Rationale, Concept & Architecture of a Distributed MCDA application Designer

R. Bis dorff (UniLu)  
Gilles Dodinet & Michel Zam (KarmicSoft, Lamsade)  
5th Decision Deck Workshop  
Brest, September 2009

Agenda

- Rationale
- DecisionDeck software short story
- High hopes and limitations
- D4 architecture / bricks
- D2/D3/D4/Dz interchange overview
- Q&A

Rationale

- Common requirements for any Decision Deck software
  A. Problem data input, computation and visualization of results
  B. Role oriented user management and collaboration
  C. Extensibility: adding and enhancing MCDA methods

Decision Deck software story

- D2: Desktop + plugins
- CL/SV, runs on LAN
### Decision Deck High Hopes and limitations

- **Spreading MCDA methods**
  - Deserving but slow growing community ...
- **Objective reasons & walls**
  - MCDA Method designers: requirement specification / books, algorithms
  - Java/XML developers: implementation => skills difficult to find
  - MCDA tools users: ....
  - Data standardization: promising => stabilization & evolution issues
  - Data and process traceability => Missing link (feedback, trust)
- **But it’s not MCDA’s community fault!**
  - Engineering limitation => Computer science responsibility

### Expectations

<table>
<thead>
<tr>
<th>D2 + D3</th>
<th>D4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Problem Data input, computation and visualization of results</td>
<td>RIA/Distributed/Cloud deployment &amp; ergonomy</td>
</tr>
<tr>
<td>D2 heavy client, D3 no data input</td>
<td></td>
</tr>
<tr>
<td>B. Role oriented user management (D2) versus distributed workflow (D3)</td>
<td>Configurable user roles &amp; processes</td>
</tr>
<tr>
<td>C. Extensibility: adding &amp; enhancing MCDA methods</td>
<td>Declarative designer (no Java, no XML) + MDE</td>
</tr>
<tr>
<td>D2 + plugins, D3-WS</td>
<td></td>
</tr>
<tr>
<td>Auditing &amp; Traceability</td>
<td></td>
</tr>
<tr>
<td>(feature) inheritance + interface (XMCDA) = continuous platform</td>
<td>Online Viral Community (Moodle-like)</td>
</tr>
</tbody>
</table>
D4 Architecture

- Unified portal
  - “Distributed (Declarative) Designer for Decision Deck”
  - Online webapp designer/launcher for MCDA webapps
  - Experimental prototype and ongoing project

- Bricks
  - Data
  - GUI
  - Computation
  - Process
  - Auditing

Global entry point

D4: Distributed MCDA webapp Designer and Launcher

- Description

D2 - remastered

- Persistance
- Standard architecture JEE/JPA/DBMS
- Fine coarse (atomic) granularity & traceability
- Generic storage schema
- Concurrent data and schema evolution
- XSD/XML I/O (XMCDA++)
- Declarative designer

Data Brick
GUI Brick

- RIA (D3 like) data centered GUI: ExtJS
- High level templates & dynamic factories (Mydraft)
- Distributed Declarative Designer

Custom actions

- Custom action ➔ client code ➔ server code ➔ ...

Computation Brick

- (d2 / d3: java plugins & WS)
- Black Box (WS + XMCDA++ I/O)
- White Box (scripting languages Python, JS, …)
  - Scripts: stored as persistent data
  - Script Editor
- Advantages
  - No Java/XML skills required, no admin deployment effort required, radical shorter application life cycle, better dissemination, open source algorithmic implementations

Process Brick

- Configurable State Machines
- User roles and grants management system
- MCDA process modelling
- Abstract bricks chaining language
- Declarative Designer
Auditing Brick

- Fine coarse historization of data and data structure evolutions
- Process execution logs
- Karmicsoft traceability engine
- Still using the RDBMS

D4 vision

- GUI
- Process
- Computation
- Data
- Auditing
- JEE / JPA
- DBMS

Global interaction

- D2/cl
- D3
- Dz/cl
- WS

Conclusion

- « Distributed Deigner for Decision Deck »
- High-level (methodologist level) online tool
- Pre-alpha version of an experimental prototype
- Hosted: http://leopold-loewenheim.uni.lu
- More to come soon. Please stay tuned