An MCDA components workflow design, execution and deployment tool

--Latest developments--

Sébastien Bigaret & Patrick Meyer
Télécom Bretagne

17 September 2009 @ 5th Decision Deck Workshop, Brest, France

Outline of the talk

- The essential purpose of diviz
- Latest developments
- A live demo
- A few technical details
- diviz’s future

Observations and questions

- A lot of MCDA “methods” are sequences of elementary algorithmic procedures;
  
  \[ \ldots \rightarrow \text{outranking relation} \rightarrow \text{exploitation} \rightarrow \ldots \]

- Many MCDA “methods” (could) recycle such procedures;

  \[ \ldots \text{like, e.g., data visualisation components} \]

- Have you never wished to see what would happen if \ldots

  \[ \ldots \text{you replaced the outranking relation of Electre by a Rubis one?} \]

The essential purpose of diviz
Observations and questions

- Difficult to make existing MCDA softwares **interoperate**;

  _How to easily reuse the value functions of UTA in MacBeth?_

- Hard to **compare** the outputs of different methods on the same input problem;

  _How does the ranking of the alternatives via a weighted sum differ from that of Promethee?_

- **Laborious** to obtain MCDA softwares (commercial, not available for download, undocumented piece of code, . . .);

  _I like the procedure described in this paper, where can I test it?_

Some general answers

- Make the MCDA softwares **publicly available**;

  _Via an open source calculation library, web services, an MCDA software . . ._

- **Decompose** the MCDA "methods" into elementary components;

- Give the possibility to create **workflows** of such components;

- Separate the GUI from the algorithm;

- Do not reprogram endlessly data visualisation modules;

  _A barchart of criteria weights is first of all a barchart of numeric values associated with criteria!_

Selected answers from the Decision Deck project

- Algorithms and “methods” can interoperate via the **XMCDA** data standard;

- MCDA “methods” and elementary algorithmic components are available as **web services**;

- Data visualisation modules are available as web services;

- The web services framework allows to **integrate** nearly any command line program;

  _programming language independence_

Selected answers from the Decision Deck project

- Currently, web services can be accessed via:

  - standard **SOAP** requests;

  _all services_

  - Decision Deck’s **D2** software.

  _Rubis web service via the Rubis plugin_

  - Decision Deck’s **D3** internet application.

  _Rubis, Kappalab, Random Performance Table_

  - Decision Deck’s **diviz** software;

  _Kappalab functions, weighted sum, Condorcet disaggregation, visualisation elements, concordance relation, ACUTA, SMAA, . . ._
Goals:
- help researchers to construct algorithmic MCDA workflows (="methods") from elementary components;
- help teachers to present MCDA “methods” and let the students experiment their own creations;
- help to easily compare results of different “methods”;
- allow to easily add new MCDA components;
- avoid heavy calculations on your local computer by executing the methods on distant servers;

Properties:
- all components are currently (opensource) web services;
- complete history of past executions;
- components are interoperable through the use of XMCDA;
- components can be reused by many different workflows.

**diviz**’s backbones: XMCDA

- A standardised XML recommendation to represent objects and data structures issued from the field of MCDA;
- General idea: express MCDA concepts through a few general data types written in XML;
- Current version: XMCDA 2.0.0.

http://www.decision-deck.org/xmcd
**diviz’s backbones: XMCDA**

Convenient and flexible representation in a web browser:

**XMCDA + XSL + CSS**

**diviz’s backbones: Praxis**

- A generic algorithmic components workflow design, execution and deployment framework;
- Developed in the LUSSI department of TELECOM Bretagne (P. Picouet, S. Bigaret, P. Tanguy, F. Cadier, ...);
- “Tuned” for various domains:
  - **BioSide**: to help biologists to design and run in-silico experiments;
  - **Euclide**: to test and compare financial models;
  - **diviz**: to design, test, compare and run MCDA “methods”.

**Latest developments and current state**

- **17 August 2009**: private beta released for a few selected testers;
  
  Thank you for your help, a lot of bugs could be fixed!

- **16 September 2009**: public beta released;
  
  You can still help us by sending us bug reports or improvement requests!

**diviz-users@mlistes.telecom-bretagne.eu**
**diviz-announcements@mlistes.telecom-bretagne.eu**
Latest “visible” developments

- **diviz website**: [http://www.decision-deck.org/diviz/](http://www.decision-deck.org/diviz/)
  
  Documentation, workflows, tutorials and descriptions of the programs

- Many new programs available:
  

The name?

diviz means *decision* in Breton ...

Time for a live demo

A few technical details
A generic framework driven by programs’ descriptions only!

Key points:
- Different deployment configurations;
- Execution engine:
  - Fail safe & error recovery;
  - Support for redundancy;
  - Load balancing capable.
- XML-based resources’ description:
  - name, types;
  - domain of validity;
  - inter-dependencies;
  - I/O are typed.

- The workspace directory is human readable!
What diviz is

- A tool for MCDA components workflow (*methods*)
  - design,
  - execution,
  - and **deployment**;
- A simple and standardised data visualisation tool;
- Platform independent;
- Open source.

What diviz is **not**

- A decision aid process designer and manager;
- A manager of stakeholders of the decision aid process;
- An XMCDA data editor;

Diviz’s future

- Basic data edition and manipulation (outside of diviz);
  *with the XMCDA standard, this should not be too difficult.*
- Workflow deployment;
  *so anybody can reuse the workflows that you publish!*
- A more user friendly and lighter GUI *à la* Eclipse;
- Various representations of the workflows.
  *more compact, less links, . . .*
- Allow to integrate "custom" code (R, python, ...)

What we can do
How you can help

What you have to do

WS architecture, independent from diviz.

That’s all folks

http://www.decision-deck.org/diviz