpace is a web-based framework, making it possible to explore solutions for problems involving one or more methods. Persistent workspaces also make it possible to reuse solutions. In this sense, DecSpace is intended for use in teaching and researching in MCDA methods, as well for professional use as a Decision Support System for engineering and management scenarios where decisions based on those methods are to be considered. DecSpace has an open architecture, making it possible to use methods implemented locally, fully integrated as part of the local application, or methods available from remote servers, if exposed as web services.

Keywords: Multiple Criteria Decision Aiding; Decision Support Systems; Web application; User experience.

1. INTRODUCTION

Multiple Criteria Decision Aiding (MCDA) is a domain of Operations Research and Management Science mainly devoted to the development of methods, techniques and tools aiming to help decision makers in decision processes.

The main purpose of MCDA methods is to support a decision maker during the decision-making process, while taking personal preferences into account. The use of this kind of methods can be challenging for those that are not experts in MCDA.

The number of MCDA methods and publications are increasing, and with it so does the software available, including spreadsheets containing method computations, web or smartphone applications, contributing to the increase use of these methods amongst researchers and professionals.

DecSpace is a framework for the use of MCDA methods by non-specialists, with the clear objective of lowering the complexity of using those methods. The main goals are the ability to reuse existing open reference implementations that already exist for some methods (which is the case for many of them, therefore they do not need to be re-implemented), the provision of an interface that welcomes users from diverse backgrounds and with different levels of experience with MCDA and to provide a web-based framework, making the best use of the present state of the art in the related technology (Barbosa, 2017).

This extended abstract is organized as follows. Section 2 introduces a brief overview of MCDA. Section 3 presents some existing decision support tools. Section 4 is devoted to present DecSpace. Section 5 introduces the DecSpace domain model. Section 6 provides the details of the implementation. Section 7 presents an overview of the conducted evaluation. Section 8 presents some concluding remarks.
2. **Overview of MCDA**

MCDA is useful for facilitating in decision situations that involve several criteria, while the preferences of the decision makers are taken into account. Adopting an MCDA approach is pertinent in several domains, such as healthcare (Marsh, Goetghebeur, Thokala & Baltussen, 2017), environment (Huang, Keisler, & Linkov, 2011), finance (Doumpos & Zopounidis, 2014), energy management (Mardani et al., 2017), among many others (see, for example, Guarnieri, 2015).

Generally, a set of objects, actions or alternatives (hereinafter called actions) are assessed according to a multitude of characteristics, also called attributes, considered relevant for the decision situation at hand. In the case of the value sets of these attributes are preferentially ordered, they are called criteria (Roy, 1996). For instance, one can face decision situations dealing with the choice among actions (e.g., choosing a car, see Bouyssou et al., 2000, chapter 6), or the rank of actions (e.g., ranking universities, see Corrente et al., 2017), or the classification of them into predefined categories (e.g., classifying countries in terms of governance performance and efficiency, see Costa, Figueira, Vieira & Vieira, 2017).

Decision making is present in our lives, from simple decisions to complex decision situations, usually involving conflicting criteria. In particular, for organizations, public or private, in several sectors, decision making is crucial for their business. Thus, managers (or stakeholders) need the support offered by MCDA methods to make effective decisions. Indeed, an MCDA approach can be suitable even for dealing with decisions related to complex systems presented, for example, in industrial processes, public policies, and supply chain management.

An MCDA approach allows to structure and have a better understanding of the decision situation at hand and handle it in a logical and systematic way. In addition, an interaction with the stakeholders enables a transparent process and a deeper knowledge of the whole decision situation. Besides that, having tools to support the decision aiding process is an added value. In practice, the MCDA methods need to be supported by multi-criteria software tools or decision support systems (DSS). Indeed, having DSS is pertinent to facilitate the collection of the data, all computations and the analysis of the results (including sensitivity and robustness analyses). DSS are relevant to all actors that participate in the decision-making process, in the sense that DSS can facilitate the application of the methods and techniques, even to those that do not have experience a priori with MCDA, providing an easy way of modeling the problem, and visualize the data, processes and results (Liu & Stewart, 2004). Besides the applicability and relevance of MCDA for several domains, this reinforces the significance of the development of methods and their appropriate implementation.

3. **Overview of MCDA Tools**

The application of MCDA methods needs to be supported by adequate software. Numerous software solutions have been implemented to support the application of MCDA methods in real-world decision situations, but very few offer a great solution for visualization purposes.

To really understand the problem, we need to comprehend the visualization tools identified, using the following state-of-the-art web technologies, utilized to achieve the promise of interactive browser-based system visualization application. Each tool is examined from a visualization representation, meaning that we are looking for the best evaluation of the objectives, architecture, technology applied and main application functionalities. This analysis will help to figure out what are the best characteristics of these software programs and what are the main aspects that they are lacking, which this new framework will try to overcome, in order to make a very inclusive implementation.

**Diviz** is an accessible software tool for building, executing and sharing complex workflows formed by MCDA methods, developed by the Decision Deck Project (Meyer & Bigaret, 2012). One of the main features is that it supports the construction of academic MCDA methods, by combining several elementary calculation elements, offering the tools to consolidate new decision methods, in every implemented language, without having to rewrite them (Meyer & Bigaret, 2012). This design makes
possible to efficiently integrate additional methods without great struggle. The design of the MCDA workflows is performed by an intuitive graphical user interface, allowing the user to execute it in order to obtain various possible outputs of the algorithms. These calculations are implemented on high performance computing servers by using the XML Multiple Criteria Decision Aiding (XMCDA) web-services. After the execution of the workflow, the outputs of each component can be viewed and analyzed by the user, allowing for a complete concept of the algorithm, while the parameters can be changed.

_D-Sight_ provides a selected amount of Strategic Solutions, with a specialization in the areas of MCDA and user-friendly software development (Hayez, De Smet, & Bonney, 2011). This system provides responsiveness analyses to easily visualize and measure the impact of changing factors on projects. By developing a unique mathematically validated methodology for multi-criteria analysis and bound it in a state-of-the-art software, it is possible to see all this relevant data under the model of a graph or chart or even automatically see every likely scenario, making the best decisions based on data and thorough analysis.

_Expert Choice_ is a real-time decision-making software that consolidates cutting-edge technology with time-tested mathematics to provide a well-informed decision (Barfod, 2014). The objective is to promote a collaborative, straightforward, and accurate approach to complex decision-making, changing complex and mathematically incorrect decision processes into a repeatable, understandable, and valid process. With an easy-to-use software and minimal training, leverages decision-making practices that are established to deliver more optimal decisions.

_Visual PROMETHEE_ is an MCDA software, that offers an intuitive and comprehensive interface, by providing all the data available and editable, it offers a distinctive view of color and shapes to visualize groups of action or criteria, criteria statistics, and smart assistants are available to help the user to assess preferences and priorities, and multiple scenarios can be defined to reflect the points of view of several stakeholders or different hypotheses (Mareschal, 2013).

_MACBETH_ is a non-numerical interactive approach that demands only qualitative decisions about “attractiveness differences” to support a decision-maker or a combination of decision-makers, evaluate and quantify the relative attractiveness of the alternatives (Bana e Costa, De Corte, & Vansnick, 2003). This questioning approach where the elements are compared in pairs, requires only a qualitative preference judgment where their consistency is automatically verified, generating a numerical scale that is representative of the decision maker’s decisions.

After an extensive analysis of the three tools taken into account, for being the most solid and better possibilities in the market for presenting the best usability and interface, many of them have very interesting features but none offers the complete experience, it is important to point that for some tools a license is required (some offer a demo), that is expensive for the average user. The analyzed features are presented in Table 1 with the conclusions.

<table>
<thead>
<tr>
<th>Feature</th>
<th>D-Sight</th>
<th>Expert Choice</th>
<th>Visual PROMETHEE</th>
<th>MACBETH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Open-source</td>
<td>y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 3D visualization techniques</td>
<td></td>
<td></td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>3 Data files import and export</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>4 Results comparison</td>
<td>y</td>
<td></td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>5 Level of expertise required</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 1 – Studied tools features
The application of MCDA framework can be challenging for users who are not experts in MCDA methods. There are not many frameworks that focus on usability or user experience making it very difficult to use. Nonetheless its application is becoming more important every day, so the need to create a new application surged.

4. DECSPACE OVERVIEW

DecSpace is a framework that offers MCDA methods for the user to solve problems of different kinds within just one application that is compatible with all web browsers. In this section, the new prototype is analyzed from different perspectives, taking into consideration the architecture, features and technologies.

DecSpace provides the following use cases as illustrated in Figure 1:

- **Types of Users**: There are four types of users with different permissions: the developer has the objective of adding new methods; the administrator manages all registered users and projects; finally, registered users and anonymous users, the difference between them is that registered users have an account and their work is persistent;

- **User login and registration**: The first step to use the application is to provide a username, an email, and a password, to access the user personal area, it is possible to enter as a guest, but some features are removed from the experience, as a logged user it is possible to reset your password or log out;

- **My Projects**: This area allows the user to manage projects, to create a project, and to choose if the project is public or private, the following fields identify each project: Name, Creation Date, Last Update and Privacy Setting, for each project, open, duplicate and delete features are available;

- **Public Projects**: This area contains all public projects that are shared by users, the following fields identify the projects: Project Name, User name, Creation Date and Last Update, for each project, a duplication and open feature is available, the purpose of this area is to provide complete projects to all users;

- **Method Catalog**: All the information for the methods created are in this area, they are divided into two categories, local methods, created by the developers of the framework, and remote methods, through SOAP requests between the diviz server, it is possible to reuse all their different methods. Each method presents some relevant information, a practical example for guidance and a step-by-step tutorial, for the most inexperience users;

- **Workspace**: This is where all the technical work happens. The objective is to have a simple working area that executes user’s choice methods as fast and responsive as possible, as far as features go, it should be possible to import and export a CSV, XMCDA or JSON file or use locally stored workflows, these can be saved, refreshed, executed and deleted. Any method on the catalog is ready to be used, they can be dragged to the workspace and connected between themselves.

The proposed architecture is structured in three main tiers as presented in Figure 2. The objective is to confine most of the complexity to a single tier and restrain the interface details to another tier, which allows to focus on the usability and user interface:

- **Client tier**: The client tier consists of an interaction with the application tier to send user requests and after these requests are processed a response is returned to the client, which allows the user to perform the most various tasks the system has to offer. In terms of infrastructure, the interaction with the web server is done with simple HTTP requests;

- **Application tier**: The application tier defines the great complexity of the system, where most of the computational activity is performed. It acts as the server for client requests, by receiving and processing user requests sent by the client tier, and also carrying out the
connections to the data tier. The location is a web server that executes the workflows and several MCDA methods;

- **Data tier**: The data tier is an entity that contains all the information of the application. It communicates with the application tier by receiving, processing and replying to the data requests. It is formed by a database and communicates with the application tier with the aid of the MongoJS library.

DecSpace implementation uses the MEVN Stack approach, using the combination of the following programming languages and libraries that synergize well together, which simplifies and accelerates the development of web applications:

- **MongoDB**: It is an open-source cross-platform document database that provides high performance, availability and automatic scaling. MongoDB supports dynamic schema design, allowing the documents in a collection to have different fields and structures;
- **Node.js**: It is an open source development platform for executing event-driven JavaScript runtime environment, specially designed for building scalable network applications;
- **Express.js**: It is a minimal and flexible Node.js framework that provides a robust set of features for web and mobile applications, to help organize your web application into a model view controller (MVC) architecture on the server side;
- **Vue.js**: A progressive framework for building user interfaces, Vue is designed from the ground up to be incrementally adoptable.

Aside the MEVN stack technology. Hypertext markup language (HTML) is used for designing and displaying the basic structure of the web pages; Cascading style sheets (CSS) for the visual style of the web pages; JavaScript for advanced handling on the front-end side of the application.

![Use Cases Diagram](image-url)

**Figure 1 – Use Cases Diagram**
The most consistent concepts and the relations between them should be described to adequately understand the necessity for this application, as represented in Figure 3. This application starts with users, which are identified by its unique email address and must have an associated password, username, privacy setting, date of registering, and date of the last login are all stored. This information is used to create an account, if all the necessary data are correctly introduced the account is created, if not an error message will help the user correct the error. To access projects and information about personal user account it is necessary to be logged in, every registered user can own multiple projects and read multiple public projects. Each project is identified by its id, project name, owner, date of creation and last update date, it should provide the create, open, duplicate and delete features. A single project may incorporate several workflows, which are identified by their id as well. They also contain the date in which they were saved, and a save name entered by the user. A workflow consists of various modules that can be connected between them. A connection is identified by its name and has both a source module and a target module. A module has two types: data, input module or output module, or a method module, which are identified by the attribute method parameters, that differs for each respective method, and name, as of the last version, there are six different types of local method modules that correspond to the available methods (Additive Aggregation, CAT-SD – Categorization by Similarity-Dissimilarity, Inquiry, Order By, Sort, and SRF), there is also a remote method that corresponds to all available methods in diviz, the contents of this remote method are performed in XMCDA. The method catalog contains all the methods information for the users to consult the preferred ones.
6. **DECLSPACE IMPLEMENTATION**

The various aspects of the implementation of DeclSpace are portrayed here, the solution implementation is divided in three parts front-end, back-end and database.

The front-end of the application is comprised of components, each component represents a website page or feature, each file is separated in different concerns, HTML, JavaScript, and CSS. The main goal is to attract different users, to an innovative framework that is responsive, very quick, secure, that adapts to all devices like smartphones or tablets, and as an innovative yet simple interface. The concept of this final design decisions went to a lot of different ideas, different prototypes and decisions, and lastly through user tests and suggestions, concluding in the application provided.

DeclSpace introduces the application presenting a homepage with diverse features giving access to the most basic features as shown in Figure 4; “FAQ”, which presents the most asked questions, regarding the functioning and features of DeclSpace, and respective answers; “Method Catalog” offers information about all the implemented methods, along with tutorials and examples as displayed in Figure 5; for each registered user a “My Projects”, “Public Projects”, and “Settings” are available to control all project and user functionalities, an example for the “My Projects” area as is depicted in Figure 6. The main workspace is where all the workflows are created, is quite intuitive with a drag and drop feature and a color scheme of the data input and output making it very simple, modules should be able to be connected between themselves or to different data files, an important addition is the possibility to analyze results from tables and graphs as shown in Figure 7.

Device Mode is an innovative feature, tested in Chrome DevTools to enable streamlined device display emulation of various devices that included iPhone, Samsung Galaxy, and iPad. The goal is to provide the best experience possible in each type of device, to make a pleasant activity of using DeclSpace.

The back-end application serves indirectly in support of the front-end services, usually by being closer to the required resource or having the capability to communicate with the required resource. In this case, two models were designed “user” and “project” interacting with both the front-end and the database.
The solution devised for the database was setting the development version in an open cloud service without worrying about performance and looking for a simple and effective way to control and manage data. For the deployment version, the database was changed to a local database to present better performance and fewer issues with services.

Figure 4 – DecSpace v2.0 Homepage

Figure 5 – DecSpace v2.0 Method Catalog
A system evaluation plan has the objective to test the framework on all crucial features, our desire with this evaluation is to understand everything that feels wrong or confusing for all users, also correcting bugs and errors that emerge. The system evaluation guide was designed with the objective of explaining the various steps of the evaluation, so that valid results could be achieved and that the test sessions would occur as active as possible. After each step of the guide, the user is encouraged to discuss the best and worst mechanics, all the steps are recorded and the metrics collected. After the completion of the guide, it is mandatory to answer the survey, composed of thirteen questions about usability, aesthetics, recommendations.

The main adjustments or modifications addressed by the users that tested the framework are: Change the name of “User Preferences” to ”Method Catalog Preferences”, ”My Projects Preferences” and
"Public Projects Preferences"; More information in error messages; Change the execute icon; Change "Edit Privacy" to "Change Privacy"; In tutorial change the placement and color of the "Finish" and "Back" buttons; Apply some outline in "OK" buttons; Change methods avatar; Open project by clicking on the line of the project; Graphics interface redesigned; Fixes in icons placements in all the framework.

8. Conclusions

The purpose of MCDA is to provide support to decision makers during decision processes with methods and techniques, as well as frameworks for computational support. Handling these frameworks can be challenging for users who are not experts in MCDA methods. For that reason, requirements related to usability and user-friendly issues need to be taken into account in the development of these frameworks. In this paper, we proposed a new approach to the MCDA phenomenon with a framework that employs forefront technology, well-studied interaction techniques. The fundamental objective is to support a decision maker to structure and to better understand the decision problem at hand.

DecSpace is a web-based framework that provides MCDA methods and allows to create projects with multiple workflows using one or more of such methods. Indeed, an increasing number of MCDA methods are available, implemented locally or remotely exposed by diviz server and as web services. We show that DecSpace is intended to be a user-friendly solution for supporting the decision processes in different scenarios, for both non-expert users and MCDA expert users. We presented the main features of DecSpace and highlighted that is focused on the usability and user interface. Future work relies on improving usability and user experience issues.

Bibliography