Presentation of the DECISION-DECK project

Raymond Bisdorff

University of Luxembourg, FSTC/CSC, L-1359, Luxembourg, raymond.bisdorff@uni.lu

Abstract

The DECISION-DECK (D2) project's goal is an open source collaborative development of a generic Multiple Criteria Decision Analysis (MCDA) software platform composed of modular components. The genuine purpose is to provide effective tools for decision aid consultants, for researchers in the field of MCDA, as well as for OR teachers.

1. Purpose

The D2 project's objective is to provide an open source software, composed of various modular components, pertaining to the field of Multiple Criteria Decision Analysis (MCDA). It should give a user the possibility to add, modify or simply use existing plugged-in functionalities (plugins). These constituents can either be complete MCDA methods or elements common to a large range of procedures. The typical end-user of the D2 platform is an MCDA researcher, an MCDA consultant or a teacher in an academical institution.

The following MCDA methods are presently implemented:

- sorting of alternatives into ordered classes based on an outranking relation (IRIS),
- progressive best choice method based on an outranking relation (Rubis),
- best choice method based on an additive aggregation model accepting imprecise information on the scaling coefficients (VIP), and
- ranking of alternatives with a set of value functions (UTA-GMS/GRIP).

In order to make these various software plugins reusable by the whole community, their implementation has to be compliant with a set of predefined conventions and formats. Quite obviously, the success of the D2 project relies on a collaborative development effort from the MCDA community.

The D2 software is written in the Java programming language and is therefore platform independent. Its latest version can be downloaded from the collaborative software development management system Sourceforge. Two kinds of implementation designs are available: on the one hand a Java client which implements locally the MCDA methods (D2), and on the other hand a distributed web service and AJAX based architecture, serving the MCDA methods from a distributed web server (D3). Note that for example the Rubis choice method is implemented as such a web service on the Ernst - Schroeder server at the University of Luxembourg.

One of the most valuable features of the Decision - Deck software is the effective consideration of specific roles such as decision maker, evaluator, coordinator or facilitator in a given decision analysis project. For instance, evaluators from different distant places may communicate their evaluations via their local D2 clients to the common decision analysis project under the supervision of the project coordinator, whereas the decision maker may input his personal preferences via method-specific criteria tuning facilities offered in his local client (see Figure 1).

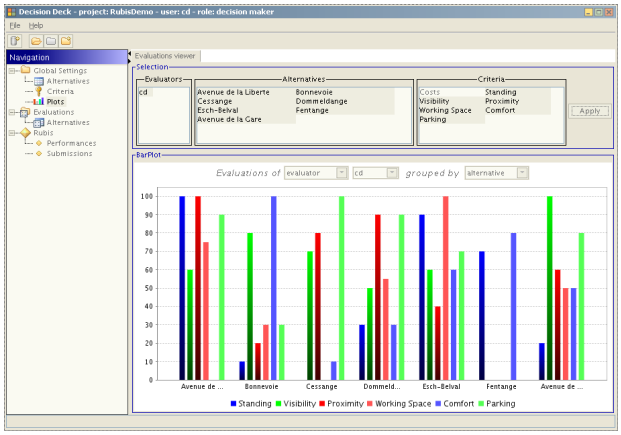

Figure 1: One of the interesting features offered by the DECISION - DECK software is the common availability of visualisation resources as illustrated in the picture above. The snapshot, taken from a D2 Rubis plugin session, shows the performances of the alternatives on a subset of criteria in a column chart style.

One of the ongoing tasks of the D2 project is to develop

and maintain the Universal MCDA Modeling Language (UMCDA-ML), an XML standard which describes in a generic way the inputs and the outputs of MCDA methods, as well as the different steps of a decision analysis workflow. The purpose of UMCDA-ML (see Figure 2) is on the one hand, to allow an easy integration of MCDA web services, such as the Rubis server above and, on the other hand, to facilitate communications and data exchanges between core components of the software platform.

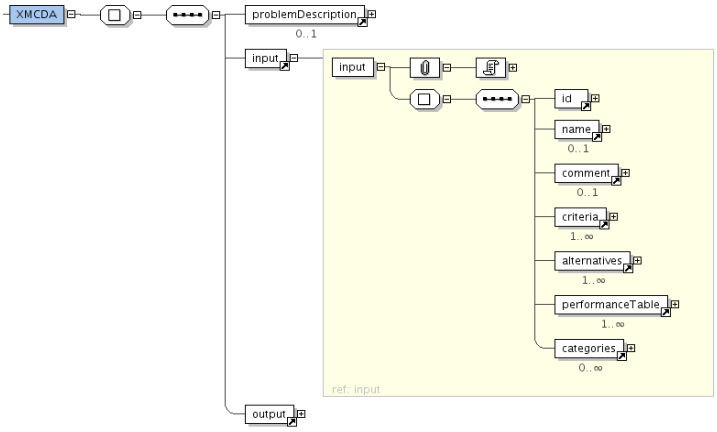

Figure 2: Summary view of a part of the UMCDA - ML specifications.

2. Founding institutional partners

- Faculté Polytechnique de Mons, MathrRO, (M. Pirlot)
- University Paris - Dauphine, Lamsade, (V. Mousseau)
- University of Luxembourg, Computer Science and Communications, (R. Bisdorff)
- Karmic software research, Paris (M. Zam)
- INESC Coimbra, Coimbra, Portugal (L. Dias)
- Université Libre de Bruxelles, Belgique (Y. De Smet)

3. References

- The official DECISION-DECK Project Site: http://www.decision-deck.org
- The DECISION-DECK Sourceforge software download site: http://decision-deck.sourceforge.net
- The Ernst-Schroeder D3 server at the University of Luxembourg: http://ernst-schroeder.uni.lu/d3/ (user=demo, password=D3_Demo)

Distributed DECISION-DECK - D^3

Raymond Bisdorff

University of Luxembourg, FSTC/CSC, L-1359, Luxembourg, raymond.bisdorff@uni.lu

Abstract

The Distributed DECISION-DECK (D^3) project's goal is to design and implement MCDA methods as Web Services (MCDA-WS). Operational requirements are the recommendation of XML standards for representing MCDA problems and solutions. In case of asynchronous MCDA-WS, as required for instance for the Rubis best choice decision method, a specific distributed D^3 session server is needed.

1. MCDA web services

One of the drawback of the original DECISION-DECK (D^2) rich java client software consists in the necessity to reprogramm all new MCDA tool in the Java Eclipse IDE. Many researchers have programmed, however, their methods and tools in other systems and languages. The idea of the D^3 project is to use the web service technology for distributing these existing ressources to potential D^2 clients. An example of such a ressource is the RUBIS server installed at the University of Luxembourg. It takes as input a specific performance tableau and renders as output the RUBIS best choice recommendation. The connection protocol is SOAP (literal RPC) over http with XML encoded input and output documents (See Figure 1)

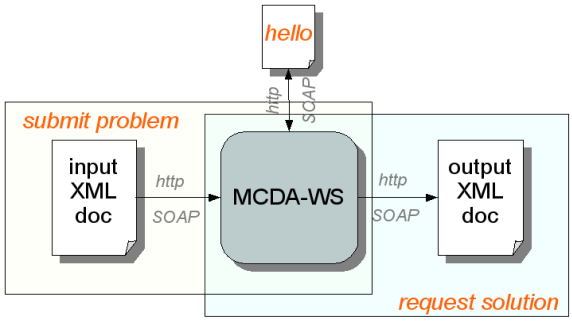

Figure 1: The standard asynchronous MCDA-WS model

Three standard access ports are defined for such an MCDA-WS: a *hello* port for checking the connection with the web service provider, a *submit_problem* port allowing to transfer the XML input document to the distant MCDA solver, and a *request_solution* port allowing to access the XML output document of the MCDA solver.

Due to the asynchronous implementation, a specific D^3 session manager is required in order to coordinate the *submit_problem* and *request_solution* port access.

2. The D^3 server design

The D^3 session manager establishes the operational link between a D^2 client MCDA session and a given MCDA-WS provider as seen in Figure 2 below.

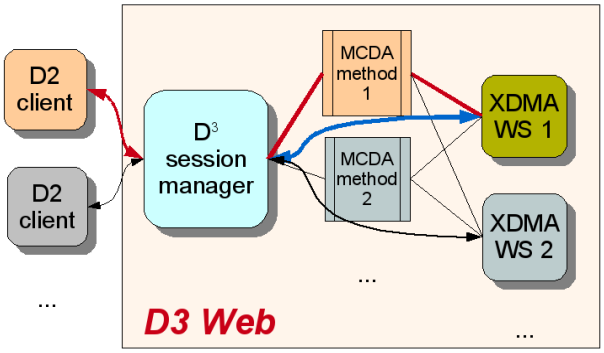

Figure 2: The D^3 plaform architecture

The client may thus choose to use a given MCDA method or tool served by a given MCDA-WS provider.

3. References

A prototype of D^3 server may be consulted at the University of Luxembourg:

http://ernst-schroeder.uni.lu/d3/

(user:demo, password: D3_Demo).

Presentation of the D^2-RUBIS plugin

Raymond Bisdorff

University of Luxembourg, FSTC/CSC, L-1359, Luxembourg, raymond.bisdorff@uni.lu

Abstract

The D^2-RUBIS plugin is a DECISION-DECK resource for tackling the progressive best choice problem in the context of multiple criteria decision aiding. Its genuine purpose is to help a decision maker to determine a single best decision alternative.

1. The Rubis best choice method

Methodologically the RUBIS MCDA method focuses on pairwise comparisons of potential decision alternatives with respect to their performances observed on a consistent family of criteria, which leads to the concept of bipolar-valued outranking digraph (see Figure 1).

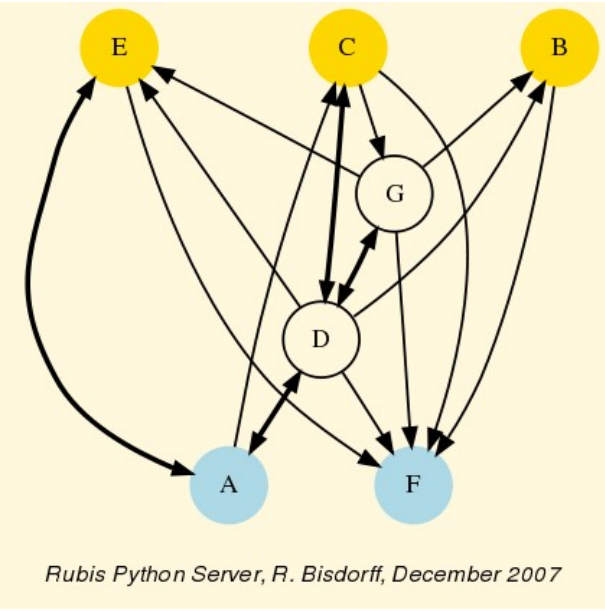

Figure 1: Example of bipolar outranking digraph with RUBIS progressive best choice recommendation marked in yellow.

The best choice method is grounded on a set of five pragmatic principles: - non-retainement for well motivated reasons, - minimal size, - efficient and informative, - effectively better, and - maximally credible, which a best choice recommendation should follow in the context of a progressive decision aiding methodology.

The thorough study and implementation of these five principles as formal properties of potential choices in the bipolar outranking digraph allowed the authors Bisdorff, Meyer and Roubens to define and compute the RUBIS choice recommendation as an extension of the classical digraph kernel concept.

2. The RUBIS Python server

The RUBIS MCDA method is programmed in Python and runs on the <enrst-schroeder.uni.lu> RIA server at the University of Luxembourg. The RUBIS Python server offers a web service access compliant with the D^3 standards and is accessible from a specific D^2 client plugin module.

3. References and links

R. Bisdorff, P. Meyer, M. Roubens (2007), RUBIS: a bipolar-valued outranking method for the choice problem. 4OR, *A Quarterly Journal of Operations Research*, Springer-Verlag, forthcoming. (Online) Electronic version: DOI: 10.1007/s10288-007-0045-5.

R. Bisdorff, M. Pirlot and M. Roubens (2006). Choices and kernels from bipolar valued digraphs. *European Journal of Operational Research,* 175 (2006) 155-170.

R. Bisdorff (2007), *The Python Digraph implementation for Rubis: User manual*. University of Luxembourg, 2007, http://ernst-schroeder.uni.lu/bisdorff/Digraph

D2-Decision-Deck Consortium (2007), *The D2-Decision-Deck project*. http://www.decision-deck.org/

The *Rich Internet Application (RIA) server* at the University of Luxembourg, http://ernst-schroeder.uni.lu/