The Decision Deck Project

Towards Open Source Software Tools Implementing Multiple Criteria Decision Aid

Decision Deck Consortium

Raymond Bisdorff[†] and Patrick Meyer[‡]

[†]University of Luxembourg,[‡]TELECOM Bretagne

July 2010 @ *MCDM*$\{M|A\}$ Summer School

École Centrale Paris

Decision Deck's purpose

The Decision Deck project aims at collaboratively developing **open source software tools** implementing Multiple Criteria Decision Aid (MCDA).

Its purpose is to provide effective tools for three types of users :

- **practitioners** who use MCDA tools to support actual decision makers involved in real world decision problems ;
- **teachers** who present MCDA methods in courses, for didactic purposes ;
- **researchers** who want to test and compare methods or to develop new ones.

Decision Deck's purpose

Promote MCDA research and make it more visible to the "outside world".

Generate new open research issues and support them.

Help structuring a community composed of

- researchers in the field of MCDA ;
- software developers ;
- users/decision aid consultants.

Decision Deck's purpose

Promote MCDA research and make it more visible to the "outside world".

Generate new open research issues and support them.

Help structuring a community composed of

- researchers in the field of MCDA ;
- software developers ;
- users/decision aid consultants.

Decision Deck's purpose

Promote MCDA research and make it more visible to the "outside world".

Generate new open research issues and support them.

Help structuring a community composed of

- researchers in the field of MCDA ;
- software developers ;
- users/decision aid consultants.

Decision Deck's purpose

Promote MCDA research and make it more visible to the "outside world".

Generate new open research issues and support them.

Help structuring a community composed of

- researchers in the field of MCDA ;
- software developers ;
- users/decision aid consultants.

Decision Deck's purpose

Promote MCDA research and make it more visible to the "outside world".

Generate new open research issues and support them.

Help structuring a community composed of

- researchers in the field of MCDA ;
- software developers ;
- users/decision aid consultants.

Outline of the talk

- Overview of the Decision Deck project ;
  - *A little bit of history & visible activities ;*
  - *The Decision Deck Consortium & 6 initiatives ;*
- Focus on 3 initiatives ;
- The future & what you can do.

Outline of the talk

- Overview of the Decision Deck project ;
  - *A little bit of history & visible activities ;*
  - *The Decision Deck Consortium & 6 initiatives ;*
- Focus on 3 initiatives ;
  - XMCDA standard ;
  - [illegible] ;
  - [illegible] ;
- The future & what you can do.

Outline of the talk

- Overview of the Decision Deck project ;
  - *A little bit of history & visible activities ;*
  - *The Decision Deck Consortium & 6 initiatives ;*
- Focus on 3 initiatives ;
  - *XMCDA standard ;*
  - *MCDA web services ;*
  - *GUIs ;*
- The future & what you can do.

Outline of the talk

- Overview of the Decision Deck project ;
  - *A little bit of history & visible activities ;*
  - *The Decision Deck Consortium & 6 initiatives ;*
- Focus on 3 initiatives ;
  - `XMCDA` *standard ;*
  - *MCDA web services ;*
  - *diviz.*
- The future & what you can do.

Outline of the talk

- Overview of the Decision Deck project ;
  - *A little bit of history & visible activities ;*
  - *The Decision Deck Consortium & 6 initiatives ;*
- Focus on 3 initiatives ;
  - `XMCDA` *standard ;*
  - *MCDA web services ;*
  - *diviz.*
- The future & what you can do.

Outline of the talk

- Overview of the Decision Deck project ;
  - *A little bit of history & visible activities ;*
  - *The Decision Deck Consortium & 6 initiatives ;*
- Focus on 3 initiatives ;
  - `XMCDA` *standard ;*
  - *MCDA web services ;*
  - *diviz.*
- The future & what you can do.

Outline of the talk

- Overview of the Decision Deck project ;
  - *A little bit of history & visible activities ;*
  - *The Decision Deck Consortium & 6 initiatives ;*
- Focus on 3 initiatives ;
  - `XMCDA` *standard ;*
  - *MCDA web services ;*
  - *diviz.*
- The future & what you can do.

Outline of the talk

- Overview of the Decision Deck project ;
  - *A little bit of history & visible activities ;*
  - *The Decision Deck Consortium & 6 initiatives ;*
- Focus on 3 initiatives ;
  - `XMCDA` *standard ;*
  - *MCDA web services ;*
  - *diviz.*
- The future & what you can do.

But first . . .

. . . what is MCDA ?

- **Alternatives** (decision actions) are evaluated on multiple preference dimensions (**criteria**, attributes) ;

  e.g. cars evaluated according to their price, av. fuel consumption, look, max. speed, . . .

- **Help** to determine the *best* alternative, rank the alternatives or assign them to ordered classes ;

- By taking into account the **preferences** of the decision maker.

But first . . .

. . . what is MCDA ?

- **Alternatives** (decision actions) are evaluated on multiple preference dimensions (**criteria**, attributes) ;

  e.g. cars evaluated according to their price, av. fuel consumption, look, max. speed, . . .

- **Help** to determine the *best* alternative, rank the alternatives or assign them to ordered classes ;

- By taking into account the **preferences** of the decision maker.

But first . . .

. . . what is MCDA ?

- **Alternatives** (decision actions) are evaluated on multiple preference dimensions (**criteria**, attributes) ;

  e.g. cars evaluated according to their price, av. fuel consumption, look, max. speed, . . .

- **Help** to determine the *best* alternative, rank the alternatives or assign them to ordered classes ;

- By taking into account the **preferences** of the decision maker.

But first . . .

. . . what is MCDA ?

- **Alternatives** (decision actions) are evaluated on multiple preference dimensions (**criteria**, attributes) ;

  e.g. cars evaluated according to their price, av. fuel consumption, look, max. speed, . . .

- **Help** to determine the *best* alternative, rank the alternatives or assign them to ordered classes ;

- By taking into account the **preferences** of the decision maker.

But first . . .

. . . how does the software *situation* look like in the field ?

- many different **methods** ;
- many different **softwares** ;
- **no unified** software to test the same problem on various methods.

But first . . .

. . . how does the software *situation* look like in the field ?

- many different **methods** ;
- many different **softwares** ;
- **no unified** software to test the same problem on various methods.

But first . . .

. . . how does the software *situation* look like in the field ?

- many different **methods** ;
- many different **softwares** ;
- **no unified** software to test the same problem on various methods.

Overview of the Decision Deck project

- *A bit of history & visible activities ;*
- *The Decision Deck Consortium & 6 initiatives.*

Overview of the Decision Deck project

1. A bit of history & visible activities

Decision Deck's history

- **2003**

  EVAL project, financed by the Wallon Region (B), (SMG-ULB, MathRO-Mons, SCSI-ULB) ;

- **2006**

  Lamsade (Paris-Dauphine) joined the project and restructured the existing platform with plugins (in conjunction with KarmicSoft)

Decision Deck's history

- **2003**

  EVAL project, financed by the Wallon Region (B), (SMG-ULB, MathRO-Mons, SCSI-ULB) ;

- **2006**

  Lamsade (Paris-Dauphine) joined the project and restructured the existing platform with plugins (in conjunction with KarmicSoft)

Birth of the Decision Deck project and of the D2 client

Decision Deck's history

- **2003**

  EVAL project, financed by the Wallon Region (B), (SMG-ULB, MathRO-Mons, SCSI-ULB) ;

- **2006**

  Lamsade (Paris-Dauphine) joined the project and restructured the existing platform with plugins (in conjunction with KarmicSoft)

  Birth of the Decision Deck *project and of the D2 client ;*

Decision Deck's history

- **2003**

  EVAL project, financed by the Wallon Region (B), (SMG-ULB, MathRO-Mons, SCSI-ULB) ;

- **2006**

  Lamsade (Paris-Dauphine) joined the project and restructured the existing platform with plugins (in conjunction with KarmicSoft)

Birth of the **Decision Deck** *project and of the D2 client* ;

Decision Deck's history

- **2003**

  EVAL project, financed by the Wallon Region (B), (SMG-ULB, MathRO-Mons, SCSI-ULB) ;

- **2006**

  Lamsade (Paris-Dauphine) joined the project and restructured the existing platform with plugins (in conjunction with KarmicSoft)

  Birth of the **Decision Deck** *project and of the D2 client* ;

Decision Deck's history

- **2007 – 2008**

  SMA (UL) joined in and invested in the Decision Deck project (RUBIS plugin for **D2**, **D3**, **web services**, **XMCDA-1.0**);

- **2007 – 2010**

  Contributions from Portugal (INESC Coimbra) and Poland (ICS Poznan) (plugins for **D2**);

- **2008 – 2010**

  Contributions from Télécom Bretagne (**diviz** prototype, **XMCDA-2.0**, diviz web services);

  Contributions from UL (**XMCDA-2.0** RUBIS server, **D4** prototype).

Decision Deck's history

- **2007 – 2008**

  SMA (UL) joined in and invested in the Decision Deck project (RUBIS plugin for **D2**, **D3**, **web services**, `XMCDA-1.0`);

- **2007 – 2010**

  Contributions from Portugal (INESC Coimbra) and Poland (ICS Poznan) (plugins for **D2**);

- **2008 – 2010**

  Contributions from Télécom Bretagne (diviz prototype, `XMCDA-2.0`, diviz web services);

  Contributions from UL (`XMCDA-2.0` RUBIS server, D4 prototype).

Decision Deck's history

- **2007 – 2008**

  SMA (UL) joined in and invested in the Decision Deck project (RUBIS plugin for **D2**, **D3**, **web services**, **XMCDA-1.0**);

- **2007 – 2010**

  Contributions from Portugal (INESC Coimbra) and Poland (ICS Poznan) (plugins for **D2**);

- **2008 – 2010**

  Contributions from Télécom Bretagne (**diviz** prototype, **XMCDA-2.0**, diviz web services);

  Contributions from UL (**XMCDA-2.0** RUBIS server, D4 prototype).

Decision Deck's history

- **2007 – 2008**

  SMA (UL) joined in and invested in the Decision Deck project (RUBIS plugin for **D2**, **D3**, **web services**, **XMCDA-1.0**);

- **2007 – 2010**

  Contributions from Portugal (INESC Coimbra) and Poland (ICS Poznan) (plugins for **D2**);

- **2008 – 2010**

  Contributions from Télécom Bretagne (**diviz** prototype, **XMCDA-2.0**, diviz web services);

  Contributions from UL (XMCDA-2.0 RUBIS server, D4 prototype).

Decision Deck's history

- **2007 – 2008**

  SMA (UL) joined in and invested in the Decision Deck project (RUBIS plugin for **D2**, **D3**, **web services**, **XMCDA-1.0**);

- **2007 – 2010**

  Contributions from Portugal (INESC Coimbra) and Poland (ICS Poznan) (plugins for **D2**);

- **2008 – 2010**

  Contributions from Télécom Bretagne (**diviz** prototype, **XMCDA-2.0**, diviz web services);

  Contributions from UL (**XMCDA-2.0** RUBIS server, **D4** prototype).

Decision Deck's history

- **2007 – 2008**

  SMA (UL) joined in and invested in the Decision Deck project (RUBIS plugin for **D2**, **D3**, **web services**, XMCDA-**1.0**);

- **2007 – 2010**

  Contributions from Portugal (INESC Coimbra) and Poland (ICS Poznan) (plugins for **D2**);

- **2008 – 2010**

  Contributions from Télécom Bretagne (**diviz** prototype, XMCDA-**2.0**, diviz web services);

  Contributions from UL (XMCDA-**2.0** RUBIS server, **D4** prototype).

Decision Deck's history

- **2007 – 2008**

  SMA (UL) joined in and invested in the Decision Deck project (RUBIS plugin for **D2**, **D3**, **web services**, XMCDA-**1.0**);

- **2007 – 2010**

  Contributions from Portugal (INESC Coimbra) and Poland (ICS Poznan) (plugins for **D2**);

- **2008 – 2010**

  Contributions from Télécom Bretagne (**diviz** prototype, XMCDA-**2.0**, diviz web services);

  Contributions from UL (XMCDA-**2.0** RUBIS server, **D4** prototype).

Visible activities

- 6 past workshops

  Luxembourg, Paris, Coimbra, Mons, Brest, Coimbra ;

- 1 future workshop

  Paris, Centre de Paris, October 7-8, 2010

- 2 developers days

- 6 steering meetings

  Luxembourg, Paris, Brussels

- 7 specifications meetings

  Luxembourg, Paris, ...

Visible activities

- 6 past workshops

  Luxembourg, Paris, Coimbra, Mons, Brest, Coimbra ;

- 1 future workshop

  Ecole Centrale de Paris, October 7–9, 2010 ;

- 2 developers days

- 6 steering meetings

  Luxembourg, Paris, Brest, ...

- 7 specifications meetings

  Luxembourg, Paris, ...

Visible activities

- 6 past workshops

  Luxembourg, Paris, Coimbra, Mons, Brest, Coimbra ;

- 1 future workshop

  Ecole Centrale de Paris, October 7–9, 2010 ;

- 2 developers days

  Luxembourg, Paris ;

- 6 steering meetings

  Luxembourg, Paris, Brest, ...

- 7 specifications meetings

  Luxembourg, Paris, ...

Visible activities

- 6 past workshops

  Luxembourg, Paris, Coimbra, Mons, Brest, Coimbra ;

- 1 future workshop

  Ecole Centrale de Paris, October 7–9, 2010 ;

- 2 developers days

  Luxembourg, Paris ;

- 6 steering meetings

  Luxembourg, Paris, Brest, ...

- 7 specifications meetings

  Luxembourg, Paris, ...

Visible activities

- 6 past workshops

  Luxembourg, Paris, Coimbra, Mons, Brest, Coimbra ;

- 1 future workshop

  Ecole Centrale de Paris, October 7–9, 2010 ;

- 2 developers days

  Luxembourg, Paris ;

- 6 steering meetings

  Luxembourg, Paris, Brussels, ...

- 7 specifications meetings

  Luxembourg, Paris, ...

Visible activities

- 6 past workshops

  Luxembourg, Paris, Coimbra, Mons, Brest, Coimbra ;

- 1 future workshop

  Ecole Centrale de Paris, October 7–9, 2010 ;

- 2 developers days

  Luxembourg, Paris ;

- 6 steering meetings

  Luxembourg, Paris, Brussels, . . .

- 7 specifications meetings

  Luxembourg, Paris, . . .

Visible activities

- 6 past workshops

  Luxembourg, Paris, Coimbra, Mons, Brest, Coimbra ;

- 1 future workshop

  Ecole Centrale de Paris, October 7–9, 2010 ;

- 2 developers days

  Luxembourg, Paris ;

- 6 steering meetings

  Luxembourg, Paris, Brussels, . . .

- 7 specifications meetings

  Luxembourg, Paris, . . .

Visible activities

- 6 past workshops

  Luxembourg, Paris, Coimbra, Mons, Brest, Coimbra ;

- 1 future workshop

  Ecole Centrale de Paris, October 7–9, 2010 ;

- 2 developers days

  Luxembourg, Paris ;

- 6 steering meetings

  Luxembourg, Paris, Brussels, . . .

- 7 specifications meetings

  Luxembourg, Paris, . . .

Visible activities

- 6 past workshops

  Luxembourg, Paris, Coimbra, Mons, Brest, Coimbra ;

- 1 future workshop

  Ecole Centrale de Paris, October 7–9, 2010 ;

- 2 developers days

  Luxembourg, Paris ;

- 6 steering meetings

  Luxembourg, Paris, Brussels, . . .

- 7 specifications meetings

  Luxembourg, Paris, . . .

Visible activities

- 6 past workshops

  Luxembourg, Paris, Coimbra, Mons, Brest, Coimbra ;

- 1 future workshop

  Ecole Centrale de Paris, October 7–9, 2010 ;

- 2 developers days

  Luxembourg, Paris ;

- 6 steering meetings

  Luxembourg, Paris, Brussels, . . .

- 7 specifications meetings

  Luxembourg, Paris, . . .

Overview of the Decision Deck project

2. The Decision Deck Consortium & 6 initiatives

The Decision Deck Consortium

- A french non profit association[1] which **steers** and **manages** the project ;
- Headed by an administration board

  V. Mousseau (*pres.*), P. Meyer (*trea.*), M. Pirlot (*sec.*), R. Bisdorff, O. Cailloux ;
- Guided by a general assembly ;
- Individual memberships ! (30€)
- Formerly known as the "steering committee".

[1] Association loi 1901

The Decision Deck Consortium

- A french non profit association[1] which **steers** and **manages** the project ;
- Headed by an administration board

  V. Mousseau (*pres.*), P. Meyer (*trea.*), M. Pirlot (*sec.*), R. Bisdorff, O. Cailloux ;

- Guided by a general assembly ;
- **Individual** memberships ! (30€)
- Formerly known as the "steering committee".

[1] Association loi 1901

The Decision Deck Consortium

- A french non profit association[1] which **steers** and **manages** the project ;
- Headed by an administration board

  V. Mousseau (*pres.*), P. Meyer (*trea.*), M. Pirlot (*sec.*), R. Bisdorff, O. Cailloux ;
- Guided by a general assembly ;
- **Individual** memberships ! (30€)
- Formerly known as the "steering committee".

[1] Association loi 1901

The Decision Deck Consortium

- A french non profit association[1] which **steers** and **manages** the project ;
- Headed by an administration board

  V. Mousseau (*pres.*), P. Meyer (*trea.*), M. Pirlot (*sec.*), R. Bisdorff, O. Cailloux ;
- Guided by a general assembly ;
- **Individual** memberships ! (30€)
- Formerly known as the "steering committee".

[1] Association loi 1901

The Decision Deck Consortium

- A french non profit association[1] which **steers** and **manages** the project ;
- Headed by an administration board

  V. Mousseau (*pres.*), P. Meyer (*trea.*), M. Pirlot (*sec.*), R. Bisdorff, O. Cailloux ;
- Guided by a general assembly ;
- **Individual** memberships ! (30€)
- Formerly known as the "steering committee".

[1] Association loi 1901

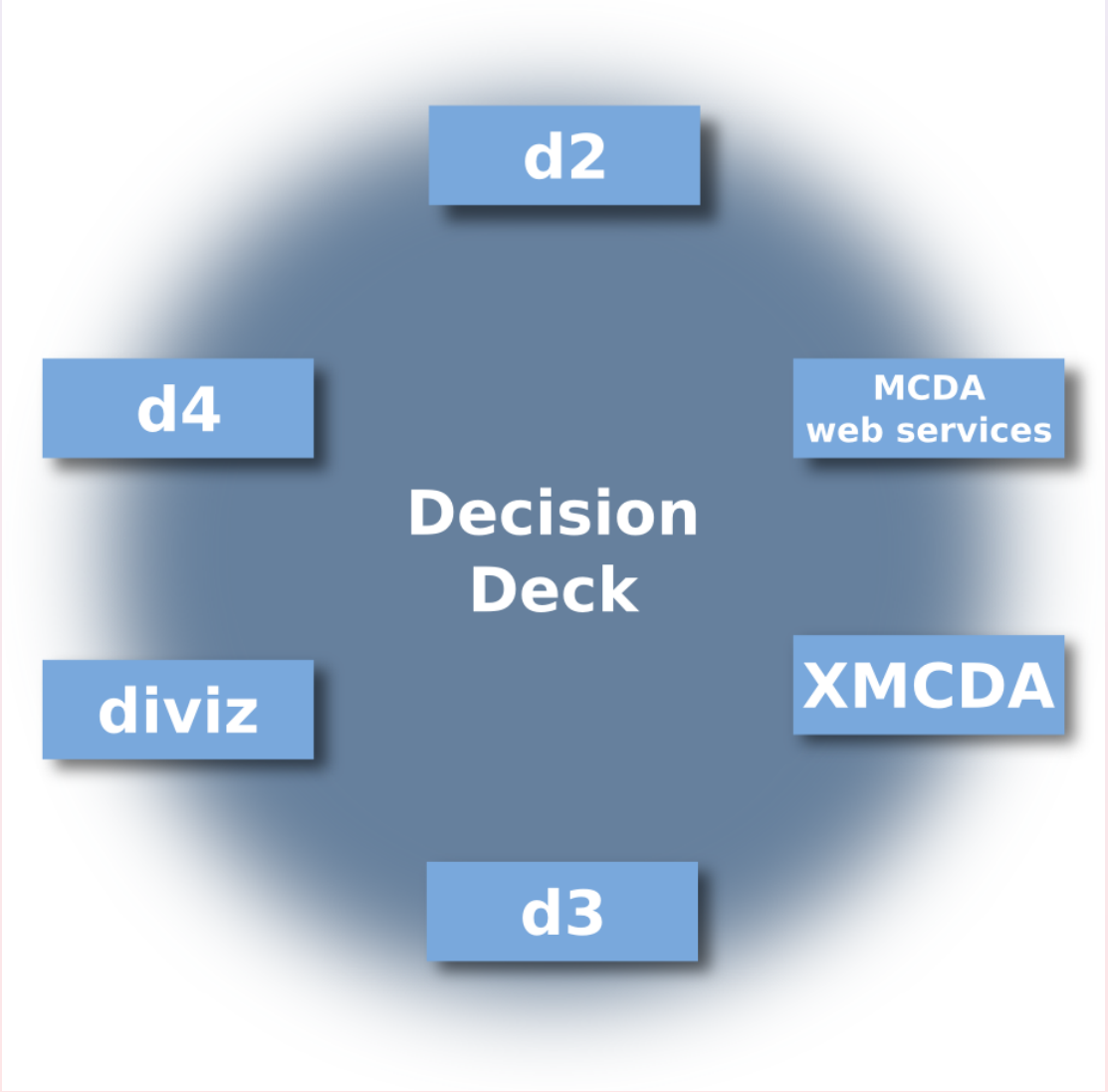
The Decision Deck Consortium

- A french non profit association[1] which **steers** and **manages** the project ;
- Headed by an administration board

  V. Mousseau (*pres.*), P. Meyer (*trea.*), M. Pirlot (*sec.*), R. Bisdorff, O. Cailloux ;
- Guided by a general assembly ;
- **Individual** memberships ! (30€)
- Formerly known as the “steering committee”.

[1] Association loi 1901

6 scientific initiatives



6 scientific initiatives

D2

A rich open source Java client offering several MCDA methods.

- MCDA methods can be added as plugins ;
- Role management and a first attempt of collaborative work ;
- Currently offering IRIS, RUBIS and VIP, UTA-GMS/GRIP.

6 scientific initiatives

D2

A rich open source Java client offering several MCDA methods.

- MCDA methods can be added as **plugins**;
- **Role** management and a first attempt of collaborative work;
- Currently offering IRIS, RUBIS and VIP, UTA-GMS/GRIP.

6 scientific initiatives

D2

A rich open source Java client offering several MCDA methods.

- MCDA methods can be added as **plugins**;
- **Role** management and a first attempt of collaborative work;
- Currently offering IRIS, RUBIS and VIP, UTA-GMS/GRIP.

6 scientific initiatives

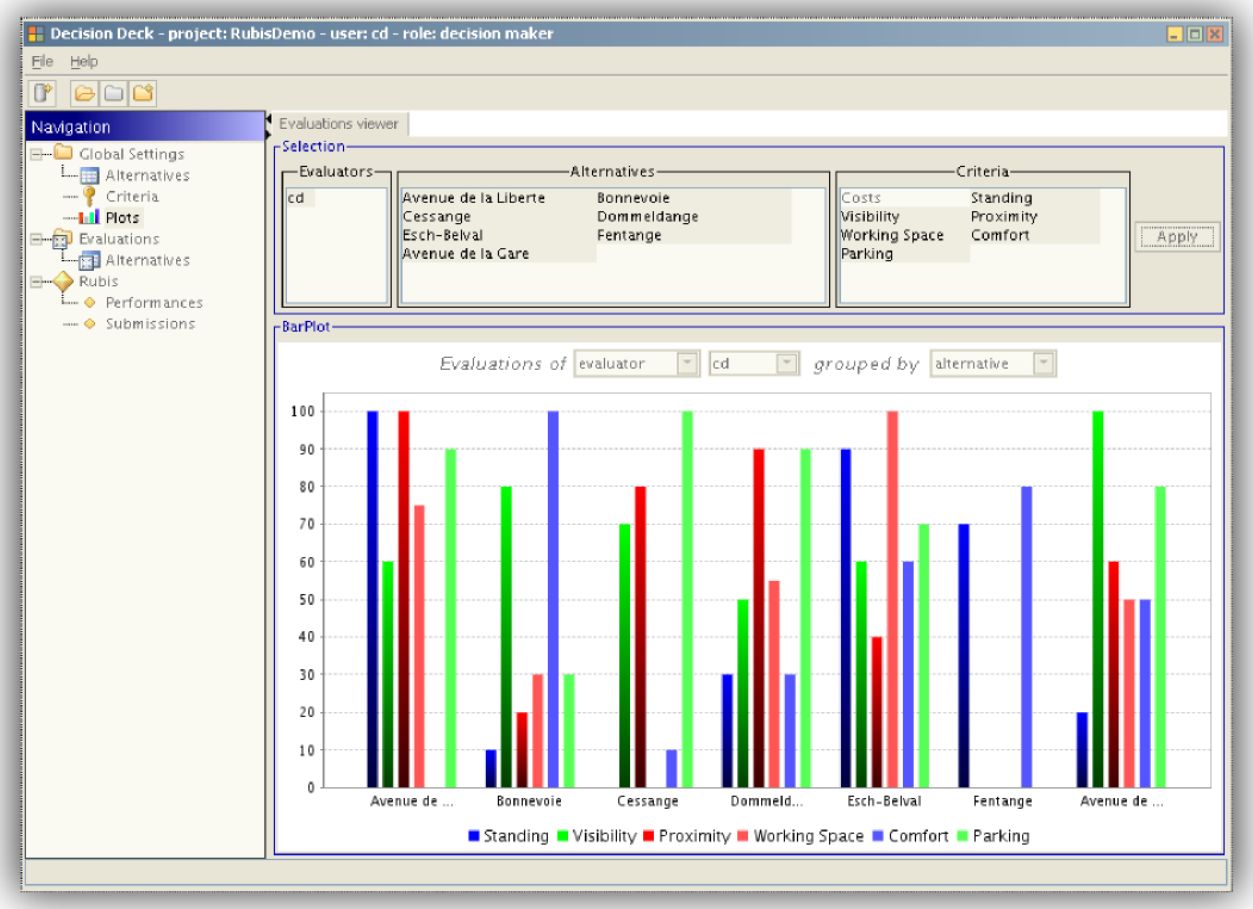
D2

A rich open source Java client offering several MCDA methods.

- MCDA methods can be added as **plugins**;
- **Role** management and a first attempt of collaborative work;
- Currently offering IRIS, RUBIS and VIP, UTA-GMS/GRIP.

6 scientific initiatives

D2



Time for a demo !

6 scientific initiatives

MCDA web services

Algorithmic components or complete MCDA methods accessible online.

- Reuse of existing implementations of algorithms ;
- Use of any programming language ;
- Currently offering the RUBIS solver and the KAPPALAB R library.

Further details later !

6 scientific initiatives

MCDA web services

Algorithmic components or complete MCDA methods accessible online.

- Reuse of existing implementations of algorithms ;
- Use of any programming language ;
- Currently offering the RUBIS solver and the KAPPALAB R library.

Further details later !

6 scientific initiatives

MCDA web services

Algorithmic components or complete MCDA methods accessible online.

- Reuse of existing implementations of algorithms ;
- Use of any programming language ;
- Currently offering the RUBIS solver and the KAPPALAB R library.

Further details later !

6 scientific initiatives

MCDA web services

Algorithmic components or complete MCDA methods accessible online.

- Reuse of existing implementations of algorithms ;
- Use of any programming language ;
- Currently offering the RUBIS solver and the KAPPALAB R library.

Further details later !

6 scientific initiatives

MCDA web services

Algorithmic components or complete MCDA methods accessible online.

- Reuse of existing implementations of algorithms ;
- Use of any programming language ;
- Currently offering the RUBIS solver and the KAPPALAB R library.

Further details later !

6 scientific initiatives

MCDA web services

Algorithmic components or complete MCDA methods accessible online.

- Reuse of existing implementations of algorithms ;
- Use of any programming language ;
- Currently offering the RUBIS solver and the KAPPALAB R library.

Further details later !

6 scientific initiatives

XMCDA

A standardised XML recommandation to represent objects and data structures issued from the field of MCDA.

- Allow different MCDA algorithms to interact and be easily callable ;
- Direct applications :
  - MCDA web services
  - Standard visualisation of data

6 scientific initiatives

XMCDA

A standardised XML recommandation to represent objects and data structures issued from the field of MCDA.

- Allow different MCDA algorithms to interact and be easily callable ;
- Direct applications :
  - MCDA web services ;
  - Standard visualisation of data.

6 scientific initiatives

XMCDA

A standardised XML recommandation to represent objects and data structures issued from the field of MCDA.

- Allow different MCDA algorithms to interact and be easily callable ;
- Direct applications :
  - MCDA web services ;
  - Standard visualisation of data.

6 scientific initiatives

XMCDA

A standardised XML recommandation to represent objects and data structures issued from the field of MCDA.

- Allow different MCDA algorithms to interact and be easily callable ;
- Direct applications :
  - MCDA web services ;
  - Standard visualisation of data.

6 scientific initiatives

XMCDA

A standardised XML recommandation to represent objects and data structures issued from the field of MCDA.

- Allow different MCDA algorithms to interact and be easily callable ;
- Direct applications :
  - MCDA web services ;
  - Standard visualisation of data.

6 scientific initiatives

XMCDA

```
<alternatives name="myAlternatives">
        <alternative id="x1" name="Red Ferrari"/>
        <alternative id="x2" name="Blue Corvette">
                <type>real</type>
                <active>true</active>
                <reference>false</reference>
        </alternative>
        <alternative id="x3" name="UFO">
                <type>fictive</type>
        </alternative>
 </alternatives>
```

Further details later !

6 scientific initiatives

D3

An open source rich internet application for `XMCDA` web services management.

- Call and basic management of web services ;
- Interface in a web browser.

6 scientific initiatives

D3

An open source rich internet application for `XMCDA` web services management.

- Call and basic management of web services ;
- Interface in a web browser.

6 scientific initiatives

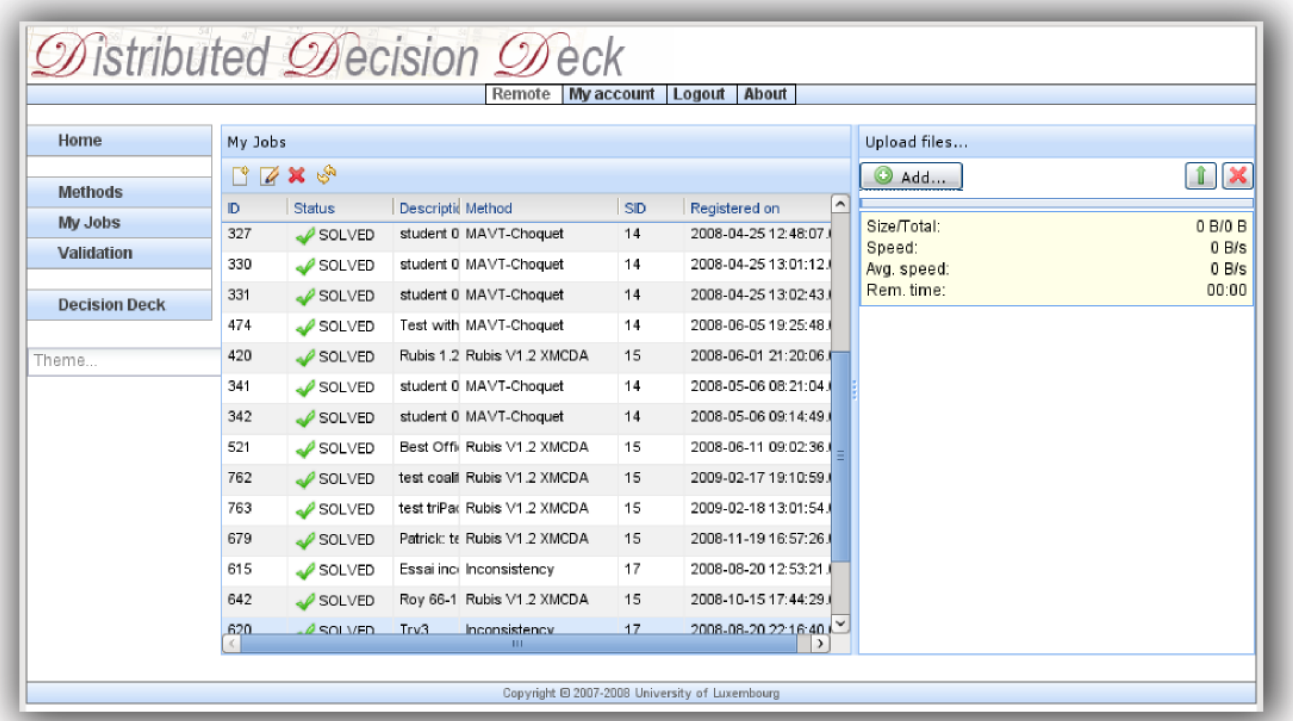
D3

An open source rich internet application for `XMCDA` web services management.

- Call and basic management of web services ;
- Interface in a web browser.

6 scientific initiatives

D3



Time for a demo !

6 scientific initiatives

diviz

An open source Java client and server for `XMCDA` web services composition, workflow management and deployment.

- Call and advanced management of web services ;
- Oriented towards algorithms (and not decision aid processes).

6 scientific initiatives

diviz

An open source Java client and server for `XMCDA` web services composition, workflow management and deployment.

- Call and advanced management of web services ;
- Oriented towards algorithms (and not decision aid processes).

6 scientific initiatives

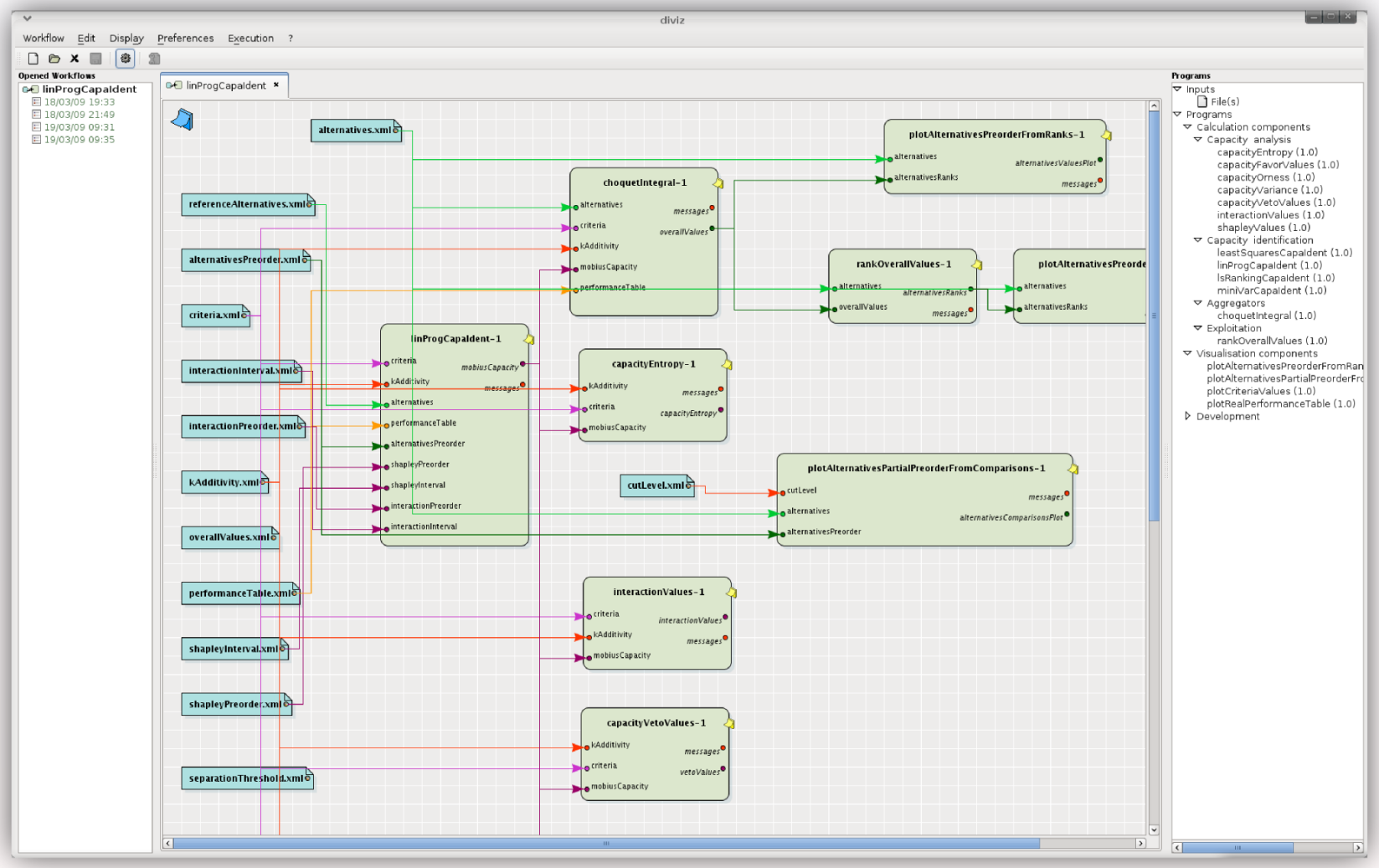
diviz

An open source Java client and server for `XMCDA` web services composition, workflow management and deployment.

- Call and advanced management of web services ;
- Oriented towards algorithms (and not decision aid processes).

6 scientific initiatives

diviz



Further details later !

6 scientific initiatives

D4

A rich internet application host for implementing, running and auditing `XMCDA` compatible decision aid processes.

- Oriented towards decision aid processes and algorithms ;
- Interface in a web browser.

6 scientific initiatives

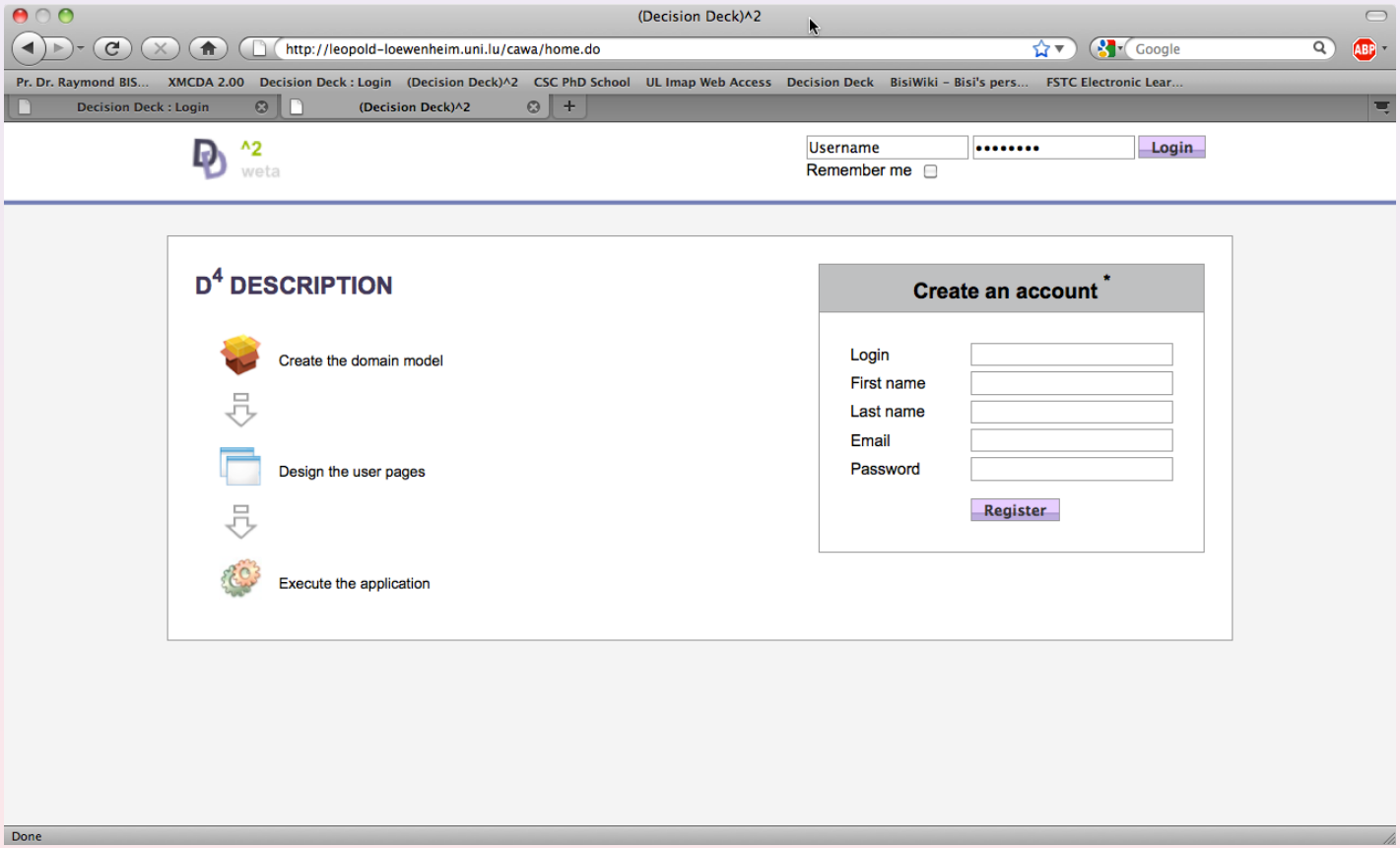
D4

A rich internet application host for implementing, running and auditing `XMCDA` compatible decision aid processes.

- Oriented towards decision aid processes and algorithms ;
- Interface in a web browser.

6 scientific initiatives

d4



Time for a demo !

Key websites

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

  General information about the project ;

- http ://decision-deck.sourceforge.net

  Technical information about the D2 and D3 ;

- http ://www.decision-deck.org/d3/

  Portal of the D3 server in Luxembourg ;

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

  All information about the `XMCDA` *standard* ;

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

  All information on the diviz initiative.

- http ://leopold-loewenheim.uni.lu/cawa/

  Portal of the D4 server in Luxembourg.

Focus on three initiatives

- `XMCDA` *standard ;*
- *MCDA web services ;*
- *diviz.*

Focus on three initiatives

1. `XMCDA` *standard*

XMCDA : *Observations*

A standard data format does not exist to test a same MCDA problem instance on various methods (and softwares) ;

Existing MCDA methods / algorithms cannot *communicate*.

2007

Creation of the **specification committee** in Decision Deck to propose a standardised format for MCDA data : XMCDA.

XMCDA : *Introduction*

XMCDA is an instance of **UMCDA-ML**.

UMCDA-ML is intended to be a universal modelling language to express MCDA concepts and generic decision aid processes.

XMCDA focusses more particularly on MCDA **concepts** and **data structures** and is defined by an **XML schema**.

XMCDA : *Introduction*

The goals of XMCDA are to ease :

- the **interaction** of different MCDA algorithms ;
- the execution of various algorithms on the **same problem** instance ;
- the **visual representation** of MCDA concepts and data structures via standard tools like web browsers.

XMCDA is maintained by the specifications committee of the Decision Deck project.

XMCDA : *Introduction*

Abstract description of the XMCDA structure is performed via a detailed XML schema ;

See schema documentation for further details :
http ://www.decision-deck.org/xmcda

General idea : express MCDA concepts through a few general XML structures.

XMCDA : *Conventions*

- **MCDA concept** : a real or abstract construction related to the field of MCDA which needs to be stored in XMCDA ;

  for example, the importance of the criteria ;

- XMCDA **type** : XML structure that we created for the purpose of XMCDA ;

  for example, `criteriaValues` *to store general values related to a set of criteria*.

XMCDA : *Structure outline*

Several tags under the root element XMCDA.

A few general categories :

- Project or file description ;
- Output messages from methods (log or error messages) and input information for methods (options) ;
- Description of major MCDA concepts as attributes, criteria, alternatives, categories ;
- The performance table ;
- Further preferential information related to criteria, alternatives, attributes or categories.

XMCDA : *Conventions on the tagnames*

The name of a tag starts by a **lower-case** letter ;

The rest of the name is in mixed case with the first letter of each internal word capitalised ;

We use **whole words** and avoid as much as possible acronyms and abbreviations :

`methodParameters`, `performanceTable` and `preferenceInformation`

Objects of the same type can be gathered in a **compound** tag named after the plural form of its components (e.g., `alternatives`).

XMCDA : *Conventions on the attributes*

Three attributes can be found in the main data tags :
`id`, `name` and `mcdaConcept` ;

`id` : *machine readable* code or identifier of an object ;

```
<alternativesSet id="set1">
        <element>
                <alternativeID>a03</alternativeID>
        </element>
        <element>
                <alternativeID>a04</alternativeID>
        </element>
</alternativesSet>
```

XMCDA : *Conventions on the attributes*

`name` : *human-readable* name of an object

```
<parameter id="numIt" name="number of iterations">
        <integer>3</integer>
 </parameter>
```

`mcdaConcept` : MCDA type of a particular instance of an XMCDA structure

```
<alternativesSet mcdaConcept="kernel" name="a kernel
                              with two elements">
        <element>
                <alternativeID>a03</alternativeID>
        </element>
        <element>
                <alternativeID>a04</alternativeID>
        </element>
 </alternativesSet>
```

Do not mix up with the object's name ! !

XMCDA : *Elementary types – value*

```
<values>
        <value><integer>8</integer></value>

        <value><rankedLabel>
                <label>Good</label>
                <rank>1</rank>
        </rankedLabel></value>

        <value><rational>
                <numerator>10</numerator>
                <denominator>3</denominator>
        </rational></value>

       <value><real>3.141526</real></value>
</values>
```

Note that there also exists a type called `numericValue` which restricts `value` to numerical values.

XMCDA : *Elementary types – intervals, points & scales*

```
<interval>
        <lowerBound><value>[..]</value></lowerBound>
        <upperBound><value>[..]</value></upperBound>
</interval>
```

```
<point>
        <abscissa><real>2.7182818</real></abscissa>
        <ordinate><integer>23</integer></ordinate>
</point>
```

Scales can be qualitative, quantitative or nominal.

```
<scale>
        <quantitative>
                <min><real>0.00</real></min>
                <max><real>1.00</real></max>
        </quantitative>
</scale>
```

XMCDA : *Elementary types – functions*

A `function` can either be a constant, a linear, a piecewise linear function or simply a set of points.

```
<function>
        <constant><real>456.3847</real></constant>
</function>

<function>
        <linear>
                <slope><real>4.00</real></slope>
                <intercept><real>4.00</real></intercept>
        </linear>
</function>

<function>
        <points>[..]</points>
</function>
```

XMCDA : *Elementary types – description*

A description is present in any XMCDA type.

```
<alternatives>
       <description>
              <title>The list of alternatives</title>
              <comment>European cars
                    are considered.</comment>
       </description>
      [..]
<alternatives>
```

XMCDA : *How to describe the current project ?*

`projectReference` : description of the current project by different tags from the `description` type.

```
<projectReference id="testProblem">
      <version>1.2</version>
      <creationDate>2008-10-20T22:24:02</creationDate>
      <author>Patrick Meyer and Thomas Veneziano</author>
</projectReference>
```

XMCDA : *How to specify method-specific options ?*

Some methods require some specific options in order to guide the resolution of a decision problem.

```
<methodParameters>
        <approach>outranking</approach>
        <problematique>choice</problematique>
        <methodology>Rubis</methodology>
        <parameter name="variant">
              <value>
                 <label>standard</label>
              </value>
        </parameter>
</methodParameters>
```

XMCDA : *How to store method-specific messages ?*

Certain methods might generate some error or log messages.

```
<methodMessages>
        <errorMessage>
                <number>404</number>
                <name>Error 404</name>
                <message>
                        Data not found.
                        Did you specify a bad file name?
                </message>
        </errorMessage>
        <logMessage>
                <number>0</number>
                <name>OK</name>
                <message>Execution successful.</message>
        </logMessage>
</methodMessages>
```

XMCDA : *How to define alternatives ?*

```
<alternatives name="myAlternatives">
      <alternative id="x1" name="Red Ferrari"/>
      <alternative id="x2" name="Blue Corvette">
            <type>real</type>
            <active>true</active>
            <reference>false</reference>
      </alternative>
      <alternative id="x3" name="UFO">
            <type>fictive</type>
      </alternative>
 </alternatives>
```

XMCDA : *How to define criteria / attributes ?*

```
<criteria>
        <criterion id="g1">
                <description>
                        <comment>Power in horsepowers</comment>
                </description>
                <attributeReference>att1</attributeReference>
                <scale>
                        <quantitative>
                                <preferenceDirection>
                                        max
                                </preferenceDirection>
                                <minimum><real>50</real></minimum>
                                <maximum><real>200</real></maximum>
                        </quantitative>
                </scale>
        </criterion>
        <criterion id="g2"/>
 </criteria>
```

XMCDA : *How to define categories ?*

```
<categories>
      <category id="g" name="goodStudents">
            <active>true</active>
      <category>
      <category id="m" name="mediumStudents">
            <active>false</active>
      <category>
 </categories>
```

XMCDA : *The performance table*

```
<performanceTable>
      <alternativesPerformance>
            <alternativeID>alt1</alternativeID>
            <performance>
                  <criterionID>g1</criterionID>
                  <value><real>72.10</real></value>
                  </performance>
            <performance>
                  <criterionID>g2</criterionID>
                  <value><real>82.62</real></value>
            </performance>
      </alternativesPerformance>
      <alternativesPerformance>
            <alternativeID>alt2</alternativeID>
            [..]
      </alternativesPerformance>
</performanceTable>
```

XMCDA : *etc* ...

You've got the general ideas !

Also possible to store advanced preferential information on alternatives, criteria, attributes and categories.

For further details : http ://www.decision-deck.org/xmcda.

In particular, have a look at the *Quick guide to* XMCDA.

XMCDA : *time for a demo*

- An XMCDA instance ;
- XSD ;
- XSL + CSS : visualisation in a web browser.

XMCDA : *The specifications committee*

Maintenance of XMCDA & management of its future versions ;

Proposal of **evolutions**, according to needs expressed by users of XMCDA ;

Regular specifications meetings and discussions ;

Dissemination issues of the XMCDA releases ;

Forthcoming work on XMCDA ;

Don't hesitate to join us, if you're interested !

XMCDA : *Conclusion ?*

A few **general** types to represent a lot of concepts ;

Your participation is welcome ;

Some things are certainly missing ;

Try to implement your method and tell us what is wrong ;

General idea for programmers : **try to make compromises and be flexible** !

Focus on three initiatives

2. MCDA web services

MCDA web services

Observations :

- MCDA researchers are often not computer scientists ;
- MCDA researchers have programmed their algorithm(s) in the programming language they know best ;
- MCDA researchers are generally not interested in reimplementing their algorithm(s) in an *imposed* programming language.

MCDA web services

Observations :

- MCDA researchers are often not computer scientists ;
- MCDA researchers have programmed their algorithm(s) in the programming language they know best ;
- MCDA researchers are generally not interested in reimplementing their algorithm(s) in an *imposed* programming language.

MCDA web services

Observations :

- MCDA researchers are often not computer scientists ;
- MCDA researchers have programmed their algorithm(s) in the programming language they know best ;
- MCDA researchers are generally not interested in reimplementing their algorithm(s) in an *imposed* programming language.

MCDA web services

Raymond Bisdorff's idea (2007)

Instead of asking researchers to rewrite their MCDA algorithms in a specific programming language, allow them to publish their programs online s.t. they can be accessed over a network, as publicly available web services.

Consequences :

- Programming language independence (+);
- GUI-less :
  - Exclusive focus on the algorithmic part (+);
  - Harder to interact with the program (−);
- At any time, the latest version of the program (+).

MCDA web services

Raymond Bisdorff's idea (2007)

Instead of asking researchers to rewrite their MCDA algorithms in a specific programming language, allow them to publish their programs online s.t. they can be accessed over a network, as publicly available web services.

Consequences :

- Programming language independence (+);
- GUI-less :
  - Exclusive focus on the algorithmic part (+);
  - Harder to interact with the program (−);
- At any time, the latest version of the program (+).

MCDA web services

Raymond Bisdorff's idea (2007)

Instead of asking researchers to rewrite their MCDA algorithms in a specific programming language, allow them to publish their programs online s.t. they can be accessed over a network, as publicly available web services.

Consequences :

- Programming language independence (+) ;
- GUI-less :
  - Exclusive focus on the algorithmic part (+) ;
  - Harder to interact with the program (−) ;
- At any time, the latest version of the program (+).

MCDA web services

Raymond Bisdorff's idea (2007)

Instead of asking researchers to rewrite their MCDA algorithms in a specific programming language, allow them to publish their programs online s.t. they can be accessed over a network, as publicly available web services.

Consequences :

- Programming language independence (+) ;
- GUI-less :
  - Exclusive focus on the algorithmic part (+) ;
  - Harder to interact with the program (−) ;
- At any time, the latest version of the program (+).

MCDA web services

Raymond Bisdorff's idea (2007)

Instead of asking researchers to rewrite their MCDA algorithms in a specific programming language, allow them to publish their programs online s.t. they can be accessed over a network, as publicly available web services.

Consequences :

- Programming language independence (+);
- GUI-less :
  - Exclusive focus on the algorithmic part (+);
  - Harder to interact with the program (−);
- At any time, the latest version of the program (+).

MCDA web services

Raymond Bisdorff's idea (2007)

Instead of asking researchers to rewrite their MCDA algorithms in a specific programming language, allow them to publish their programs online s.t. they can be accessed over a network, as publicly available web services.

Consequences :

- Programming language independence (+);
- GUI-less :
  - Exclusive focus on the algorithmic part (+);
  - Harder to interact with the program (−);
- At any time, the latest version of the program (+).

MCDA web services

How to use the web services ?

Via various **client** softwares, like :

- D2 (via one of the plugins, called Rubis) ;
- D3 ;
- Command line (via a SOAP encapsulation) ;
- diviz.

What data is exchanged ?

`XML` files respecting the `XMCDA` standard !

MCDA web services

How to use the web services ?

Via various **client** softwares, like :

- D2 (via one of the plugins, called Rubis) ;
- D3 ;
- Command line (via a SOAP encapsulation) ;
- diviz.

What data is exchanged ?

`XML` files respecting the `XMCDA` standard !

MCDA web services

How to use the web services ?

Via various **client** softwares, like :

- D2 (via one of the plugins, called Rubis) ;
- D3 ;
- Command line (via a SOAP encapsulation) ;
- diviz.

What data is exchanged ?

`XML` files respecting the `XMCDA` standard !

MCDA web services

How to use the web services ?

Via various **client** softwares, like :

- D2 (via one of the plugins, called Rubis) ;
- D3 ;
- Command line (via a SOAP encapsulation) ;
- diviz.

What data is exchanged ?

`XML` files respecting the `XMCDA` standard !

MCDA web services

How to use the web services ?

Via various **client** softwares, like :

- D2 (via one of the plugins, called Rubis) ;
- D3 ;
- Command line (via a SOAP encapsulation) ;
- diviz.

What data is exchanged ?

`XML` files respecting the `XMCDA` standard !

MCDA web services

How to use the web services ?

Via various **client** softwares, like :

- D2 (via one of the plugins, called Rubis) ;
- D3 ;
- Command line (via a SOAP encapsulation) ;
- diviz.

What data is exchanged ?

`XML` files respecting the `XMCDA` standard !

MCDA web services

How to use the web services ?

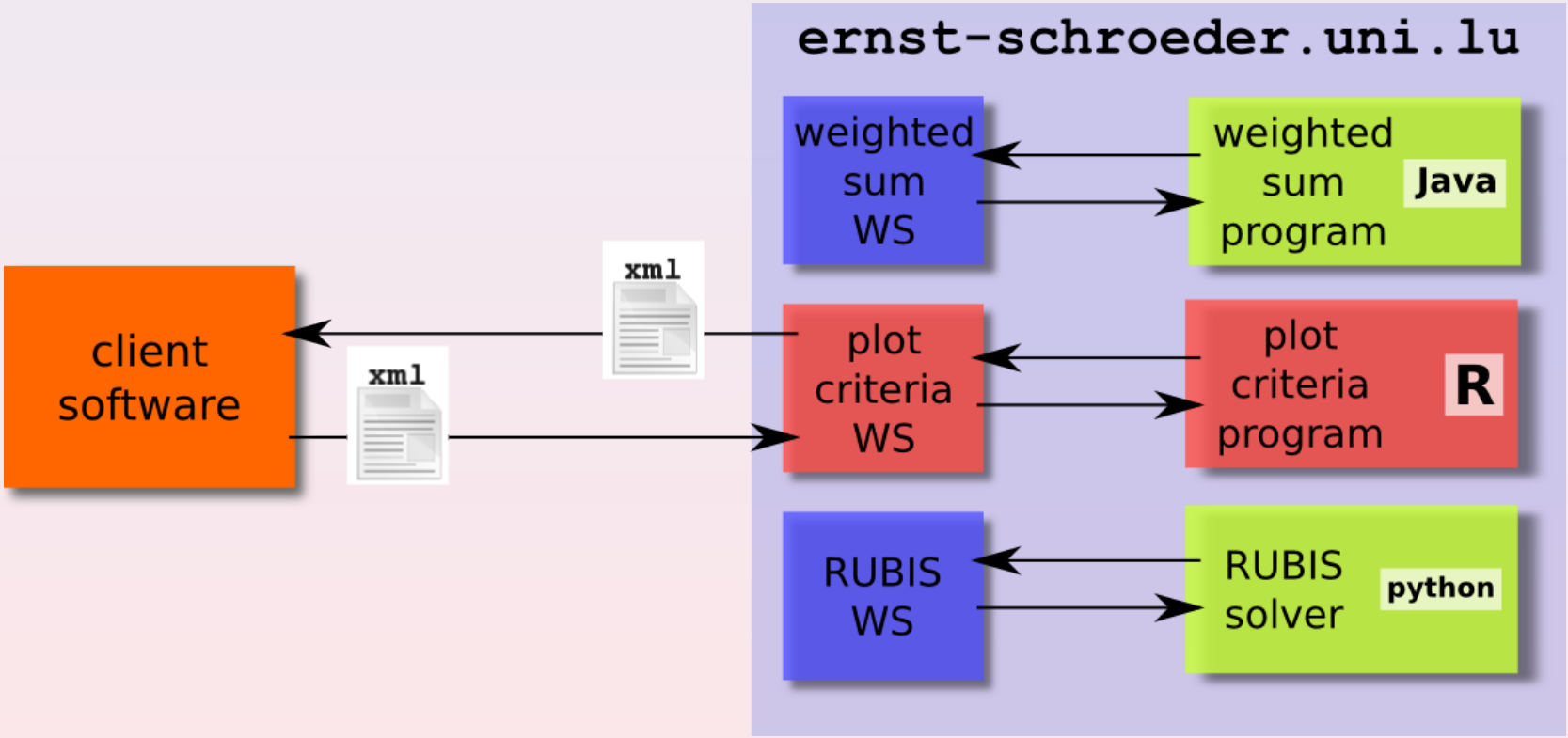
Via various **client** softwares, like :

- D2 (via one of the plugins, called Rubis) ;
- D3 ;
- Command line (via a SOAP encapsulation) ;
- diviz.

What data is exchanged ?

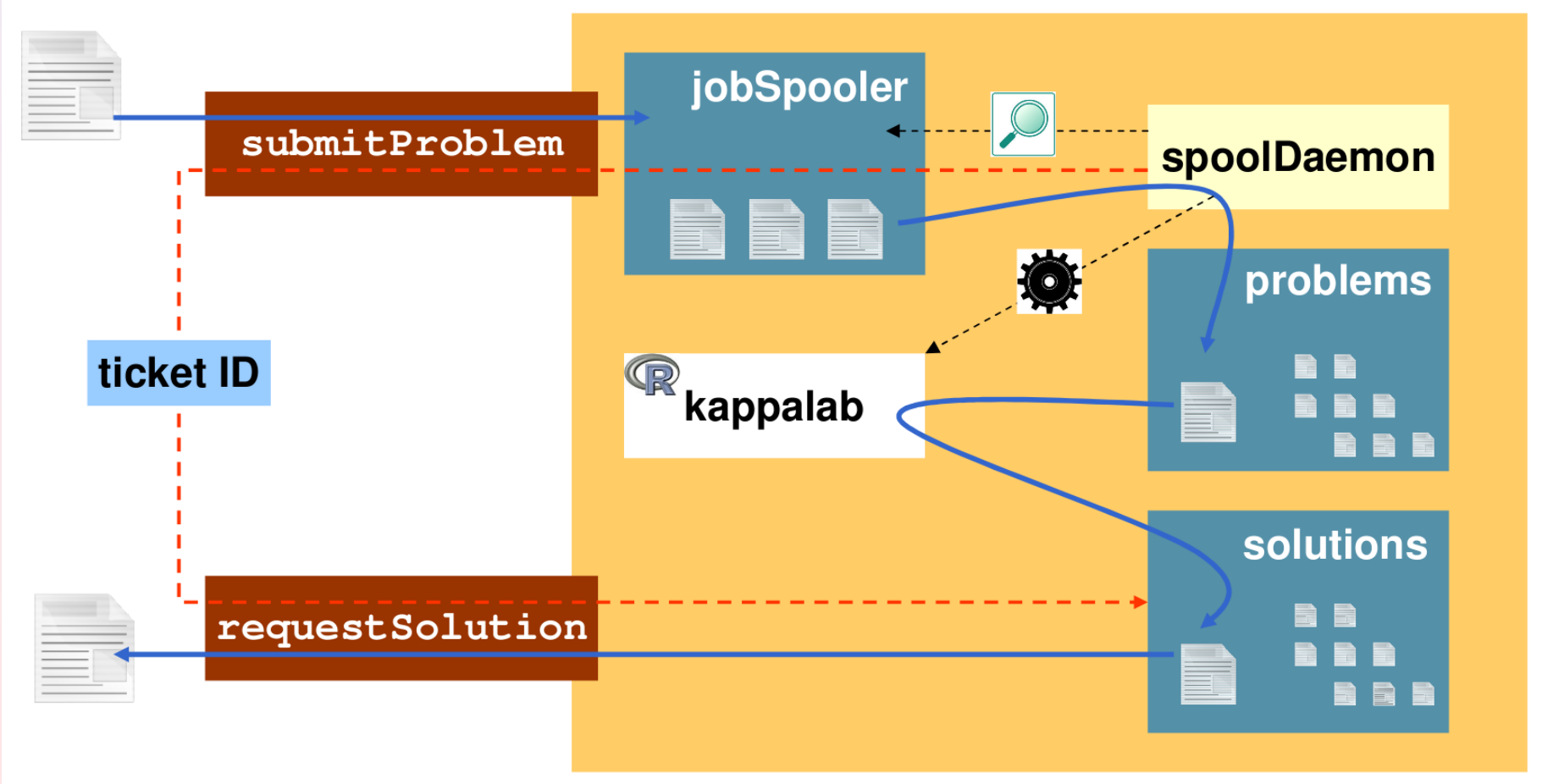
`XML` files respecting the `XMCDA` standard !

MCDA web services



MCDA web services

Web service architecture :



MCDA web services

Properties :

- Programming language independance
- Asynchronous
- Interoperable

MCDA web services

Properties :

- Programming language independance

  Nearly any GUI-less program can be run behind the WS ;

  Java, Python, C, C++, Perl, ..., R, ...

- Asynchronous

  Useful in case the calculations are time consuming ;

- Interoperable

  The output of a WS can be reinjected into another WS

Properties :

- Programming language independance

  Nearly any GUI-less program can be run behind the WS ;

  Java, Python, C, C++, Perl, . . ., R, . . .

- Asynchronous

  Useful in case the calculations are time consuming ;

- Interoperable

  The output of a WS can be reinjected into another WS ;

MCDA web services

Properties :

- Programming language independance

    Nearly any GUI-less program can be run behind the WS ;

    Java, Python, C, C++, Perl, . . ., R, . . .

- Asynchronous

    `submitProblem & requestSolution`

    Useful in case the calculations are time consuming ;

- Interoperable

    The output of a WS can be reinjected into another WS

MCDA web services

Properties :

- Programming language independance

  Nearly any GUI-less program can be run behind the WS ;

  Java, Python, C, C++, Perl, ..., R, ...

- Asynchronous

  `submitProblem` & `requestSolution`

  Useful in case the calculations are time-consuming ;

- Interoperable

  The output of a WS can be reinjected into another WS

MCDA web services

Properties :

- Programming language independance
  - Nearly any GUI-less program can be run behind the WS ;
    - Java, Python, C, C++, Perl, ..., R, ...
- Asynchronous
  - `submitProblem` & `requestSolution`
    - Useful in case the calculations are time-consuming ;
- Interoperable
  - The output of a WS can be reinjected into another WS

MCDA web services

Properties :

- Programming language independance
  - Nearly any GUI-less program can be run behind the WS ;
    - Java, Python, C, C++, Perl, ..., R, ...
- Asynchronous
  - `submitProblem` & `requestSolution`
    - Useful in case the calculations are time-consuming ;
- Interoperable
  - The output of a WS can be reinjected into another WS.

MCDA web services

Properties :

- Programming language independance
  - Nearly any GUI-less program can be run behind the WS ;
    - Java, Python, C, C++, Perl, ..., R, ...
- Asynchronous
  - `submitProblem` & `requestSolution`
    - Useful in case the calculations are time-consuming ;
- Interoperable
  - The output of a WS can be reinjected into another WS.

MCDA web services

Properties :

- Programming language independance
  - Nearly any GUI-less program can be run behind the WS ;
    - Java, Python, C, C++, Perl, ..., R, ...
- Asynchronous
  - `submitProblem` & `requestSolution`
    - Useful in case the calculations are time-consuming ;
- Interoperable
  - The output of a WS can be reinjected into another WS.

Focus on three initiatives

3. diviz

diviz

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 and workflows ;
- allow to easily add new MCDA components ;
- avoid heavy calculations on your local computer by executing the methods on distant servers ;

diviz

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 and workflows ;
- allow to easily add new MCDA components ;
- avoid heavy calculations on your local computer by executing the methods on distant servers ;

diviz

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 and workflows ;
- allow to easily add new MCDA components ;
- avoid heavy calculations on your local computer by executing the methods on distant servers ;

diviz

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 and workflows ;
- allow to easily add new MCDA components ;
- avoid heavy calculations on your local computer by executing the methods on distant servers ;

diviz

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 and workflows ;
- allow to easily add new MCDA components ;
- avoid heavy calculations on your local computer by executing the methods on distant servers ;

diviz

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 and workflows ;
- allow to easily add new MCDA components ;
- avoid heavy calculations on your local computer by executing the methods on distant servers ;

diviz

Properties :

- all components are (opensource) **web services** ;
- history of past executions ;
- use of XMCDA to make elementary components interoperable ;
- use of XMCDA + XSL for a standardised representation of input and output data.

diviz

Properties :

- all components are (opensource) **web services** ;
- **history** of past executions ;
- use of `XMCDA` to make elementary components **interoperable** ;
- use of `XMCDA` + XSL for a standardised **visualisation** of input and output data.

diviz

Properties :

- all components are (opensource) **web services** ;
- **history** of past executions ;
- use of `XMCDA` to make elementary components **interoperable** ;
- use of `XMCDA` + XSL for a standardised **visualisation** of input and output data.

diviz

Properties :

- all components are (opensource) **web services** ;
- **history** of past executions ;
- use of `XMCDA` to make elementary components **interoperable** ;
- use of `XMCDA` + XSL for a standardised **visualisation** of input and output data.

diviz

Properties :

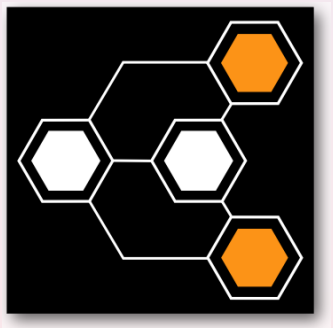
- all components are (opensource) **web services** ;
- **history** of past executions ;
- use of `XMCDA` to make elementary components **interoperable** ;
- use of `XMCDA` + XSL for a standardised **visualisation** of input and output data.

diviz

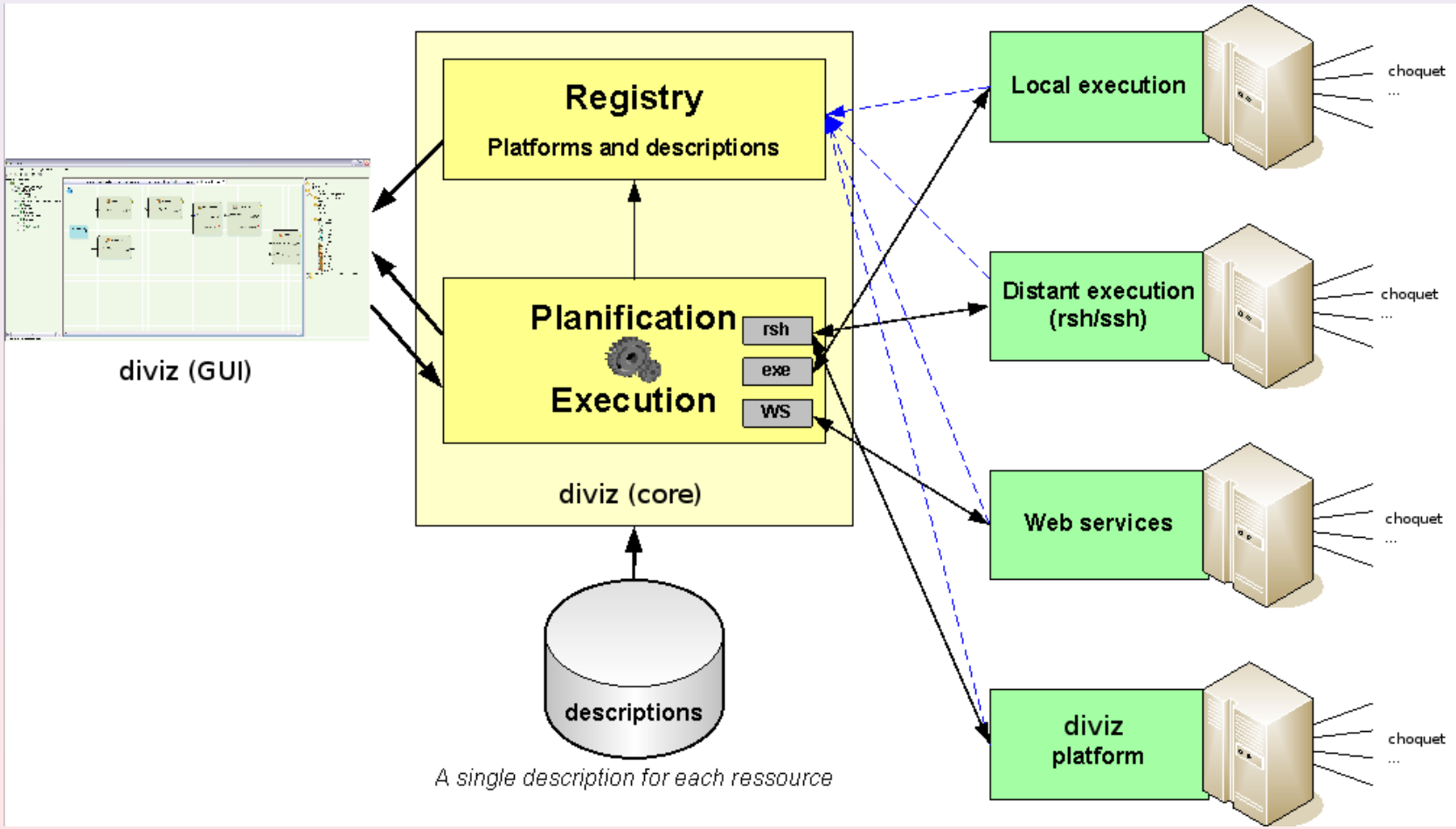
The name ?

diviz means *decision* in Breton ...  ...

A live demo



diviz : Architecture



diviz : Architecture

A generic framework driven by programs' descriptions only !

Key points :

- Different deployment configurations ;
- Execution engine :
  - XML-based resources' description :
    - name, type,
    - domain of validity,
    - inter-dependencies,
    - I/O are typed.

diviz : Architecture

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 :
  - Parse types ;
  - Check validity ;
  - Inter-dependencies ;
  - I/O are typed.

diviz : Architecture

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 :
  - Parse type ;
  - Check validity ;
  - Inter-dependencies ;
  - I/O are typed.

diviz : Architecture

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, type,
  - domain of validity,
  - inter-dependencies,
  - I/O are typed.

diviz : Architecture

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.

diviz : Architecture

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.

diviz : Architecture

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.

diviz : Architecture

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.

diviz : Architecture

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.

diviz : Architecture

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.

diviz : Architecture

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.

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

- 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

- 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

- 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

- 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

- 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

- 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 role manager.

What diviz is **not**

- A decision aid process designer and manager ;
- A role manager.

The future & what you can do.

How you can help the project

- Join the Decision Deck Consortium
  (contact me at `patrick.meyer@telecom-bretagne.eu`) ; or,
- Support our project (development, standardisation, . . .) ; or,
- Test the software solutions & let us know your opinion.

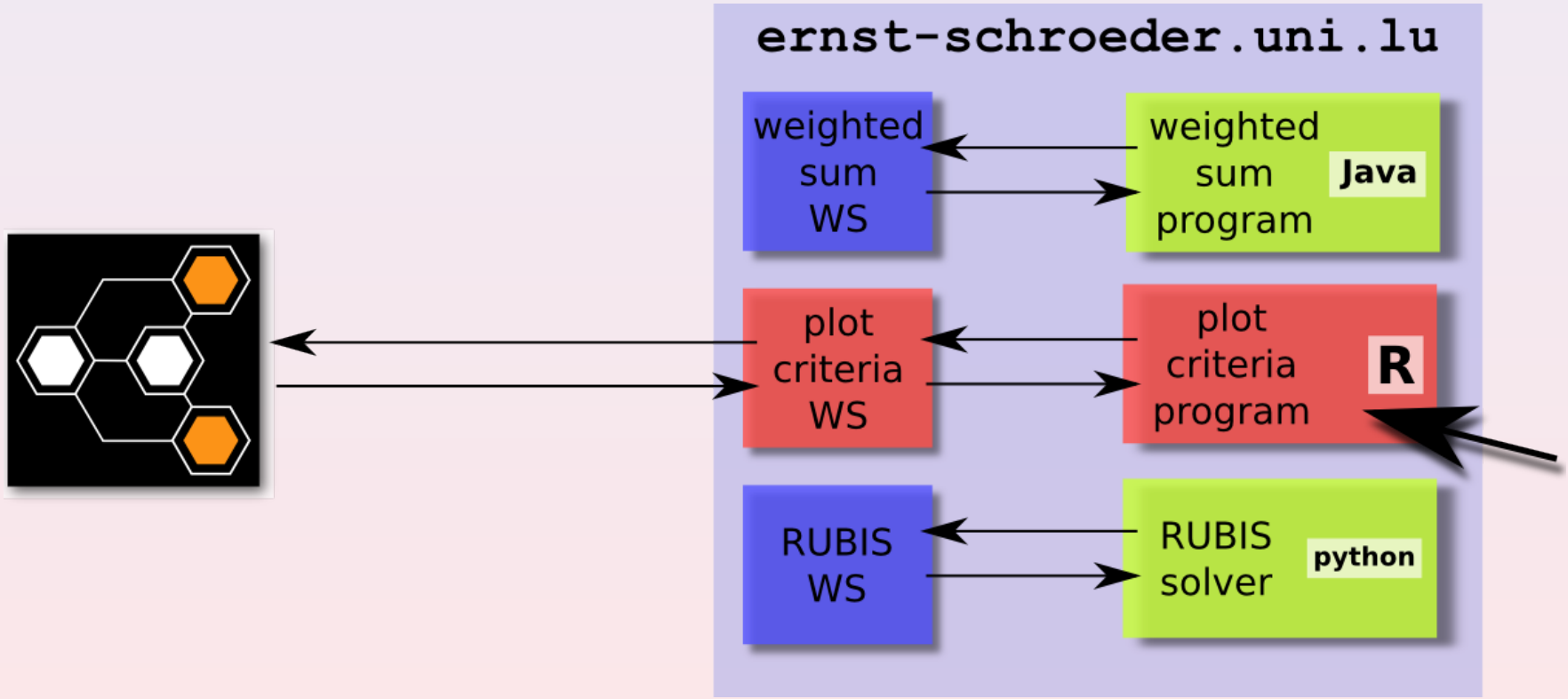
How you can help the project

- Join the Decision Deck Consortium
  (contact me at `patrick.meyer@telecom-bretagne.eu`) ; or,
- Support our project (development, standardisation, ...) ; or,
- Test the software solutions & let us know your opinion.

How you can help the project

- Join the Decision Deck Consortium
  (contact me at `patrick.meyer@telecom-bretagne.eu`) ; or,
- Support our project (development, standardisation, ...) ; or,
- Test the software solutions & let us know your opinion.

Developping web services



WS architecture, independent from diviz.

What you have to do to develop a web service (with integration into diviz)

Rough **recipe** :

- **Determine** the `XMCDA` data types that your command line program needs ;
- **Adapt** your program to read and write `XMCDA` files ;

  existing R library & Python library !

- 2 input parameters for your program :
- Specify the mandatory and optional input and output data files and `XMCDA` data types ;
- Send us the program with the specifications.

See also `http://www.decision-deck.org/diviz` for detailed instructions.

What you have to do to develop a web service (with integration into diviz)

Rough **recipe** :

- **Determine** the `XMCDA` data types that your command line program needs ;
- **Adapt** your program to read and write `XMCDA` files ;

  existing R library & Python library !

- 2 input parameters for your program :
  - input data directory ;
  - output data directory ;
- Specify the mandatory and optional input and output data files and `XMCDA` data types ;
- Send us the program with the specifications.

See also `http://www.decision-deck.org/diviz` for detailed instructions.

What you have to do to develop a web service (with integration into diviz)

Rough **recipe** :

- **Determine** the `XMCDA` data types that your command line program needs ;
- **Adapt** your program to read and write `XMCDA` files ;

  existing R library & Python library !

- 2 input parameters for your program :
  - Input data directory ;
  - Output data directory ;
- Specify the mandatory and optional input and output data files and `XMCDA` data types ;
- Send us the program with the specifications.

See also `http://www.decision-deck.org/diviz` for detailed instructions.

What you have to do to develop a web service (with integration into diviz)

Rough **recipe** :

- **Determine** the `XMCDA` data types that your command line program needs ;
- **Adapt** your program to read and write `XMCDA` files ;

  existing R library & Python library !

- 2 input parameters for your program :
  - Input data directory ;
  - Output data directory ;
- Specify the mandatory and optional input and output data files and `XMCDA` data types ;
- Send us the program with the specifications.

See also `http://www.decision-deck.org/diviz` for detailed instructions.

What you have to do to develop a web service (with integration into diviz)

Rough **recipe** :

- **Determine** the `XMCDA` data types that your command line program needs ;
- **Adapt** your program to read and write `XMCDA` files ;

  existing R library & Python library !

- 2 input parameters for your program :
  - Input data directory ;
  - Output data directory ;
- **Specify** the mandatory and optional input and output data files and `XMCDA` data types ;
- Send us the program with the specifications.

See also `http://www.decision-deck.org/diviz` for detailed instructions.

What you have to do to develop a web service (with integration into diviz)

Rough **recipe** :

- **Determine** the `XMCDA` data types that your command line program needs ;
- **Adapt** your program to read and write `XMCDA` files ;

  existing R library & Python library !

- 2 input parameters for your program :
  - Input data directory ;
  - Output data directory ;
- **Specify** the mandatory and optional input and output data files and `XMCDA` data types ;
- **Send** us the program with the specifications.

See also `http://www.decision-deck.org/diviz` for detailed instructions.

What you have to do to develop a web service (with integration into diviz)

Rough **recipe** :

- **Determine** the `XMCDA` data types that your command line program needs ;
- **Adapt** your program to read and write `XMCDA` files ;

  existing R library & Python library !

- 2 input parameters for your program :
  - Input data directory ;
  - Output data directory ;
- **Specify** the mandatory and optional input and output data files and `XMCDA` data types ;
- **Send** us the program with the specifications.

See also `http://www.decision-deck.org/diviz` for detailed instructions.

What you have to do to develop a web service (with integration into diviz)

Rough **recipe** :

- **Determine** the `XMCDA` data types that your command line program needs ;
- **Adapt** your program to read and write `XMCDA` files ;

  existing R library & Python library !

- 2 input parameters for your program :
  - Input data directory ;
  - Output data directory ;
- **Specify** the mandatory and optional input and output data files and `XMCDA` data types ;
- **Send** us the program with the specifications.

See also `http ://www.decision-deck.org/diviz` for detailed instructions.

What you have to do to develop a web service (with integration into diviz)

Rough **recipe** :

- **Determine** the `XMCDA` data types that your command line program needs ;
- **Adapt** your program to read and write `XMCDA` files ;

  existing R library & Python library !

- 2 input parameters for your program :
  - Input data directory ;
  - Output data directory ;
- **Specify** the mandatory and optional input and output data files and `XMCDA` data types ;
- **Send** us the program with the specifications.

See also `http ://www.decision-deck.org/diviz` for detailed instructions.

What you have to do to develop a web service (with integration into diviz)

Rough **recipe** :

- **Determine** the `XMCDA` data types that your command line program needs ;
- **Adapt** your program to read and write `XMCDA` files ;

  existing R library & Python library !

- 2 input parameters for your program :
  - Input data directory ;
  - Output data directory ;
- **Specify** the mandatory and optional input and output data files and `XMCDA` data types ;
- **Send** us the program with the specifications.

See also `http ://www.decision-deck.org/diviz` for detailed instructions.

How to stay informed ?

Low traffic informational mailing list of the Decision Deck project :

https ://mlistes.telecom-bretagne.eu/wws/subscribe/decisiondeck-info

Low traffic informational mailing list of the diviz software :

https ://mlistes.telecom-bretagne.eu/wws/subscribe/diviz-announcements