diviz: an MCDA workflow design, execution and sharing tool

Sébastien Bigaret, Patrick Meyer
Institut Télécom, Télécom Bretagne
UMR CNRS 3192 Lab-STICC
Technopôle Brest-Iroise CS 83818
F-29238 Brest Cedex 3
Université européenne de Bretagne
{sebastien.bigaret,patrick.meyer}@telecom-bretagne.eu

ABSTRACT. The diviz software is a tool for designing, executing and sharing Multiple Criteria Decision Aid (MCDA) techniques. The target user community of diviz reaches from researchers who require to construct algorithmic workflows of elementary MCDA components to teachers who need to present MCDA methods and let their students experiment them. Besides, one of the principal features of diviz is to ease the comparison of results results issued from different methods on the same decision problem. This allows among others to check the robustness of the output of a decision aid process with respect to the choice of the decision aid technique.

KEYWORDS. algorithmic MCDA workflows, open-source software, methods comparison.

1. INTRODUCTION

Research activities in and around the field of Multiple Criteria Decision Aid have developed quite rapidly over the past years, and they have resulted in various streams of thought and methodological formulations for the resolution of complex decision problems. In particular, many so-called MCDA *methods* have been proposed in the literature and are very often available as software programs.

Unfortunately, at least four major difficulties arise when it comes to using these procedures in practice:

1. Different techniques are generally implemented in separate software products, with heterogeneous user interfaces;
2. Testing multiple MCDA algorithms on one problem instance is not easy, because of the various input data formats required by the software applications;
3. A lot of MCDA algorithms which are presented and published in scientific articles are not easily available and consequently often only used by their authors;
4. Several MCDA software products are not free (neither from a financial, nor from an open-source point of view), which can be considered as a weakness for their large dissemination.

In other scientific research fields, as, e.g., statistics or data mining, there exist software platforms which allow to easily compare different analysis methods and to test them on a given dataset inside a common framework. Among the most famous ones, one can cite platforms such as the GNU R statistical system (R Development Core Team, 2009) or the Weka suite of machine learning software (Hall et al., 2009). Both of these suites are open-source and OS independent, which has certainly contributed to their large dissemination and acceptance among many researchers and users.

The diviz software is one of the initiatives of the Decision Deck[1] project and is developed at the LUSSI department of Télécom Bretagne in France. Its essential purpose is to overcome the main difficulties of the current software situation in MCDA research mentioned above. diviz is an open-source software which allows to design complex workflows[2] of MCDA algorithms in a very intuitive way and to execute them. One of the key characteristics of these executions is that they are performed via web-services on distant servers.

In this short article, we present diviz, its properties and its latest developments.

2. DESCRIPTION AND PROPERTIES OF DIVIZ

The diviz tool is an easy to use software to build, execute and share complex workflows of MCDA algorithms. In the literature, such workflows are often called *methods* (consider, e.g., the ELECTRE method (Roy, 1968), the UTA method (Jacquet-Lagrèze and Siskos, 1982), etc). diviz enables to conveniently combine programs implementing MCDA

[1] http://www.decision-deck.org

[2] By workflow we mean a sequence of connected algorithmic steps.

algorithms, which can originate from various methodological schools, researchers and programmers.

In this section we present the main features of diviz and show how it allows to rebuild classical MCDA *methods*, to develop new ones by combining various elementary calculation components[3], and how it can be used as a research and dissemination tool.

Workflow design

The design of the MCDA workflows is performed via an intuitive graphical user interface, where each algorithm is represented by a box which can be linked to data files or supplementary calculation elements by connectors. Thus, the design of complex algorithmic workflows does not require any programming skills, but only necessitates to understand the functioning of each calculation module.

The inputs and outputs of these elementary components can be manifold and are *typed*. To illustrate this, consider the following example.

> ***Example**: diviz allows to use a component called weightedSum. This element calculates the weighted sum of alternatives' profiles with respect to a set of weights associated with a list of criteria. Consequently, weightedSum requires four inputs: the description of the criteria (at least their ids), the description of the alternatives (at least their ids), the performance table containing the numerical evaluation of each alternative on each of the criteria and the set of numerical weights associated with the criteria. The main output of this component is a list of alternatives' ids associated with the weighted sum of their evaluations. Note that we present later further details on the data typing.*

To construct a new MCDA workflow, the user chooses one or more modules from a list of available calculation elements which he can drag and drop in a dedicated workspace (see Figure 1 below). Then he adds data files to the workspace and connects them appropriately to the inputs of the elements. Finally he connects the inputs and outputs of the components by connectors to define the structure of the workflow.

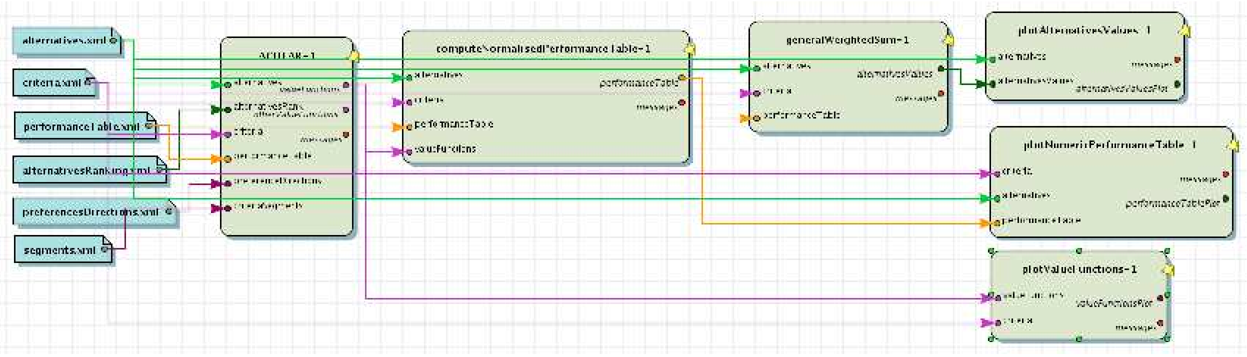

Figure 1: A typical MCDA workflow in diviz with input data files on the left

Execution and results

Once the design of the MCDA workflow is finished, it is possible to execute it in order to obtain the possibly multiple outputs of the algorithms. In diviz, these calculations are performed on computing servers through the use of web-services[4] published by the Decision Deck project. The idea behind these web-services is to allow anyone who is connected to the Internet to access a large amount of MCDA algorithms without having to install them on their personal computer. As a consequence, diviz does not contain any algorithmic modules, but requires a connection to the Internet to access the calculation resources.

If the execution of the workflow is successful, the outputs of each of the components are available to the user (see Figure 2 for the output of a module which plots a graphical representation of a performance table). This allows to easily tune the algorithms by analyzing all the intermediary results of the execution. Besides, the history of the past executions is kept in diviz and can be viewed by the user. In particular, if a workflow is modified, the former executions' results and their associated workflows are still available.

[3] Note that in the sequel we will unthinkingly use the words *component*, *program* or *module* to describe the algorithms which can be combined in diviz.

[4] http://www.decision-deck.org/ws

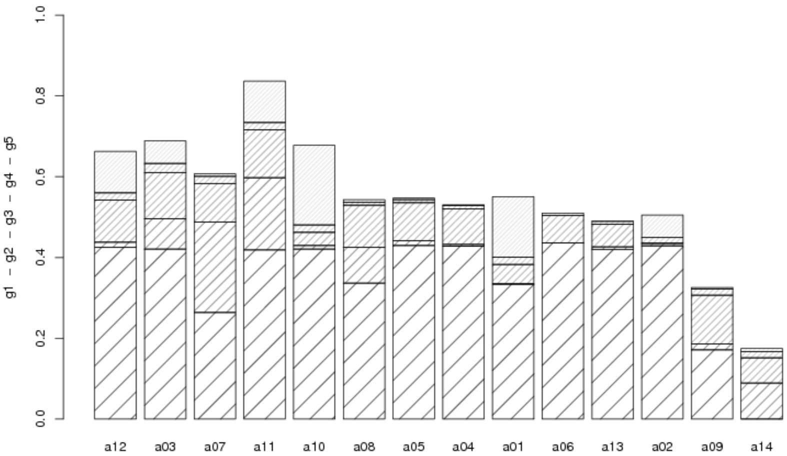

Figure 2: Graphical representation of a performance table: each bar represents an alternative, whose evaluations for the 5 criteria are stacked up).

Available algorithmic components

As already mentioned earlier, the algorithmic elements are available in diviz via web-services. At the time of writing, there are 32 components present which are divided into 4 main categories:

1. *calculation components* containing aggregation operators, disaggregation techniques, analysis elements, etc.;
2. *methods* containing full MCDA methods;
3. *visualization components* containing modules allowing to represent graphically certain input and output data elements;
4. *reporting components* containing techniques to create aggregated reports of multiple output data pieces.

Currently diviz contains, among many others, functions from the kappalab library (Grabisch et al., 2009) which can be used for Choquet integral-based MCDA, disaggregation techniques based on outranking relations (Bisdorff et al., 2009) and additive value functions (ACUTA, Bous et al., 2010), as well as the SMAA method (Lahdelma et al., 1998). A detailed list of all available components can be found on the website of diviz[5].

Note that at the time of writing, various researchers of the Decision Deck project are working on the implementation of MCDA techniques in view of publishing them as web-services which can then be accessed from diviz.

XMCDA: the key to interactions and data visualization

One of the key points of diviz is the possibility to combine various MCDA algorithms originating from heterogeneous researchers in complex workflows. To facilitate this interoperability, the Decision Deck project suggests to use a data standard called XMCDA[6] to store the inputs and outputs of MCDA programs. The XMCDA markup language is written in XML, a general-purpose syntax for defining markup languages.

It follows that the web-services proposed by Decision Deck accept data written according to XMCDA and generate data in this same format.

In diviz this has three direct consequences:

- first, all components available in diviz can interoperate. Typically, the output of one algorithm can be injected into other elementary modules without requiring data transformations;
- second, the inputs and the outputs of the elementary components in diviz are typed with respect to the different data types defined in XMCDA. This allows a very precise control of the various algorithms, and facilitates the creation of complex combinations of components;
- third, we exploit a powerful feature of XML based markup languages to convert XMCDA documents into HTML pages for a convenient visualisation of their contents in a web browser integrated in diviz.

“Methods” comparison

[5] http://www.decision-deck.org/diviz

[6] http://www.decision-deck.org/xmcda

Next to designing and executing MCDA workflows, diviz can also be a convenient tool to compare the outputs of various methods on the same input data. Up to now, such a task was far from easy, as no unified software platform for MCDA techniques existed. However, with diviz and its possibility to construct complex workflows, it is easy to connect a set of data to various workflows representing different MCDA methods and to compare their outputs.

Workflow sharing

The diviz software enables to export any workflow, with or without the data, as an archive. The latter can then be shared with any diviz user, who can import it (by loading the archive) into his software and continue the development of the workflow or execute it on the original data.

Consequently, diviz can be used as a dissemination tool in combination with a research article. Indeed, the authors of a new MCDA technique could propose the corresponding diviz workflow together with an appropriate data set as supplementary electronic material with their article. This would certainly contribute to a larger dissemination of new MCDA algorithms and facilitate their acceptance among many researchers.

A quick look at the architecture

Technically speaking, diviz is a classical 3-tier application made of: the client which has been described in this section, a component accessing the Decision Deck web-services, and a server. The diviz server's main task consists in planning and controlling the execution of the submitted workflows, making it possible for different components in the workflow to be executed in parallel, when appropriate. diviz users download the client only; the latter connects to the server which takes care of distributing the computations to the dedicated web-services; it gathers all intermediary and final results which are ultimately sent back to the user.

3. Conclusion and perspectives

In this short article we have presented diviz which can be used to design and execute algorithmic MCDA workflows and to disseminate research results. The diviz software is being constantly improved, and the number of available components is quickly growing.

To conclude this article, we would like to make a call for participation: your MCDA algorithm is welcome in diviz, and if you would like to make it available as a web-service via diviz, please have a look at the developers corner on the website of Decision Deck's web-services[7] and contact the authors of this article for any help request or question.

REFERENCES

Bisdorff R, Meyer P, Veneziano T, *Inverse analysis from a Condorcet robustness denotation of valued outranking relations*. In F. Rossi and A. Tsoukiás (Eds.), Algorithmic Decision Theory. Springer-Verlag Berlin Heidelberg, LNAI 5783, pp. 180-191, 2009.

Bous G, Fortemps P, Glineur F, Pirlot M, ACUTA: A novel method for eliciting additive value functions on the basis of holistic preference statements, European Journal of Operational Research, submitted, 2010.

Grabisch M, Kojadinovic I, Meyer P, kappalab: Non additive measure and integral manipulation functions, R package version 0.4-4, 2009.

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA Data Mining Software: An Update, *SIGKDD Explorations*, Volume 11, Issue 1, 2009.

Jacquet-Lagrèze E, Siskos Y, *Assessing a set of additive utility functions for multicriteria decision making: The UTA method,* European Journal of Operational Research, 10(2): 151–164, 1982.

Lahdelma R, Hokkanen J, Salminen P, *SMAA - Stochastic multiobjective acceptability analysis*, European Journal of Operational Research, 106 (1): 137-143, 1998.

R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0, http://www.R-project.org, 2009.

Roy B, *Classement et choix en présence de points de vue multiples (la méthode ELECTRE)*, RIRO, 8:57–75, 1968.

[7] http://www.decision-deck.org/ws/howtos.html