



UNIVERSITAT
ROVIRA I VIRGILI

Departament d'Enginyeria Informàtica i Matemàtiques

Intelligent Decision Support tools in Decision Deck

TITULACIÓ: Enginyeria informàtica

AUTOR: Jordi Canals Ros.

DIRECTORS: Aïda Valls.

Lucas Marín.

DATA: 06 / 2011.

Contents

1 – INTRODUCTION AND OBJECTIVES	5
2 – BACKGROUND AND PROBLEM SPECIFICATION	7
2.1 – THREE BASIC CONCEPTS	8
2.1.1 – <i>Alternative</i>	8
2.1.2 – <i>Criterion</i>	9
2.1.3 – <i>Decision problem</i>	9
2.2 – DIFFERENT APPROACHES TO MCDA	11
2.2.1 – <i>Multiple Attribute Utility Theory (MAUT)</i>	11
2.2.2 – <i>Outranking methods</i>	12
2.2.3 – <i>Decision rule approach</i>	14
2.3 – OVERVIEW OF THE DECISION DECK PROJECT	15
2.3.1 – <i>XMCD</i> A	17
2.3.2 – <i>Decision Desktop (D^2)</i>	17
2.3.3 – <i>Distributed Decision Deck (D^3)</i>	19
2.3.4 – <i>Diviz</i>	20
2.3.4.1 - Workflow design	20
2.3.4.2 - Execution and results	21
2.3.4.3 - Workflow sharing	22
2.4 – PROBLEM SPECIFICATION	23
3 – NEW COMPONENTS	25
3.1 – ORDERED WEIGHTED AVERAGE	25
3.1.1 – <i>Introduction</i>	25
3.1.2 – <i>Characterize a weight vector</i>	27
3.1.3 – <i>Families of OWA operators</i>	30
3.1.4 – <i>Example</i>	31
3.1.5 – <i>Final remarks</i>	33
3.2 – ULOWA	34
3.2.1 – <i>Introduction</i>	34
3.2.2 – <i>The unbalanced linguistic ordered weighted average</i>	37
3.2.3 – <i>Properties of the ULOWA operator</i>	38
3.2.4 – <i>Example</i>	39
4 – DESIGN DECISIONS	41
4.1 – INTRODUCTION	41
4.2 – SHORT DESCRIPTION OF BASIC USAGE OF DIVIZ	42
4.3 – HOW TO CREATE A MODULE FOR DIVIZ	44
4.4 – ORDERED WEIGHTED AVERAGE	47

4.4.1 – <i>State diagram</i>	47
4.4.2 – <i>Class diagram</i>	49
4.4.3 – <i>XMCDa inputs diagram</i>	50
4.4.3.1 – <i>Alternative</i>	50
4.4.3.2 – <i>Criteria</i>	51
4.4.3.3 – <i>Performance table</i>	52
4.4.3.4 – <i>Weights</i>	53
4.4.4 – <i>XMCDa output diagram</i>	54
4.4.4.1 – <i>OWA</i>	54
4.4.5 – <i>OWA module</i>	55
4.4.6 – <i>Example</i>	55
4.5 – UNBALANCED LINGUISTIC ORDERED WEIGHTED AVERAGE	57
4.5.1 – <i>State diagram</i>	58
4.5.2 – <i>Class diagram</i>	59
4.5.3 – <i>XMCDa inputs diagram</i>	60
4.5.3.1 – <i>fuzzyNumber</i>	60
4.5.4 – <i>XMCDa output diagram</i>	61
4.5.4.1 – <i>ULOWA</i>	61
4.5.5 – <i>ULOWA module</i>	62
4.6 – HOW TO DOCUMENT THE PROGRAM	62
4.6.1 – <i>OWA documentation file</i>	66
4.6.2 – <i>ULOWA documentation file</i>	68
5 – CONCLUSIONS	73
BIBLIOGRAPHY.....	75

1 – Introduction and Objectives

Humans face daily complex decision making problems in many diverse fields. This issue is especially critical in business or governmental decisions, because it may involve large amounts of money. Making decisions is often a complex task that must consider multiple criteria or attributes simultaneously. The frequency and complexity of decision making requires the development of multidisciplinary methods that help the decision maker.

This has led to the design of various methods of multi-criteria decision making (MCDA: multi-criteria decision aid), that propose different solutions to help analyse and weight the criteria, evaluate the alternatives properly (even considering uncertainty), and finally find best decision for each problem. Often these methods have been implemented with specific software and are available only by their creators. This hinders both the researchers in this area because it is not easy to compare the results of new methods with existing ones or integrate parts of different methods, and the field of applications, because using an existing method requires someone to implement it again.

Some methods have been commercialized by creating new companies, which offer the assistance to clients in order to solve a decision problem using their particular methodology and software. The software is usually not sold, or it has really high prices. Some companies are SEAS, Decision Lab.

For this reason, the project Decision Deck has been created. The Decision Deck project aims at collaboratively developing Open Source software tools implementing a platform composed of modular and interconnected software components. These software components implement the common functionalities of a large range of MCDA methods. It is being implemented as a free software that contains a library of methods for multi-criteria decision making with a very modular structure. Although different tools have been defined, nowadays efforts are mainly focused on the Diviz platform implementation, which is based on web services that communicate each other using a specific language called XMCD (an extension of XML). It provides a user-friendly interface that facilitates the interconnection and execution of the web services.

The objectives of this project are:

- Studying the tools offered by Decision Deck, particularly Diviz:
 - Identify what methods are available.
 - Making use of the system Diviz with a simple example.
 - Prepare a manual for use in English.
- Studying the language XMCD: *(Note: The original text contains a typo 'XMCD' which has been corrected to 'XMCD' based on the context of the document.)*
 - Identify the structure and basic elements.
 - Learn how you can add new elements to existing ones.
 - Propose new structures that allow XMCD to support fuzzy sets and linguistic labels.
- Developing new methods in Diviz, in particular focusing on linguistic labels.

The structure of this document is as follows. In chapter 2, we explain the main concepts regarding Multiple Criteria Decision Making (MCD), as well as, a review of the Decision Deck project and its main components. We will focus on the Diviz module, which is the newest one, and it is particularly designed for academic purposes, as well as, to facilitate the reuse and comparison of MCD modules.

Two different methodologies will be analysed in detail: the Weighted Sum Model (based on Utility Theory) and ELECTRE III (based on Outranking). User guide indication will be given, in order to facilitate the use of Diviz to test those methods with different data sets.

2 – Background and Problem Specification

The area of Multiple Criteria Decision Aid (MCDA) is focused on providing solutions to decision problems where many criteria and preferences must be taken into consideration. Many important technical aspects of MCDA are linked to classic works in economics, in particular, welfare economics, utility theory and voting oriented social choice theory. Aggregating the opinion or the preferences of voters or individuals of a community into collective or social preferences is quite a similar problem to devising comprehensive preferences of a decision-maker from a set of conflicting criteria in MCDA.

In the field of Multi-criteria Decision Aid a large number of methodologies that are acknowledged in the research community and used in real world decision processes have been proposed. These methodologies provide useful solutions to complex decision problems that arise in organizations. Several methods have been implemented independently during the years in an uncoordinated way, using different language and software tools. This software is rarely available, or it has become obsolete because it is not maintained.

In the international literature on management science and operations research there is an increasing number of real-world applications of MCDA. Of course, in the limited space here it would be impossible to provide an extensive bibliography regarding these applications. However, it is possible to outline some of the most significant areas to which MCDA has contributed. The following list reports some of these areas:

1. Finance and economics:
 - a. Business failure prediction.
 - b. Credit risk assessment.
 - c. Portfolio selection and management.
 - d. Company mergers and acquisitions.
 - e. Financial planning.

- f. Country risk evaluation.
 - g. Regional economic policy specification.
- 2. Environmental Management and Energy planning:
 - a. Forest management.
 - b. Waste management.
 - c. Power plant siting.
 - d. Water resources planning.
 - e. Nuclear power management.
 - f. Energy intensity evaluation.
- 3. Marketing:
 - a. Customer satisfaction.
 - b. Design of market penetration strategies.
 - c. Retail evaluation.
- 4. Transportation:
 - a. Highway planning.
 - b. Subway design.
- 5. Human resources management:
 - a. Job evaluation.
 - b. Personnel selection.
- 6. Education.
- 7. Agriculture.

2.1 – Three basic concepts

First of all, and for the sake of clarity of this document, in this chapter we define the basic concepts of MCDA problems, following the nomenclature in (Figueira, J.; Greco, S.; Ehrott, M., 2005): alternatives, criteria and decision problem.

2.1.1 – Alternative

The general concept of action is used to designate the element which constitutes the object of the decision, or the one which decision aiding is directed towards. The concept of action does not incorporate a priori any notion of feasibility, or possible implementation. An action is qualified as potential when it is deemed possible to

implement it, or simply when it deserves some interest within the decision aiding process.

The concept of alternative corresponds to the particular case in which two distinct potential actions can in no way be conjointly put into operation. Many authors implicitly suppose that potential actions are, by definition, mutually exclusive. Although this hypothesis is in no way compulsory, we will assume it in the rest of the document.

In all cases, A will denote the set of alternatives considered at a given stage of the decision process. By a , we will denote any single alternative. In this document we will assume that the number of actions is finite ($|A| = m$) having:

$$A = \{a_1, a_2, \dots, a_m\}$$

2.1.2 – Criterion

A criterion c is a tool constructed for evaluating and comparing alternatives according to a particular point of view. This evaluation must take into account, for each action a , all the pertinent effects or attributes linked to the point of view considered. Sometimes criteria can be understood as goals to be achieved by the best alternative.

A criterion is denoted by $c(a)$ and represents the performance of the alternative a according to some aspect of the problem. This performance can be interpreted as the degree of satisfaction of the decision maker with respect to a certain variable (feature, property).

The set of criteria will be always finite ($|C| = n$) and denoted as:

$$C = \{c_1, c_2, \dots, c_n\}$$

The scale of measurement of a criterion can be numerical or qualitative. For more information on data types, we refer the reader to (Torra & Narukawa, 2007).

2.1.3 – Decision problem

In MCDA, the definition of the type of decision problem deals with answers to questions such as the following (Figueira, J.; Greco, S.; Ehrott, M., 2005):

1. In what terms should we pose the problem?

2. What type of results should we try to obtain?
3. How does the analyst see himself fitting into the decision process to aid in reaching these results?
4. What kind of procedure seems the most appropriate for guiding his investigation?

Decision problems are traditionally classified into three types as follows:

- The *choice* problematic: The aid is oriented towards and lies on a selection of a small number (as small as possible) of “good” actions in such a way that a single alternative may finally be chosen. This does not mean that the selection is necessarily oriented towards the determination of one or all the actions of A which can be regarded as optimum. The selection procedure is based on comparisons between actions so as to justify the elimination of the greatest number of them, the subset of those actions which are selected (which can be viewed as a first choice) containing all the most satisfying actions, which remain non comparable between one another.
- The *sorting* problematic: The aid is oriented towards and lies on an assignment of each action to one category (judged the most appropriate) among those of a family of predefined categories, which are usually ordered according to some preference consideration. This family of categories must be conceived on the basis of the diverse types of treatments that motivate the sorting. For instance, a family of four categories can be based on a comprehensive appreciation leading to distinguishing between: actions for which implementation (i) is fully justified, (ii) could be advised after only minor modifications, (iii) can only be advised after major modifications, (iv) is unadvisable.
- The *ranking* problematic: The aid is oriented towards and lies on a complete or partial preorder of alternatives in A which can be regarded as an appropriate instrument for comparing actions between one another; this preorder is the result of a classifying procedure allowing us to put together in classes actions which can be judged as indifferent, and to rank these classes (some of them may remain non-comparable).

2.2 – Different approaches to MCDA

In multiple-criteria decision problems, each action (alternative) a from a given set A is represented by a vector of evaluations on particular criteria and a crucial point is the aggregation of the vector evaluations so as to say if action x is at least as good as action y , which corresponds to the weak preference relation $x \geq y$. Passing from vector evaluations to the weak preference relation requires the use of a preference model of a specific Decision-Maker (DM). This model can be either a utility function, or a more general (fuzzy, outranking) relation defined on X , or a set of “if..., then...” decision rules. The selection of the most adequate preference model is a crucial task that must be decided on advance.

2.2.1 – Multiple Attribute Utility Theory (MAUT)

This approach is based on the concept of utility function, called also value function. It transforms the vector evaluation into a scalar value $f(\cdot)$, such that $x \geq y$ if and only if $f(x) \geq f(y)$. Using this preference model one can establish a straightforward recommendations based on the Multi-Attribute Utility Theory (MAUT).

According to MAUT, the overall evaluation $f(x)$ of an object x is defined as a weighted addition of its evaluation with respect to its relevant performance criteria. For example, a digital camera can be evaluated on criteria like quality of image, flash, viewfinder, operation time, and handling. The overall evaluation can be simply found with the following aggregation function:

$$f(x) = \sum_{i=1}^n w_i f_i(x)$$

Here, $f_i(x)$ is the utility associated to the value of the alternative on the i -th criterion and w_i the weight determining the impact of the i -th criterion on the overall evaluation (also called the relative importance of the criterion), n is the number of different value dimensions, and $\sum_{i=1}^n w_i = 1$.

The use of different aggregation operators is extensive in MCDA methods based Multi-Attribute Utility Theory, such as the geometric or harmonic average. The

comparison of those ratings is used to solve the selection, ranking or classification decision problem.

2.2.2 – Outranking methods

The concept of outranking relations was born with the intention to overcome some of the difficulties of the aggregation approaches based on MAUT. For example, MAUT methods are based on the concept of dominance relation and cannot deal with other types of relations such as incomparability. Moreover, the use of ordinal criteria is difficult in MAUT, but very natural in outranking.

This approach focuses the attention to the fact that in MCDA problems one tries to establish preference orderings of alternatives (Roy, B., 1991) (Perny, P.; Roy, B., 1992). As each criterion usually leads to different ranking of the alternatives, the problem is to find a consensus ranking.

One of the most well-known outranking methods is ELECTRE (ELimination Et Choix Traduisant la Réalité). Using the definition in (Roy, 1974), an outranking relation is a binary relation S defined in X such that xSy if, given what is known about the decision-maker's preferences and given the quality of the valuations of the actions and the nature of the problem, there are enough arguments to decide that x is at least as good as y , while there is no essential reason to refute that statement.

The technique uses factual and/or subjective information to assess simultaneously a set of alternatives A under different evaluation criteria C , that can be homogeneous, heterogeneous, quantified, qualified or mixture thereof. The way is through the use of preference relations, which compare the assessments $c(a)$ partially assigned to each alternative a , identifying the best through a process of overrating.

The ELECTRE methods identify four preference situations concerning the comparison of two actions, using binary outranking relations, S , whose meaning is “*at least as good as*” (Roy, B., 1996):

- Indifference(I): it corresponds to a situation where there are clear and positive reasons that justify equivalence between the two actions (it leads to a reflexive and symmetric binary relation).

- Ex: xSy and ySx, xIy (x is indifferent to y).
- Strict Preference(P): it corresponds to a situation where there are clear and positive reasons in favour of one (identified) of the two actions (it leads to a no reflexive and asymmetric binary relation).
 - Ex: xSy and not ySx, xPy (x is strictly preferred to y).
 - Ex: ySx and not xSy, yPx (y is strictly preferred to x).
- Weak Preference(Q): it corresponds to a situation where there are clear and positive reasons that invalidate strict preference in favour of one (identified) of the two actions, but they are insufficient to deduce either the strict preference in favour of the other action or indifference between both actions, thereby not allowing either of the two preceding situations to be distinguished as appropriate (it leads to a no reflexive and asymmetric binary relation).
- Incomparability(R): it corresponds to an absence of clear and positive reasons that would justify any of the three preceding relations (it leads to a no reflexive and symmetric binary relation).
 - Ex: Not xSy and not ySx, xRy (x is incomparable to y).

The construction of an outranking relation is based on two major concepts:

1. Concordance: For an outranking xSy to be validated, a sufficient majority of criteria should be in favour of this assertion.
2. Discordance: When the concordance condition holds, none of the criteria in the minority should oppose too strongly to the assertion xSy .

From the concordance and discordance relations, a Credibility index is calculated, which corresponds to the value of “outranking” of x with regards to y . Different ways of implementing these relations lead to different versions of *ELECTRE*.

- **ELECTRE I**: it is the first method of outranking published. It is focused on solving Choice problems. It uses only strong preferences (not weak preference). The method reduces the number of alternatives forming a core of the best ones.
- **ELECTRE II**: it is a method also of outranking but more elaborated theoretically than the previous one. It allows obtaining a complete arrangement of the not dominated alternatives.

- **ELECTRE III:** in this method the relation of outranking is based on strong and weak preferences, defining different thresholds for pseudo-criteria.
- **ELECTRE IV:** adapted for cases in which the decision maker does not want to specify the preferential weight.

2.2.3 – Decision rule approach

This model studies how people make decisions by searching for rules which provide good justification of their choices. In this way, the preference model is given in terms of “*if..., then ...*” rules. Since there may be difficult to define such kind of rules by the decision maker, they are usually extracted (learned) from a set of exemplary decisions. Examples of such rules, regarding to the preferences and the performances of the values, are the following:

- “*If maximum speed of car x is at least 175 km/h and its price is at most \$12000, then car x is comprehensively at least medium*”.
- “*If car x is at least weakly preferred to car y with respect to acceleration and the price of car x is no more than slightly worse than that of car y , then car x is at least as good as car y* ”.

The rules induced from exemplary decisions represent a preferential attitude of the DM and enable the understanding of the reasons of her preference. The acceptance of the rules by the DM justifies, in turn, their use for decision support.

The set of decision rules accepted by the decision maker can be applied to a set of alternatives in order to obtain specific preference relations. From the exploitation of these relations, a suitable recommendation can be obtained to support the decision maker in the decision problem at hand.

So, the preference model in the form of decision rules induced from examples fulfils both representation and recommendation tasks.

The induction of rules from examples is a typical approach of artificial intelligence. This explains our interest in rough set theory which proved to be a useful tool for analysis of vague description in decision situations. The aim of rough set analysis is the explanation of the dependence between the values of some decision

attributes, playing the role of “dependent variables”, by means of the values of other condition attributes, playing the role of “independent variables”.

In (Greco, Matarazzo, & Slowinski, Rough sets theory for multi-criteria decision analysis, 2001) the Dominance-based Rough Sets Approach was defined. It is an extension of the classic rough sets model to deal with preference scales. So, this is the most appropriate for decision making problems, where the preference information is crucial and must be kept into the rules generated.

2.3 – Overview of the Decision Deck project

The Decision Deck project aims at collaboratively developing Open Source software tools¹ implementing Multiple Criteria Decision Aid. 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.

As stated before, the goal of the Decision Deck project is to offer an integrated platform for using the available MCDA methods developed, facilitating the testing and comparison of those methods. In addition, they offer tools to incorporate new decision methods, whatever the language in which they are implemented, without having to rewrite them. This is achieved by means of Decision Deck plug-in programming, which must fulfil some specification requirements. This design makes possible to easily integrate additional methods without great effort.

To achieve these goals, the Decision Deck project works on developing multiple software resources that are able to interact. Along the history of the project, different tools have been developed (Decision Deck, 2009):

¹ <http://www.decision-deck.org/>

- **XMCD A**: a standardized XML recommendation to represent objects and data structures issued from the field of MCDA. Its main objective is to allow different MCDA algorithms to interact and be easily callable.
- **XMCD A web services**: distributed open source computational MCDA resources, using the XMCD A standard.
- **DIVIZ**: an open source Java client and server for designing, executing and sharing MCDA methods, via the composition of XMCD A web services.
- **D²**: a rich open source Java software containing several MCDA methods.
- **D³**: an open source rich internet application for XMCD A web services management.



In order to coordinate the various activities of the Decision Deck project, it is structured as follows:

- **The Decision Deck Consortium**: a French non-profit association which steers and manages the project along the lines of this manifesto. It is headed by an administration board. The consortium is among other in charge of organising the workshops of the project every semester.
- **The Software Resources Management Groups**: they are in charge of the organisation and the management of the developments of the six identified initiatives of the Decision Deck project. Each group is coordinated by a clearly identified contact person. These management groups are in charge of organising the bi-annual developers' days of the project.
- **The Specifications Committee**: under the direction of a coordinator, its role is to maintain and develop the XMCD A standard and to approve and publish

suggested evolutions. The coordinator is in charge of organising the specifications meetings.

- **The Communication & Dissemination Committee:** under the direction of a coordinator, its role is to develop and maintain the websites of Decision Deck and to manage the communicational aspects of the project.

2.3.1 – XMCD

XMCD is a data standard which allows representing Multi Criteria Decision Analysis (MCDA) data elements in XML according to a clearly defined grammar.

XMCD is an instance of UMCDA-ML, which is the Universal Multi Criteria Decision Analysis Modelling Language and which is one of the scientific initiatives inside the Decision Deck project. UMCDA-ML is intended to be a universal modelling language to express MCDA concepts and generic decision aid processes.

XMCD focuses particularly on MCDA concepts and data structures and is defined by an XML schema.

XMCD is maintained by the Decision Deck project and supported by the COST Action IC0602 Algorithmic Decision Theory.

The goals of XMCD 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.

A very natural field of application of XMCD is given by the web services developed inside the Decision Deck project.

2.3.2 – Decision Desktop (D²)

The Decision Desktop software, or D² for short, was the first software to be developed in the Decision Deck project.

It is a desktop, client/server application, meaning that it is designed to be installed locally (it is not a web application), and uses a database to store application data, thereby enabling a multiple user usage.

One of its usage patterns is that several experts may enter evaluations in a decentralized manner, then these evaluations are analysed by a coordinator, this analysis is reviewed by one or several decision-makers.

Three hard-coded roles have been identified:

- **Coordinator:** The Coordinator has the responsibility to specify the model.
- **Decision Maker:** The Decision Maker has the responsibility of choosing an alternative.
- **Evaluator:** The evaluator has the responsibility of evaluating alternatives

Historically the software has also been named EVAL, from the name of the originating project, and Decision Deck. It is now recommended to name it decision desktop to differentiate it from the other software produced in the Decision Deck project.

D² offers several MCDA methods, like:

- **Iris** is an interactive method that aims at sorting alternatives into ordered classes. The assignment of alternatives to classes corresponds to ELECTRE Tri. The specificity of the IRIS method is that the decision maker does not have to specify the weights of criteria, but instead provides assignment examples, i.e., typical elements of the classes (C. Dias & Mousseau, IRIS: A DSS for Multiple Criteria Sorting Problems, 2003) (Damart, C. Dias, & Mousseau, 2006) (C. Dias, L.; Mousseau, V.; Figueira, J.; Clímaco, J., 2002).
- **Rubis** is a decision aid method for tackling the choice problem in the context of multiple criteria decision analysis (Bisdorff, Meyer, & Roubens, RUBIS: a bipolar-valued outranking method for the choice problem, 2008).

Its genuine purpose is to help a decision maker to determine a single best decision alternative. Methodologically we focus on pairwise comparisons of these alternatives which lead to the concept of bipolar-valued outranking digraph (Bisdorff, Pirlot, & Roubens, Choies and kernels in bipolar valued digraphs,

2006). The work is centred on a set of five pragmatic principles which are required in the context of a progressive decision aiding methodology (Meyer, 2007).

- The **VIP** (Variable Interdependent Parameters) Analysis software has been built to support the selection of the most preferred alternative among a list, considering the impacts of each alternative on multiple evaluation criteria. It is based on an additive aggregation model (value function), accepting imprecise information on the value of the scaling coefficients (*e.g.*, scaling constants, which indirectly reflect the relative importance of the each criterion) (C. Dias & Clímaco, Additive Aggregation with Variable Interdependent Parameters: the VIP Analysis Software, 2000).
- The **UTAGMS** and **GRIP** methods aim at solving the multiple criteria ranking problem. The underlying preference model is a set of monotone additive value function compatible with preference statements expressed by the decision maker. The main result of these methods is:
 - a necessary (robust) ranking that contains comparisons of alternatives that remain valid for all compatible value function,
 - a possible ranking that contains comparisons of alternatives that is valid for at least one compatible value function.

For detailed descriptions of the methods, see:

- UTAGMS (Greco, Mousseau, & Słowiński, Ordinal regression revisited:multiple criteria rankingusing a set of additive value functions, 2008).
- GRIP (Rui Figueira, J.; Greco, S.; Słowiński, R., 2009).
- Weighted Sum.

One of the advantages is that we do not need to be connected to the Internet to work because the user has a local database to store all its methods. However, this complicates the sharing of work.

2.3.3 – Distributed Decision Deck (D³)

D³ is a distributed and collaborative approach to Multi Criteria Decision Aid. It is a Rich Internet Application that allows interacting with remote web services exposing MCDA methods. They have a served (Distributed Decision Deck) resource that permits

to integrate such existing methods. Data exchanges between D³ and remote MCDA services are done through XML files using a SOAP literal RPC protocol.

In the previous section we have seen that the user must have its own database, which stores all its methods, this is a drawback, because nobody else can use them. The distributed version, as discussed below, provides the ability to share methods among the community, forcing to use Internet.

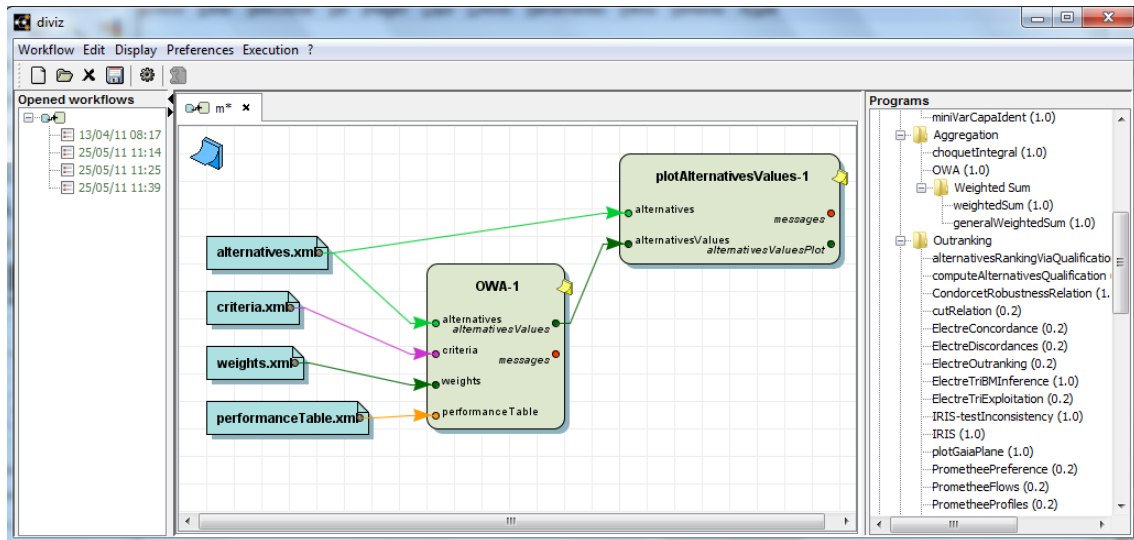
2.3.4 – Diviz

Diviz enables to conveniently combine programs implementing MCDA algorithms in a modular way, decomposing them into subsystems. It is an easy tool to build, execute and share complex workflows of MCDA algorithms. In the literature, such workflows are often called methods (Bigaret & Meyer, diviz: an MCDA workflow design, execution and sharing tool).

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

2.3.4.1 - Workflow design

The design of the MCDA workflows is performed via an intuitive graphical user interface, where each algorithm is represented by a box that 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 needs understanding the functioning of each calculation module.



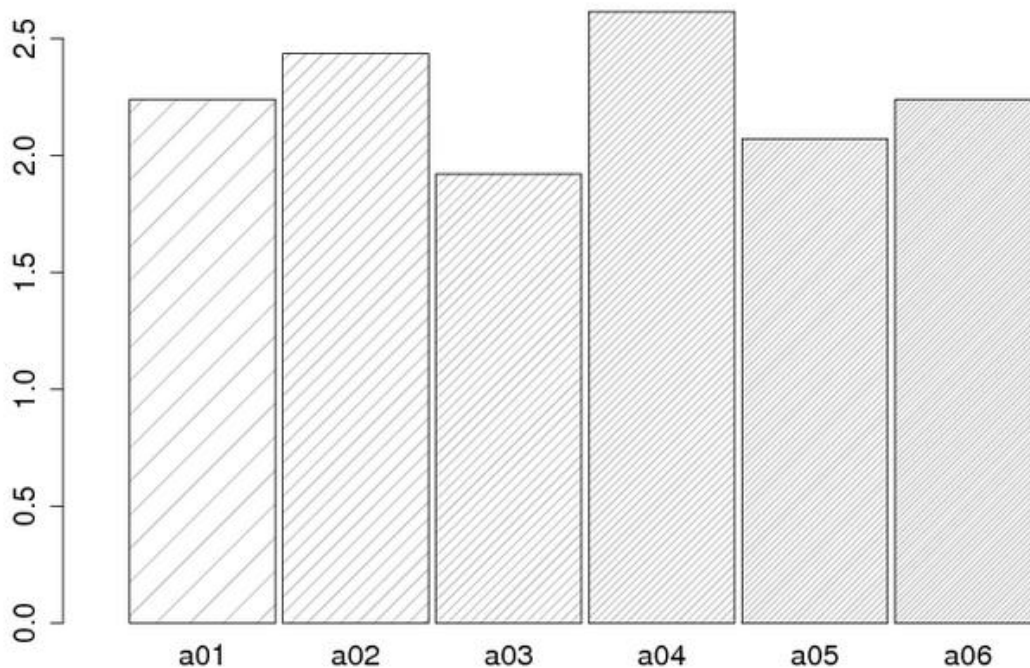
To construct a new MCDA workflow, the user chooses one or more modules from a list of available calculation elements that he can drag and drop in a dedicated workspace. Then he adds data files to the workspace and connects them appropriately to the inputs of the elements. Finally, the user connects the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

2.3.4.2 - Execution and results

Once the design of the MCDA workflow is finished, it is possible to execute it in order to obtain the output values of the algorithms. In *Diviz*, these calculations are performed on computing servers through the use of web-services 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 can be viewed and analysed by the user. Some of these outputs might represent results of intermediate calculation steps of the decision aid workflow that has been built by the user.

a01, a02, a03, a04, a05, a06



Furthermore, a historic of past executions is kept in *Diviz* and can be viewed by the user. In particular, if a workflow is modified, the former execution results and their associated workflows are still available. Again this is a key feature which helps calibrating the parameters of the algorithms which are used in a workflow, as any past execution can be recalled and reanalysed.

2.3.4.3 - Workflow sharing

The *Diviz* software enables to export any workflow, with or without the data, as an archive. The workflow can then be shared with any *Diviz* user, who can then 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* is especially interesting for academic purposes, because the teacher can prepare workflows that can be used by the students. In the same way, the students' workflows and results can be evaluated by the teacher.

In addition, *Diviz* provides some interesting tools for research, as the results or workflows obtained with these tools can become supplementary electronic material to attach to research papers.

2.4 – Problem specification

The first objective of this work has been achieved with the preliminary work on the study of the basic concepts and state of the art of MCDA and Decision Deck. It has been presented in this chapter, in the previous section. In addition, a report has been published in the *Departament d'Enginyeria Informàtica i Matemàtiques* de la URV (Canals, Valls, & Marín, 2011). It can be found in the Annex to this document.

This report contains an introduction to Decision Deck and especially to Diviz. It also provides a user guide for Diviz and some exemplary workflows for two MCDA methods. Those materials have been used this year in the course called SAD (*Sistemes d'Ajut a la Decisió*) which is given in *Enginyeria Informàtica* and *Màster de Seguretat i Sistemes Intel·ligents* at URV.

Now, we present the specification of the problem that will be studied in the rest of the document, which is the focus of this PFC. It corresponds to objectives 2 and 3.

Following the MAUT approach that is focused on aggregating the performances of the different criteria into a single overall value (i.e. global utility), we will study the the OWA (Ordered Weighted Averaging). OWA operators are particularly interesting because they permit to define an aggregation policy that introduces a degree of andness/orness (i.e. conjunction or simultaneity versus disjunction or replaceability). Different extensions of the basic definition of OWA have been defined in the literature.

In order to study the behaviour of OWA with respect to other averaging operators, we have developed a new component for the Decision Deck platform. We have some experience in the use of Diviz because it is particularly suitable to study the behaviour of different MCDA techniques on the same dataset. So, we want to extend DIVIZ including some operators from the family of OWA.

Problem specification:

- Design and implementation of a module for Diviz for the classic OWA operator that works with numerical values.
 - Inputs/outputs
- Design and implementation of a module for Diviz for a new method called ULOWA (Unbalanced Linguistic OWA) that works with an

ordered set of linguistic terms and considers a fuzzy membership function associated with the terms.

- XMCDa extension
- Inputs/outputs

3 – New components

3.1 – Ordered Weighted Average

3.1.1 – Introduction

Ordered Weighted Averaging Operators (OWA) obtains from a set of data values a single representative of them. The particularity of OWA operators is that the representative value is obtained in accordance with predetermined parameters of optimism/pessimism. Thus, each decision maker can aggregate the information on his own manner according to their degree of optimism or pessimism. That is, from a mathematical point of view, we can say that OWA operators are used to average the information within the limits defined by the minimum and maximum. In order to model this behavioural policy of the aggregation procedure, a set of weights is attached to the values according to their degree of performance. By this way, the values are first ordered from the best to the worst, and a weight is assigned to the positions of the values. In short, the decision maker has to establish a weight for each position of an ordered vector of values.

This operator was defined by R.R. Yager in (Yager, On ordered weighted averaging aggregation operators in multi-criteria decision making, 1988). Since its appearance, it has been studied by many authors and it has been widely applied to many decision making problems (Herrera, Herrera-Videman, & Verdegay, Direct approach processes in group decision making using linguistic OWA operators, 1996) (Merigó & Gil-Lafuente, The induced generalized OWA operator, 2009) (Xu, 2006) (Beliakov, Pradera, & Calvo, 2007).

Definition: A function $F: R^n \rightarrow R$ is an OWA operator of dimension n if it has an associated vector W of dimension n such that its components satisfy:

- a. $w_j \in [0,1]$
- b. $\sum_{j=1}^n w_j = 1$

And:

$$F(a_1, a_2, \dots, a_n) = \sum_{j=1}^n w_j b_j$$

Where b_j is the j -th largest element of the bag $\langle a_1, a_2, \dots, a_n \rangle$.

Notice that the fundamental aspect of this operator is the re-ordering step, in particular an argument a_i is not associated with a particular weight w_i but rather a weight is associated with a particular ordered position of argument.

An OWA operator is an average. This reflects the fact that the operator has the following properties (Yager, OWA aggregation over a continuous interval argument with applications to decision making, 2004):

1. **Symmetry:** For any permutation π , we have $F(a_{\pi(1)}, a_{\pi(2)}, \dots, a_{\pi(n)}) = F(a_1, a_2, \dots, a_n)$.
2. **Monotonicity:** If $a_i \geq d_i$ for all i , then $F(a_1, a_2, \dots, a_n) \geq F(d_1, d_2, \dots, d_n)$
3. **Boundedness:** $\min\{a_i\} \leq F(a_1, a_2, \dots, a_n) \leq \max\{a_i\}$.
4. **Idempotency:** If $a_i = a$ for all i , then $F(a_1, a_2, \dots, a_n) = F(a_1, a_2, \dots, a_n)$

The following simple example illustrates the use of the OWA operator.

Example: Assume F is an ordered weighting averaging operator of size $n = 4$ with weighting vector,

$$W = (0.2, 0.3, 0.1, 0.4)$$

- a) Calculate $F(0.6, 1, 0.3, 0.5)$.

In this case the ordered argument vector B is:

$$B = (1, 0.6, 0.5, 0.3)$$

Hence,

$$F(1, 0.6, 0.5, 0.3) = 0.2 * 1 + 0.3 * 0.6 + 0.1 * 0.5 + 0.4 * 0.3 = 0.55$$

- b) Calculate $F(0, 0.7, 0.1, 0.2)$.

Here,

$$B = (1, 0.7, 0.2, 0.0)$$

And therefore,

$$F(0,0.7,0.1,0.2) = 0.43$$

3.1.2 – Characterize a weight vector

As the set of weights is extremely important in the OWA method, because it determines the aggregation policy that the decision maker is imposing on the decision process, some measures have been introduced to characterize a weight vector. The first measure associated with an OWA aggregation with weighting vector W is evaluating its **attitudinal-character** (or **orness**), which is defined as (Yager, On ordered weighted averaging aggregation operators in multi-criteria decision making, 1988):

$$\alpha(W) = \sum_{j=1}^n w_j \left(\frac{n-j}{n-1} \right)$$

It is known that $\alpha \in [0,1]$. As a general rule, as the allocation of weight in W moves to the top, then α gets closer to one, meanwhile as the weights move to the bottom, α gets closer to zero. Furthermore, if W is symmetrical, then $\alpha(W) = 0.5$. This measure provides a characterization of the type of aggregation being performed. An α value near one indicates a bias toward considering mainly the larger values in the argument (i.e. high **orness** or disjunctive behaviour), while an α value near zero indicates preference is being given to the smaller values in the argument (i.e. high **andness** or conjunctive behaviour). A α value near 0.5 is an indication of a neutral type aggregation (i.e. averaging).

Another concept, introduced for Yager (Yager, On ordered weighted averaging aggregation operators in multi-criteria decision making, 1988), associated with the OWA weights is the **entropy**, which we define as:

$$H(W) = - \sum_{j=1}^n w_j \ln(w_j)$$

We note if W is such that $w_j = 1/n$ for all j , $H(W) = \ln(n)$, then the entropy is maximum. If W is such $w_q = 1$ and $w_j = 0$ for all $j \neq q$, then $H(W) = 0$, which implies that the information used is minimal.

Another measure to study the vector W is the degree of favouritism toward optimistic or pessimistic values, which is known as **Balance operator** (Yager, Quantifier guided aggregation using OWA operators, 1996), which we define as:

$$Bal(W) = \sum_{j=1}^n \frac{(n+1-2j)}{n-1} w_j$$

As we can see, $Bal(W) \in [-1,1]$:

- $Bal(W) = -1 \rightarrow$ Minimum operator.
- $Bal(W) = 1 \rightarrow$ Maximum operator.
- $Bal(W) = 0 \rightarrow$ Arithmetic mean.

In general, for values close to 1, the aggregation is giving more weight to larger values. On the other hand, for values close to -1, means that aggregation is giving greater importance to small values. However, for values near 0, we can say that the vector W used, is giving importance to both maximum and minimum values.

A fourth measure to study the vector W is the degree of divergence W (Yager, Heavy OWA Operators, 2002). It is said that this measure is useful for cases where the average entropy and attitudinal-character are not enough to define the policy of aggregation.

$$Div(W) = \sum_{j=1}^n w_j \left(\frac{n-j}{n-1} - \alpha(W) \right)^2$$

For the optimistic and pessimistic case, $Div(W) = 0$. We can also say that if $w_j = 1$ for some j , then $Div(W) = 0$.

We will show a situation where this measure could be useful. Suppose $n = 9$ and we have two vectors W and W^* .

$$W = (0,0.5,0,0,0,0,0.5,0)$$

$$W^* = (0,0,0,0.5,0,0.5,0,0,0)$$

Results:

- $\alpha(W) = \alpha(W^*) = 0.5$
- $H(W) = H(W^*) = \ln(2)$
- $Bal(W) = Bal(W^*) = 0$

Therefore, these three measures do not allow us to distinguish between the vectors of weights. However, with the fourth measure, it is possible to distinguish between these vectors.

Result:

- $Div(W) = 0.140625$
- $Div(W^*) = 0.015625$

That is, the vector W is giving weights to more different values than W^* , notice that the gap between positions 4 and 6 in W^* is smaller than the difference between positions 2 and 8 in W .

Now, we will propose examples to show how the formulas work with the following weight vectors. Is important to remember that the vector of arguments associated to the weights will always be sorted from highest to lowest values.

$$W_1 = (0.1, 0.1, 0.1, 0.2, 0.2, 0.3)$$

Let us see the outcome of this vector:

- $\alpha(W_1) = 0.36 \rightarrow$ Values near to zero indicate that our weight vector attaches importance to low values. This indicates Andness behaviour.
- $H(W_1) = 1.69 \rightarrow$ For higher values, we can say that it uses a lot of information.
- $Bal(W_1) = -0.28 \rightarrow$ For values under 0, we can say that the vector W used is giving importance to minimum values.

$$W_2 = (0.4, 0.4, 0.2, 0, 0, 0)$$

Results:

- $\alpha(W_2) = 0.84 \rightarrow$ With this result we can say that our vector give more importance to high values.
- $H(W_2) = 1.05 \rightarrow$ In this second vector, we see that the result is still quite high, and we can say that it uses a lot of information but not all.
- $Bal(W_2) = 0.68 \rightarrow$ This last value makes it clear that our system is biased to the highest values. We could say that our system will be optimistic.

3.1.3 – Families of OWA operators

In this section we will study a wide range of special cases of OWA operators that can be given when setting the weight vector W . As it is evident, by choosing a different configuration of the weight vector, we can get different types of aggregations (Merigó, Nuevas extensiones a los operadores OWA y su aplicación en los métodos de decisión., 2008).

1. **The optimistic approach:** It only considers the best value and discards the rest of information.

$$\left\{ \begin{array}{l} w_1 = 1 \\ w_j = 0, \quad \forall j \neq 1 \end{array} \right\} \Rightarrow F(a_1, \dots, a_n) = \text{Max}\{a_j\}$$

2. **The pessimistic approach:** Requires that all the values are good to consider that globally the alternative is good. It does not allow a compensation of low values with high values. It is implemented as the minimum, because if the minimal value is high, it means that the others are equal or even better.

$$\left\{ \begin{array}{l} w_n = 1 \\ w_j = 0, \quad \forall j \neq n \end{array} \right\} \Rightarrow F(a_1, \dots, a_n) = \text{Min}\{a_j\}$$

3. **Arithmetic mean,** Gives equal contribution to each argument, making an average between them. So high values are compensated with low ones. This reduces to the classical behaviour of aggregation operators.

$$\{w_j = 1/n, \quad \forall j\} \Rightarrow F(a_1, \dots, a_n) = (1/n) \sum_{j=1}^n a_j$$

4. **Criterion of Realism** (or *Hurwicz*): The results is obtained as a compromise between optimistic and a pessimistic decision.

$$\left\{ \begin{array}{l} w_1 = \alpha \\ w_n = (1 - \alpha) \\ w_j = 0 \quad \forall j \neq 1, n \end{array} \right\} \Rightarrow F(a_1, \dots, a_n) = \alpha \text{Max}\{a_j\} + (1 - \alpha) \text{Min}\{a_j\}$$

5. **Step-OWA operator:** We only care about one of the input values. If $k = 1$ the operator becomes maximum, and if $k = 0$ minimum (Yager, Families of OWA operators, 1993).

$$\left\{ \begin{array}{l} w_k = 1 \\ w_j = 0, \quad \forall j \neq k \end{array} \right\} \Rightarrow F(a_1, \dots, a_n) = a_k$$

6. **Window-OWA operator:** Window type operators, is characterized by using two parameters, k and m to determine the weights of the aggregation. Those two parameters define a “window” or “subset” of values on the set of arguments. This parameters are defined as follows (Yager, Families of OWA operators, 1993):

$$w_j = \left\{ \begin{array}{ll} 0 & \text{si } j < k \\ 1/m & \text{si } k \leq j < k + m \\ 0 & \text{si } j \geq k + m \end{array} \right\} \Rightarrow F(a_1, \dots, a_n) = \frac{1}{m} \sum_{j=k}^{k+m-1} a_j$$

It should be noted that k and m are integers positive such that $k + m \leq n + 1$, where n represents the size of weight vector.

7. **Olympic-OWA operator:** an Olympic OWA operator computes the arithmetic average of all aggregates except the smallest and the largest ones.

$$\left\{ \begin{array}{l} w_1 = 0 \\ w_n = 0 \\ w_j = \frac{1}{n-2} \quad \forall j \neq 1, n \end{array} \right\} \Rightarrow F(a_1, \dots, a_n) = \frac{(\sum_{j=1}^n a_j) - \text{Min}\{a_j\} - \text{Max}\{a_j\}}{n-2}$$

Where $n \geq 3$.

3.1.4 – Example

An example where the reader can observe the performance of OWA operators in decision-making process is developed below. Notice that these operators are applicable to any decision-making problem.

The example is focused on investment selection but it can be generalized to a variety of situations, such as the selection of human resources, products, homes, etc.

Suppose that an investment company is faced with five possible investments and has to select one that best suits its needs. This goal is interpreted as obtain the greatest possible benefit from the investment.

1. A_1 : invest in a company car.
2. A_2 : invest in a food company.
3. A_3 : invest in a computer company.
4. A_4 : invest in a chemical company.
5. A_5 : invest in a business of television.

The expert committee of the company sets the expected benefits of each investment. As it is a very uncertain environment, these results are conditional on different states (S_k) that could occur in the future.

The expected results for each investment are:

	S_1	S_2	S_3	S_4	S_5	S_6
A_1	30	30	50	70	20	30
A_2	40	80	-10	70	100	-50
A_3	10	0	20	40	10	150
A_4	30	40	40	50	20	50
A_5	60	-30	70	70	50	10

To evaluate the results that can be obtained according to different aggregation policies in OWA; we will study 5 different configuration of OWA: optimistic behaviour, pessimistic behaviour, arithmetic mean, criterion of realism and finally the execution of OWA with the following set of weights $W = (0.1, 0.1, 0.1, 0.2, 0.2, 0.3)$.

The results obtained with those 5 policies are:

	optimistic	pessimistic	Arithmetic Mean	Criterion of Realism	OWA
A_1	70	20	38.33	38	33
A_2	100	-50	38.33	30	16
A_3	150	0	38.33	58	25
A_4	50	20	38.33	40	34
A_5	70	-30	38.33	37	23

The best alternative for each case is indicated in a grey cell. As you can see the choice is different depending on the operator used. If we use the optimistic, the weighted average (WA), our decision will be choosing the A_3 . If we use the pessimist, the decision will be choosing the A_1 or A_4 . Using the set of weight W , we will choose the A_4 . Finally, using the arithmetic mean (AM), all investments are equally valid.

Then, we will study the other types of OWA aggregations related to limit the aggregation to a subset of values, not directly related to a conjunctive/disjunctive view of the problem. This includes the step-OWA operator, Window-OWA operator, and the Olympic-OWA operator. For each case we have the following weight vectors:

- Step-OWA operator with $k = 3$:

$$W = (0,0,1,0,0,0)$$

- Window-OWA operator with $m = 3$ and $k = 2$:

$$W = \left(0,0,\frac{1}{3},\frac{1}{3},0,0\right)$$

- Olympic-OWA operator:

$$W = \left(0,\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4},0\right) = (0,0.25,0.25,0.25,0.25,0)$$

Results obtained through these types of OWA operators are:

	Step-OWA	Window-OWA	Olympic-OWA
A_1	30	20	35
A_2	70	40	45
A_3	20	10	20
A_4	40	26.667	40
A_5	60	36.667	47.5

In this case, again the best alternative is different for each case.

3.1.5 – Final remarks

We can conclude that the determination of the weighting policy is crucial in the Ordered Weighted Average operator, because it has a great influence in the final result.

This versatility and flexibility are advantages if the decision maker is able to clearly specify the aggregation policy that is needed in each decision problem.

In general, the classical arithmetic mean operator is applied in aggregation processes, without taking into account that this model is quite restrictive. Briefly, we can find the following list of drawbacks of additive aggregation models is given (J. Dujmovic, 2007):

1. Limited number of components for evaluation.
2. Evaluation effort wasted.
3. Impossible to model mandatory requirements.
4. Impossible to model simultaneity of requirements.
5. Impossible to model sufficient conditions.
6. Impossible to model asymmetric logic relationships.
7. Incorrect optimum systems.
8. Evaluation effort imbalance.

3.2 – ULOWA

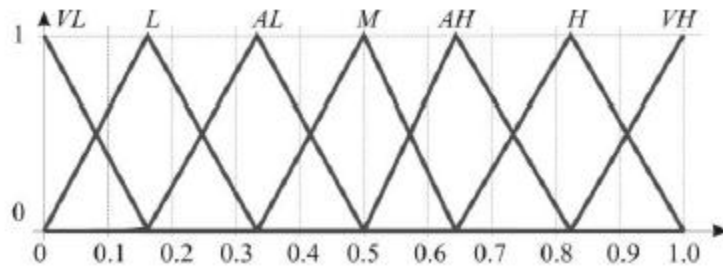
3.2.1 – Introduction

In this section, we will concentrate on the aggregation of linguistic terms, which is part of many other procedures, such as decision making (Torra & Narukawa, 2007).

In the last decades, various linguistic aggregation operators have been proposed, including linguistic median operator (Yager, Fusion of ordinal information using weighted median aggregation, 1988), linguistic weighted median operator (Yager, Fusion of ordinal information using weighted median aggregation, 1988), median-based aggregation operator (Domingo-Ferrer & Torra, 2003), 2-tuple weighted averaging operator (Herrera & Martínez, A 2-Tuple fuzzy linguistic representation model for computing with words, 2000), 2-tuple OWA operator (Herrera & Martínez, A 2-Tuple fuzzy linguistic representation model for computing with words, 2000), linguistic OWA operators (Herrera, Herrera-Videman, & Verdegay, Direct approach processes in group decision making using linguistic OWA operators, 1996), etc. In most of those aggregation operators, a set of equally-informative (balanced) terms is assumed. That is,

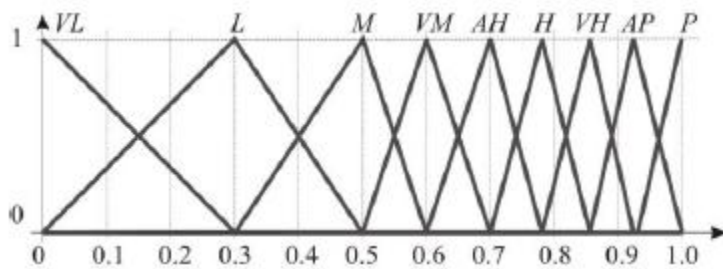
the membership functions of the fuzzy terms are symmetric and the terms are uniformly distributed around a mid (i.e., neutral) term.

For example:



In some cases, fuzzy sets should be asymmetric or more positive than negative terms are needed (or vice-versa), leading to an unbalanced set of labels.

For example:

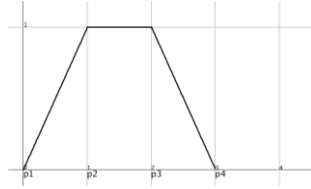


The LOWA operator defined in (Herrera, Herrera-Videman, & Verdegay, Direct approach processes in group decision making using linguistic OWA operators, 1996) is the basis of the new operator, called ULOWA (Unbalanced Linguistic Ordered Weighted Average). LOWA relies on a symbolic model of the set of linguistic terms, and the aggregation is done considering only the positions of the labels in the set of terms, but without taking into account any fuzzy set associated to the terms.

The aggregation procedure of ULOWA is the same than the one used in LOWA. However, in the convex combination of two labels, ULOWA exploits the knowledge given by the membership functions during the aggregation procedure. This approach, although it is focused on managing unbalanced label sets, can also be used with balanced label sets giving the same results than the LOWA operator.

Before explaining the method, we define the parameters needed to understand how the ULWA works.

- **Linguistic values:** The semantics of the linguistic labels are given by a trapezoidal membership function $\mu: X \rightarrow [0,1]$ that is represented with a tuple $P = (p_1, p_2, p_3, p_4)$, where $p_1 \leq p_2 \leq p_3 \leq p_4$ are the points in the reference domain X which define the trapezoid. Some special cases can be defined.
 - If $p_1 = p_2$ and $p_3 = p_4$: then A corresponds to a crisp interval.
 - If $p_2 = p_3$ the fuzzy set A is triangular.
 - If $p_1 = p_2 = p_3 = p_4$ then A is called a crisp real number.



- **Centre of gravity:** Let us denote the centre of gravity (COG) of a trapezoidal fuzzy set A as $GCO(A) = (x_A^*, y_A^*)$, which can be calculated as follows:

$$y_A^* = \begin{cases} \frac{1}{6} \left(\frac{p_3 - p_2}{p_4 - p_1} + 2 \right) & , if \ p_1 \neq p_4 \\ \frac{1}{2} & , if \ p_1 = p_4 \end{cases}$$

$$x_A^* = \frac{y_A^* (p_3 + p_2) + (p_4 - p_1)(1 - y_A^*)}{2}$$

- **Similarity between fuzzy sets:** To find the similarity between a crisp number and a triangular or trapezoidal fuzzy set, the method uses the similarity function between two fuzzy sets P and Q defined by Chen (Chen, A new similarity measure for generalized fuzzy numbers based on geometric-mean averaging operator, 2006). This function is based on the different properties of the membership function of the fuzzy set, such as the centre of gravity, the radius of gyration, or the geometric mean. For any similarity function, the following three properties should be satisfied (Chen & Chen, Fuzzy risk analysis based on

similarity measures for generalized fuzzy numbers, 2003). Given two fuzzy sets P and Q :

- P and Q are identical if and only if $Sim(P, Q) = 1$.
- $Sim(Q, P) = Sim(P, Q)$.
- If $P = (p, p, p, p)$ and $Q = (q, q, q, q)$ are two crisp numbers in $[0..1]$, then de $Sim(P, Q) = 1 - |p - q|$

$$Sim(A, B) = \left[\sqrt[4]{\prod_{i=1}^4 (2 - |p_i - q_i|)} - 1 \right]$$

3.2.2 – The unbalanced linguistic ordered weighted average

The new operator, called ULOWA (Unbalanced Linguistic Ordered Weighted Average), is defined on the basis of the LOWA operator.

If $A = \{a_1, \dots, a_m\}$ is the set of labels to aggregate, the LOWA operator is defined as:

$$\begin{aligned} LOWA(a_1, \dots, a_m) &= W \cdot B^T = C^m\{w_k, b_k, k = 1, \dots, m\} \\ &= w_1 \otimes b_1 \oplus (1 - w_1) \otimes C^{m-1}\{\beta_h, b_h, h = 2, \dots, m\} \end{aligned}$$

Where $\beta_h = w_h / \sum_{h=2}^m w_h$, $h = \{2, \dots, m\}$ and $B = \{b_1, \dots, b_m\}$ is a permutation of the elements of A , such that $B = \sigma(A) = \{a_{\sigma(1)}, \dots, a_{\sigma(m)}\}$, where $a_{\sigma(j)} \leq a_{\sigma(i)} \forall i \leq \forall j$.

C^m is the convex combination of m labels. If $\exists j \in 1 \dots m$ $w_j = 1$ and $\forall k \in 1 \dots m$, $k \neq j$ $w_k = 0$, then $C^m\{w_i, b_i, i = 1, \dots, m\} = b_j$.

When $m = 2$, with $b_1 = s_j, b_2 = s_i, s_j, s_i \in S(j \geq i)$ then: $C^2\{w_i, b_i, i = 1, 2\} = w_1 \otimes s_j \oplus (1 - w_1) \otimes s_i = s_k$ such that $k = \min\{T, i + \text{round}(w_i(j - i))\}$.

Notice that, in this definition, the resulting label s_k is obtained considering the position of the labels s_j and s_i in the linguistic domain, but without considering the form of the membership function associated to each term.

Considering only the order between the terms implies the assumption that terms are defined with symmetric membership functions and uniformly distributed. For this reason, as it has been said, the LOWA operator is not able to deal with unbalanced sets of linguistic terms.

The ULOWA operator takes the same form than LOWA, which is:

If $A = \{a_1, \dots, a_m\}$ is the set of labels to aggregate, the ULOWA operator is defined as:

$$ULOWA(a_1, \dots, a_m) = W \cdot B^T = C^m\{w_k, b_k, k = 1, \dots, m\} = \\ w_1 \otimes b_1 \oplus (1 - w_1) \otimes C^{m-1}\{\beta_h, b_h, h = 2, \dots, m\}$$

The difference with LOWA lies in the convex combination of two linguistic terms, which is calculated taking into account the membership functions of the terms. So, when $m = 2$, the convex combination of the two terms $b_1 = s_j$ and $b_2 = s_i$, with $s_j, s_i \in S (j \geq i)$ is calculated taking into account the membership functions of the labels s_j and s_i :

$$C^2\{w_i, b_i, i = 1, 2\} = w_1 \otimes s_j \oplus (1 - w_1) \otimes s_i = s_k, \text{ such that}$$

$$k = \underset{i \leq p \leq j}{argmax} \{Sim(S_p, \delta)\}$$

In this expression, δ is a crisp number defined as $\delta = (x_k, x_k, x_k, x_k)$ with $x_k = x_{s_i}^* + w_1(x_{s_j}^* - x_{s_i}^*)$, where $x_{s_i}^*$ is the x-component of the centre of gravity (COG) of the fuzzy set associated to the label $s_i = (p_1, p_2, p_3, p_4)$.

In this way, the aggregation result s_k is the linguistic term comprised between s_j and s_i with the greatest similarity with the intermediate point δ .

3.2.3 – Properties of the ULOWA operator

This method satisfies the following properties:

Property 1: The ULOWA operator is increasing monotonous with respect to the argument values, in the following sense:

Let $A = [a_1, \dots, a_m]$ and $B = [b_1, \dots, b_m]$ be two ordered argument vector such that $\forall j, a_j \geq b_j$ then: $ULOWA(A) \geq ULOWA(B)$.

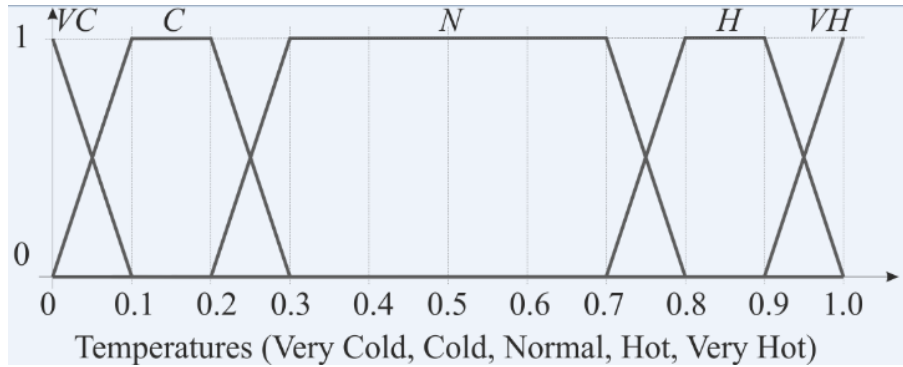
Property 2: The ULOWA operator is commutative, i.e., $ULOWA(a_1, \dots, a_m) = ULOWA(a'_1, \dots, a'_m)$ where $(a'_1, \dots, a'_m)$ is any permutation of the elements in $(a_1, \dots, a_m)$.

Property 3: The ULOWA operator is idempotent in the sense that if $\forall j, a_j = a$, then $ULOWA(a_1, \dots, a_m) = a$.

Property 4: The ULOWA operator is an “orand” operator. That is, for any weighting vector W and ordered labels $A = [a_1, \dots, a_m]$ ($a_1 \geq a_2 \geq \dots \geq a_m$), then: $\min(a_1, \dots, a_m) \leq ULOWA(a_1, \dots, a_m) \leq \max(a_1, \dots, a_m)$.

3.2.4 – Example

Let us take three options to aggregate using the ULOWA operator. The granularity of the fuzzy set is 5 and the definition of the membership functions is shown in the following image. The weighting vector is $W = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, which corresponds to the “mean” average. The linguistic labels to aggregate are VC, N, H .



1. The ordered set of linguistic labels to aggregate is H, N, VC .
2. The ULOWA aggregation begins following:

$$\begin{aligned}
 ULOWA(H, N, VC) &= W \cdot B^T = C^3\{w_k, b_k, k = 1, \dots, 3\} \\
 &= \frac{1}{3} \otimes H \oplus \left(1 - \frac{1}{3}\right) \otimes C^2\{w_k, b_k, k = 2, \dots, 3\}
 \end{aligned}$$

$$C^2\{w_k, b_k, k = 2, \dots, 3\} = \left(\frac{1}{3}\frac{3}{2} \otimes N \oplus \left(1 - \frac{2}{1}\right) \otimes VC\right) = s_k$$

Now, the two terms to aggregate are $s_j = N$ and $s_i = VL$.

3. The corresponding centres of gravity of N and VC are:

$$COG(N) = \left\{ \begin{array}{l} \frac{1}{6} \left(\frac{0.7 - 0.3}{0.8 - 0.2} + 2 \right) = 0.44 \\ \frac{0.44(0.7 + 0.3) + (0.8 + 0.2)(1 - 0.44)}{2} = 0.5 \end{array} \right\} = (0.5, 0.44)$$

$$COG(VC) = \left\{ \begin{array}{l} \frac{1}{6} \left(\frac{0 - 0}{0.1 - 0} + 2 \right) = 0.33 \\ \frac{0.33(0 + 0) + (0.1 + 0)(1 - 0.33)}{2} = 0.0335 \end{array} \right\} = (0.0335, 0.33)$$

4. The intermediate point δ is calculated as follows:

$$\delta = x_N^* + w_1(x_{VC}^* - x_N^*) = 0.5 + 0.5(0.0335 - 0.5) = 0.26$$

5. The similarities of VC , C and N with $\delta = 0.26$ are:

$$SIM(VC, \delta) = 0.76029325$$

$$SIM(C, \delta) = 0.86571264$$

$$SIM(N, \delta) = 0.71783894$$

According to the similarity function that we have defined in the previous section, the label with maximum similarity is $s_k = C$, because it is quite near the point $\delta = 0.26$.

6. Then, we continue with the other aggregations repeating the above steps.

4 – Design decisions

4.1 – Introduction

In the previous section the aggregation operators were theoretically presented, explaining how they work. This section is focused on its implementation in the Diviz software.

In important part of the job done in this PFC has been the understanding of the Diviz software requirements for adding new compatible plugins. In order to develop the software and input files we have had to adapt to the specifications of the Decision-Deck project. This has forced us to work in a very specific way as we were not allowed to change the current specifications of Diviz. The Diviz software development is coordinated by Dr. Sebastián Bigaret from the Decision Deck board.

As the communication between the Diviz modules is done with the language XMCDa, a deep comprehension of all the elements of this language has been required. XMCDa is still not an official standard, it is under construction. The language definition is in charge of the Decision Deck board, mainly Dr. Patrick Meyer is in charge of this task. When new modules are being implemented, new requirements appear for being included in the XMCDa language. These requirements are reported and studied, and after an agreement on the details of the language is reached, they are incorporated in the following version of XMCDa.

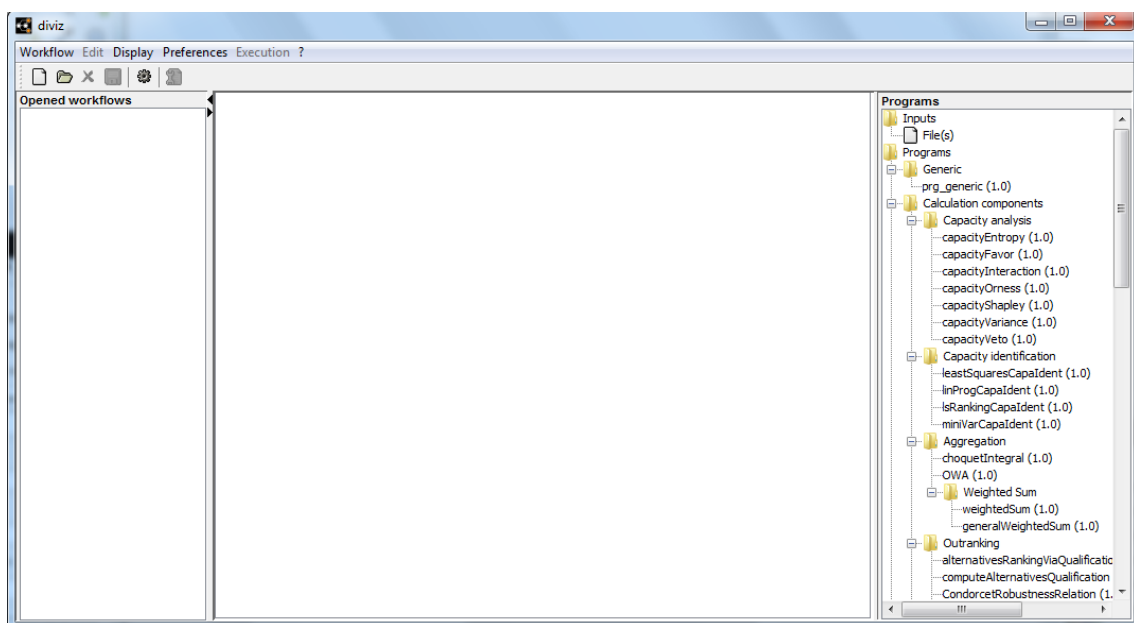
In this section we will explain the design and implementation details of the new modules, as well as all the contributions that we have made on the XMCDa language originated from the new needs that we have found. We must say that during the design and implementation of the OWA and ULOWA plugins, we have had several contacts with Dr. Patrick Meyer and Dr. Sebastián Bigaret, both by email and in a meeting held in Corte (Corsica), during the DD3 Workshop.

4.2 – Short description of basic usage of Diviz

Diviz is structured in independent modules that can be interconnected in different layers. The ones in the first layer must receive input data from XMCDAs files. For example, we can start by downloading the following data files which respect the XMCDAs standard:

- alternatives.xml (contains a list of alternatives IDs)
- criteria.xml (contains a list of criteria IDs)
- performanceTable.xml (contains a performance table)

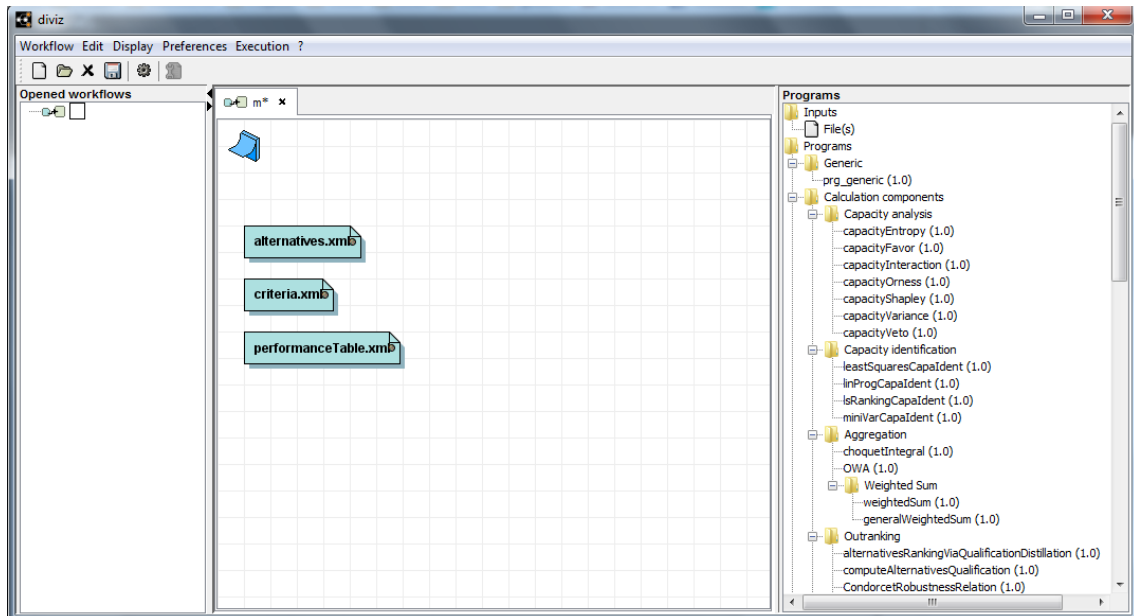
After the splash screen you can see an empty workspace. The left panel is used to display the open workflows and their past executions. The right pane contains the list of available elementary components. The centre pane is used to construct and display the workflows.



To create a new workflow, open the workflow menu and click on new. Choose a name for your workflow.

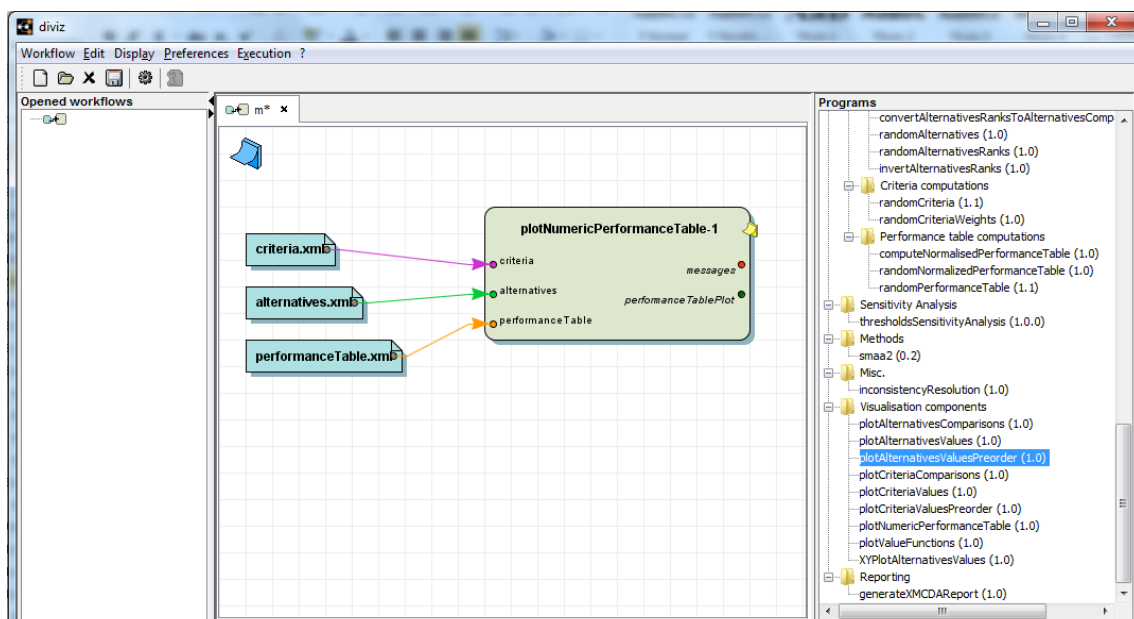
From the right panel, drag and drop a File(s) element into the central panel. A dialog window will open and ask you to choose a file. Choose the file alternatives.xml that you downloaded earlier.

Repeat this process for the remaining two files `criteria.xml` and `performanceTable.xml`. You should now have a screen which looks similar to the following picture.



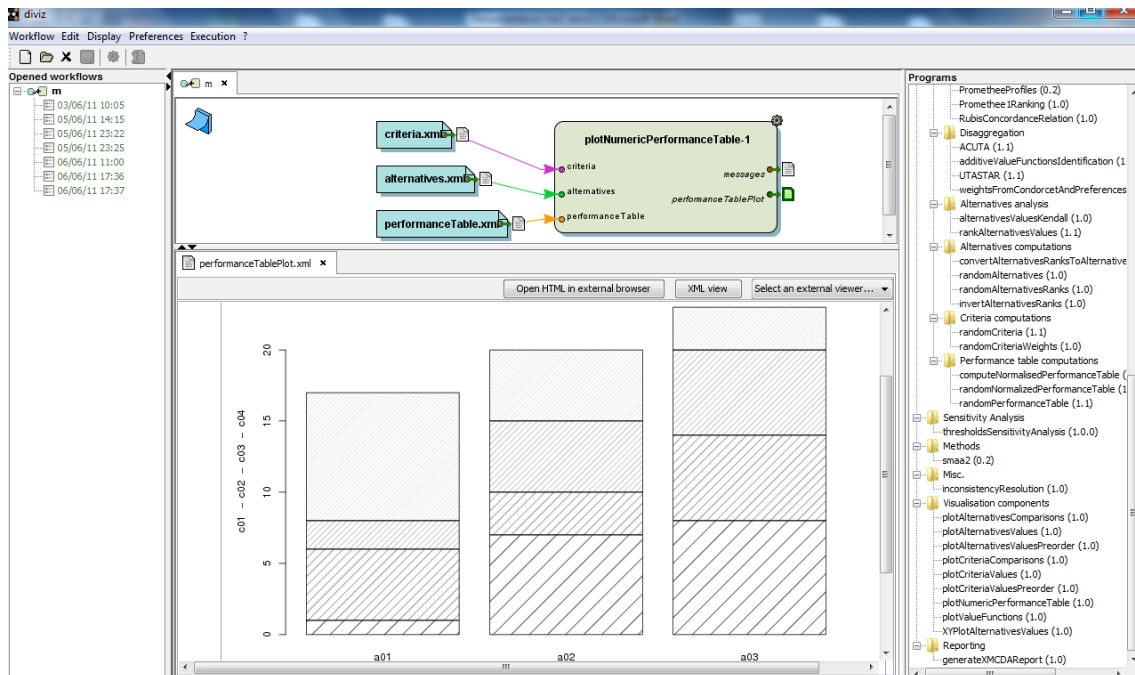
You can now add an elementary component from the right panel. For example **plotNumericPerformanceTable** (under the category Visualisation components). Simply select it and drag and drop it to the central panel.

You can now connect the data files to the right entries on the newly created component:



You have now created your first workflow in Diviz. It is time now to execute the workflow and have a look at the output of the elementary component. You can do so via the **Execution** menu via the **Run** entry.

After the workflow has been successfully executed, you can click on the little file icons in the workflow to view the results. Your screen should look like the following image.



4.3 – How to create a module for Diviz

The Decision Deck project offers architecture dedicated to web services. Its primary goal is to offer a place where MCDA algorithms, methods etc. can be hosted and made freely available to anyone.

One basic way to access these web services is to use standard SOAP requests. More user-friendly ways include the Diviz software or Decision Deck's d3 internet application.

Now imagine that you have a program and you would like to make it available both through the Decision Deck web services' architecture and a GUI software. Let us see what kind of information you need, and what the requirements are.

In order to deploy your program within the Decision Deck web services' architecture, you need to know:

- How to install your program.
- How to call your program.
- The details of its inputs and outputs.

This information is necessary for introduce you program in Diviz or d3 and it's important to tell it to Decision Deck.

To access a web service from a GUI requires the input data which is sent to the web service to be either loaded from a file or entered by hand by a user.

In order to prepare for making a program available as a web service, the following requirements must be met:

1. The executable (either a binary or a script) must be available on Linux.
2. The program must understand XMCDAs files, meaning that it accepts them as inputs, and that it is able to save its output(s) conforming to this standard as well.
3. The program must be callable through the command-line (at least). This means that even if it also offers a graphical interface, it can also be launched directly on the command-line (or, if you prefer, in so-called batch scripts) and that, in this mode, it does not expect any interaction at all with the user.

As already mentioned, your program should read its input elements from files compliant with XMCDAs. Similarly, the output elements should be written according to this standard. Besides, we have opted for the following architecture:

- Multiple input and output files (one per main XMCDAs tag).
- 2 distinct directories for the input and the output files.

We require that (at least) two options can be sent to your program via the command line:

1. The name of the input directory.

2. The name of the output directory.

The **input directory** will contain all the input files necessary for your program to work. As already mentioned, each input file is compliant with the XMCD A standard, and should contain at most one of the main XMCD A tags. Each file can freely be named according to your definition of its content (mandatory extension: .xml).

The **output directory** will contain all the output files which are produced by your program. They are also compliant with XMCD A.

The names of the input and output files can be hard-coded in your program. Consequently, your program should look in the input directory (passed as a parameter to the program) for the files, load them, execute the algorithm, and write its outputs to the output directory (passed as a parameter to the program).

To summarise, your program should somehow be callable as follows from command line (the exact syntax is up to you):

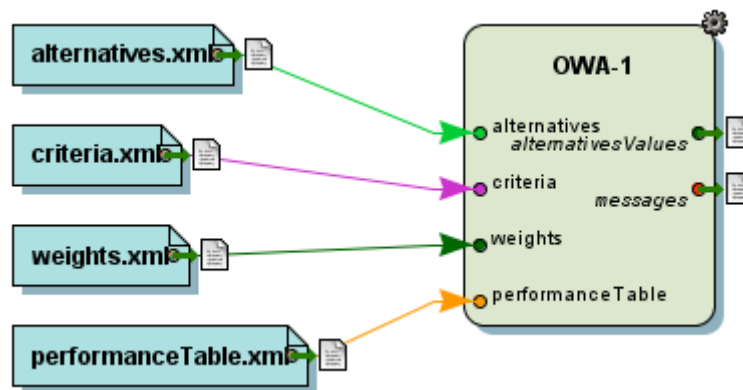
```
> program -i inputDirectory -o outputDirectory
```

where “program” is the command used to run your program, and “inputDirectory” and “outputDirectory” are the names of the input and output directories where the XMCD A files are (or will be) located.

There must be sent benchmark to Diviz responsible to verify in future versions that modules continue working correctly. These benchmarks might be downloaded as examples of the module.

4.4 – Ordered Weighted Average

The first module that we have created is called OWA. The view of a workflow including this module is given here:



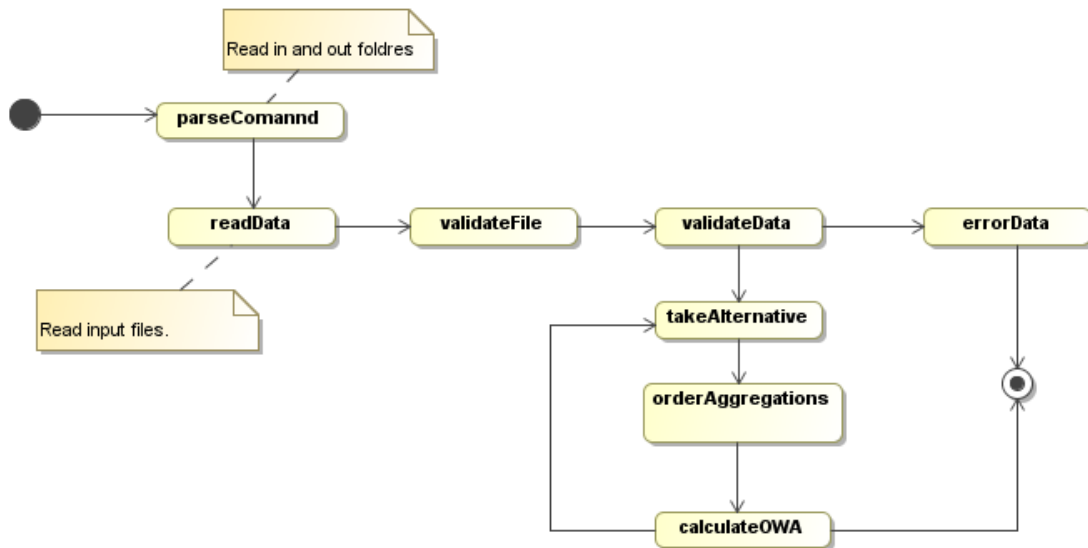
The module has been designed to be compatible with the rest of modules in Diviz, and with the standards on the input files. The input files Alternatives, Criteria and PerformanceTable are the same that use other modules.

The file Weights is different from the ones used in other weighted operators, because the weights in OWA have a different role, as explained in section 4.4.3 – XMCDAs inputs diagram. The output file also is compliant with the rest of modules, in order to facilitate their interconnection, for example, the output can be introduced into a plot module.

This structure has been validated by the Diviz developers in order to be fully compliant with the language.

4.4.1 – State diagram

In this section we can see what steps does the application. Initially, the program reads the command and then reads the input files, if there is any error in the files, or data contained therein, the application will terminate and will not run the OWA method.



There are two types of validation, validation of the file structure or data validation. This program covers the following errors:

- Namespace failure, the file should begin as follows:

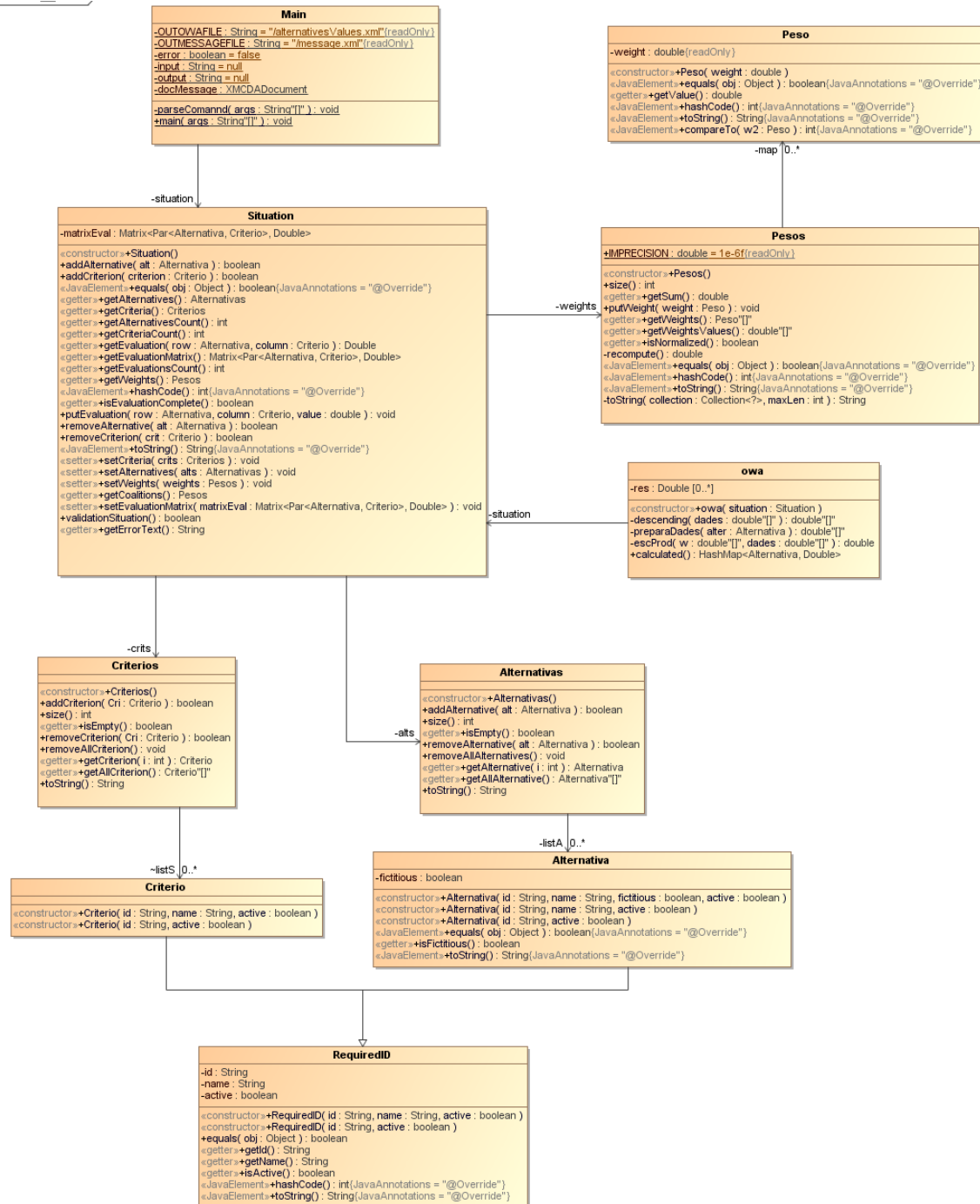
```
<xmc:XMCD xmlns:xmc="http://www.decision-deck.org/2009/XMCD-2.1.0">
```

- Error in definition files:

```
<alternatives>
  <alternative id="a01" />
  <alterna id="a02" />
</alternatives>
```

- In the data errors, it is important that the identifiers of the alternatives and criteria are the same as found in the performance table.
- The sum of the weights is 1.
- There is the same number of weights as criteria.

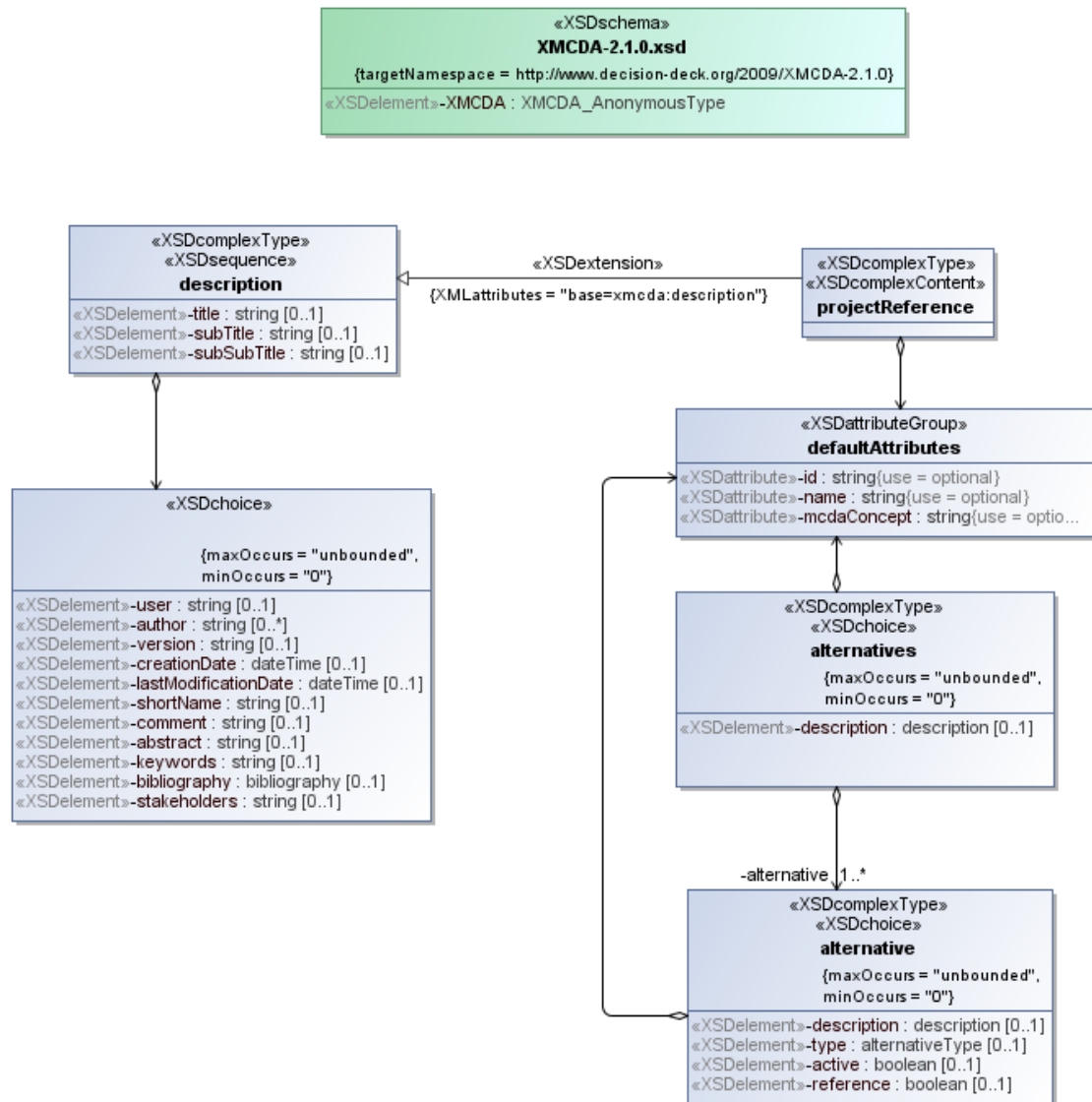
4.4.2 – Class diagram



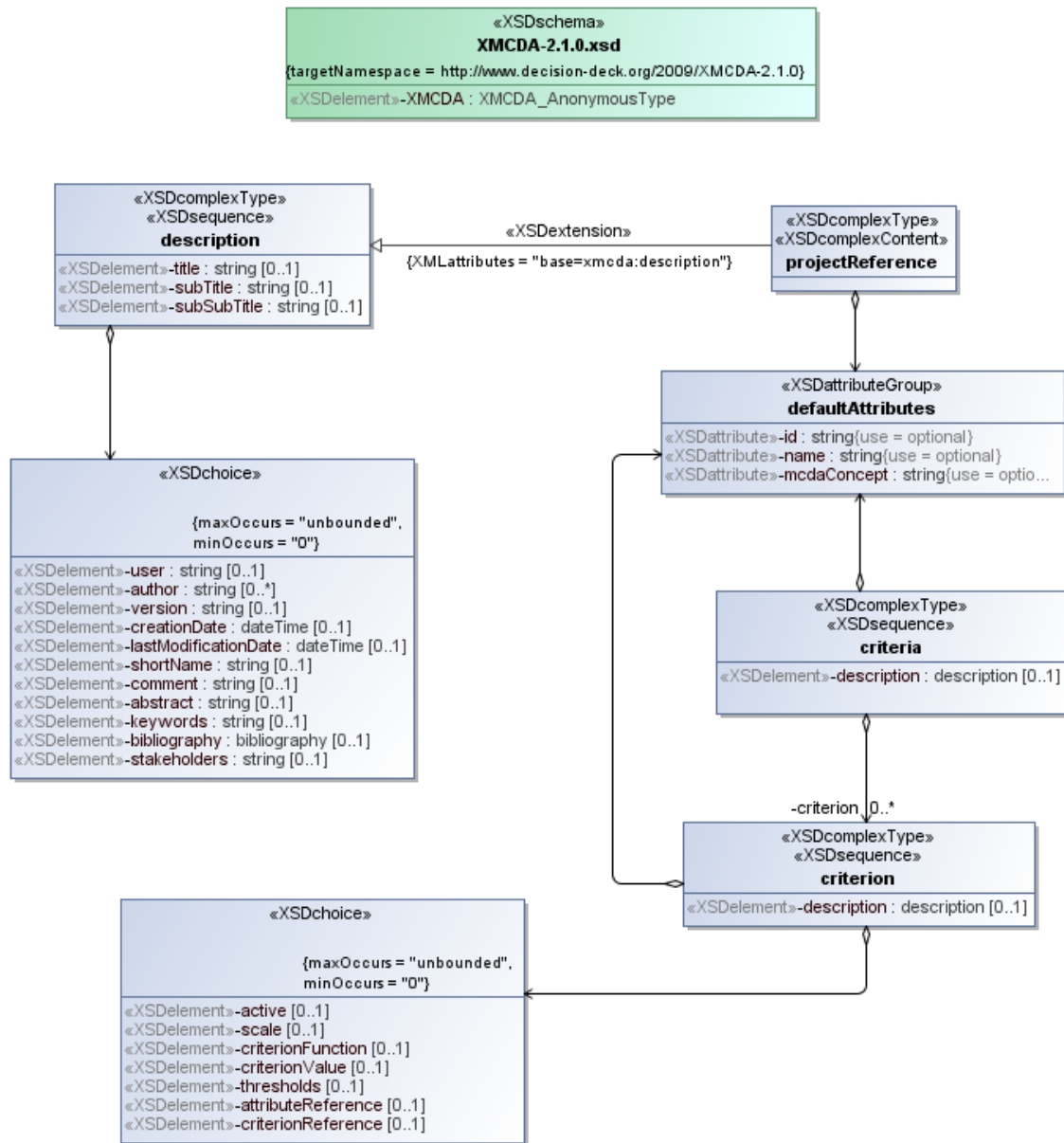
4.4.3 – XMCDa inputs diagram

In this section we display graphically the structure of the different input files. For Alternatives, Criteria and PerformanceTable we follow the standard structure. As said before, for Weights we have had to define the new structure given here.

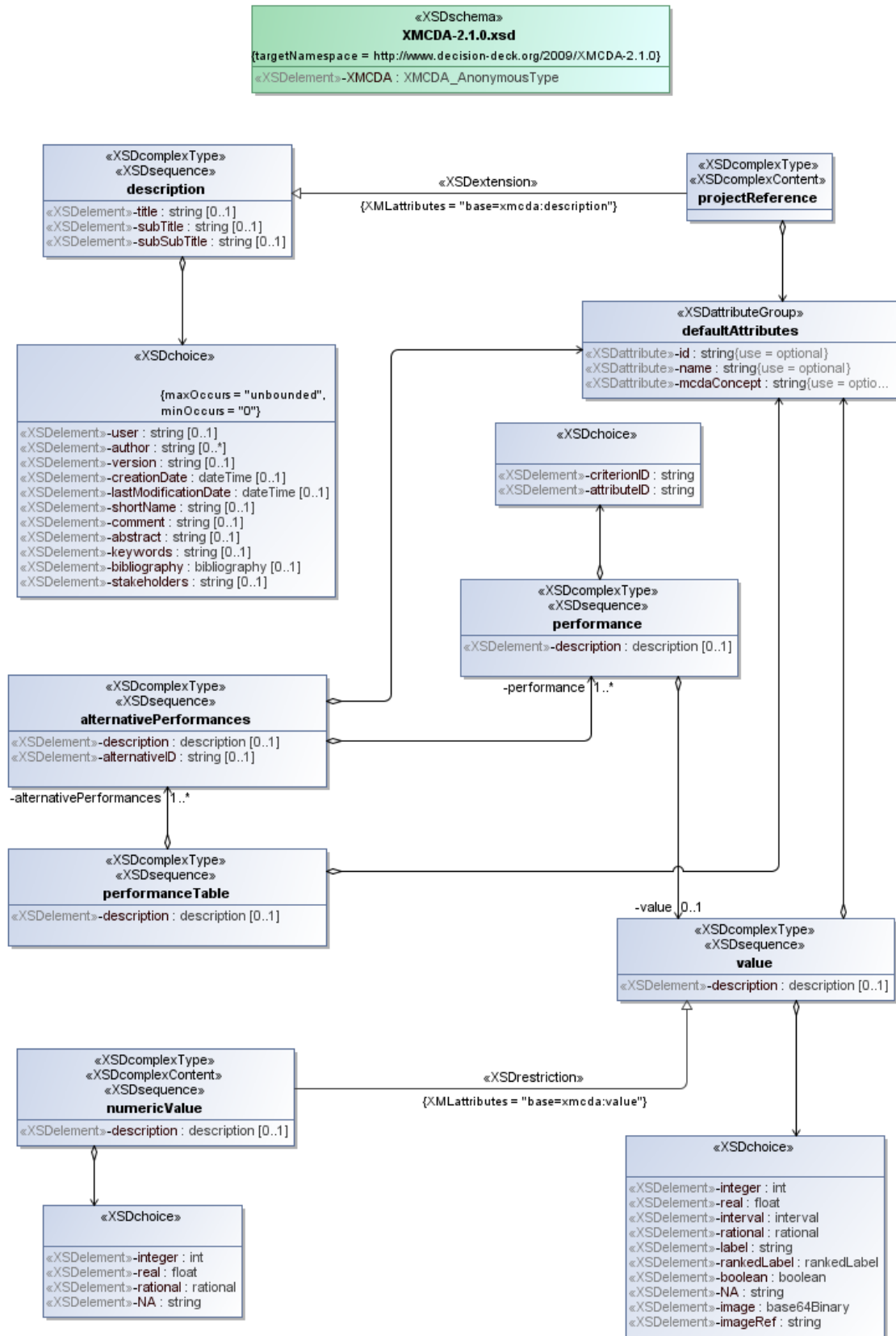
4.4.3.1 – Alternative



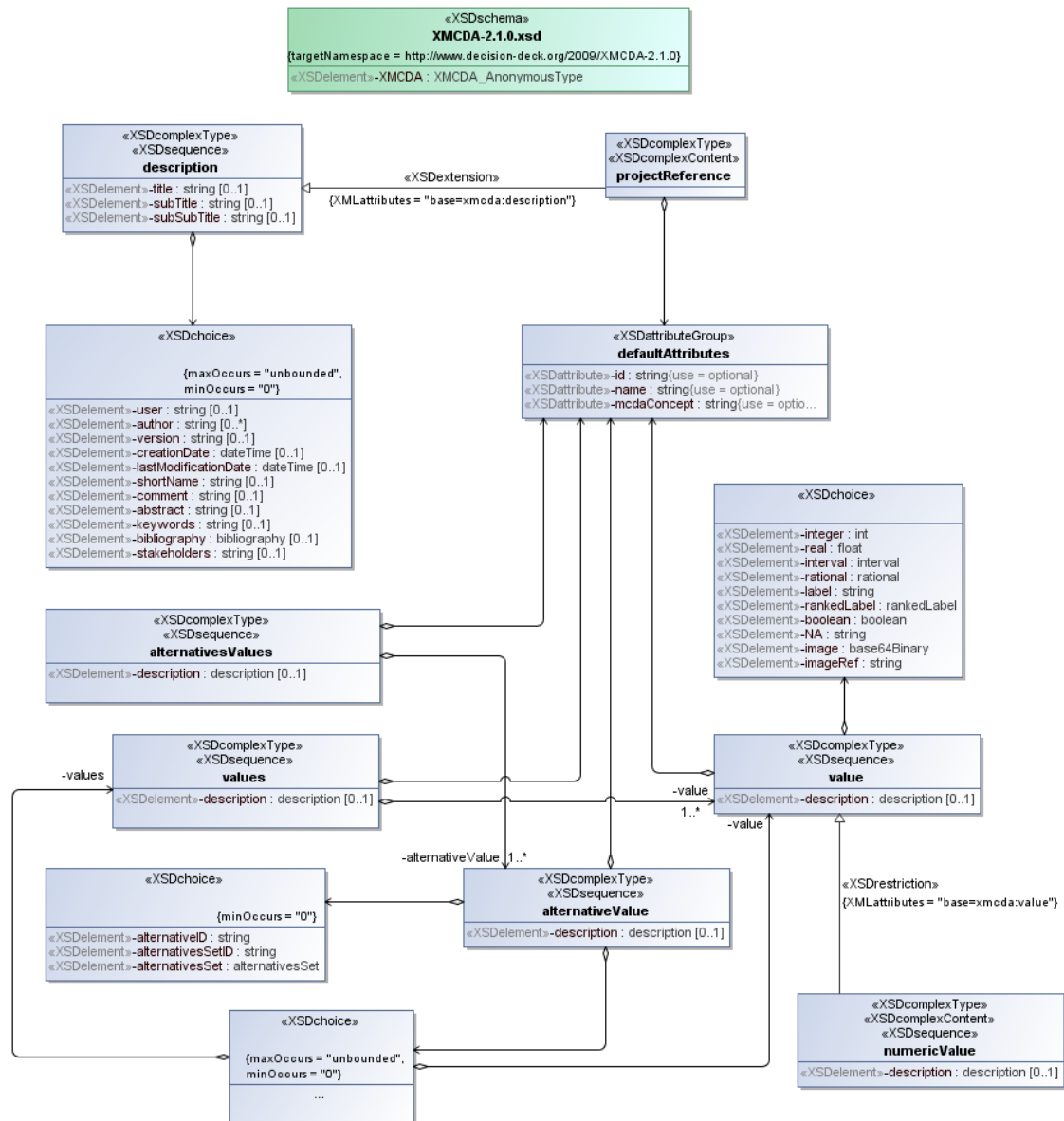
4.4.3.2 – Criteria



4.4.3.3 – Performance table



4.4.3.4 – Weights

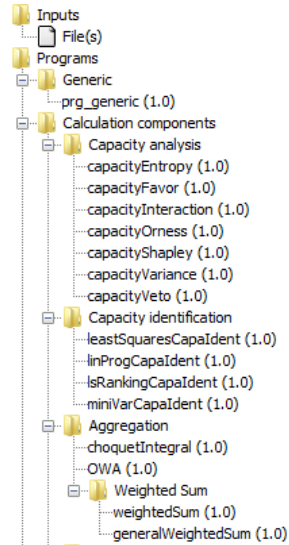


4.4.4.1 – OWA



4.4.5 – OWA module

In the latest version of Diviz (1.6) we can find the module OWA. This method can be found in the menu of aggregation together with Choquet integral.



To verify that everything works correctly we have prepared a set of test kits to evaluate the system response. These test kits are required to incorporate any new method in Diviz. The tests sent are:

- Test 1: Example without errors.
- Test 2: With incorrect *Namespace*, in file "alternatives.xml".
- Test 3: With incorrect XMCD syntax, in file "alternative.xml".
- Test 4: The id of one alternative is different for id of performanceTable. In file "alternatives.xml".
- Test 5: The weights are not normalized. In file "weights.xml".
- Test 6: The weights are not normalized therefore missing a weight. In file "weights.xml".
- Test 7: Missing information in performanceTable. In file "performanceTable.xml" on alternative "a01" missing the criterion "c06".
- Test 8: Missing criterion. In file "criteria.xml".

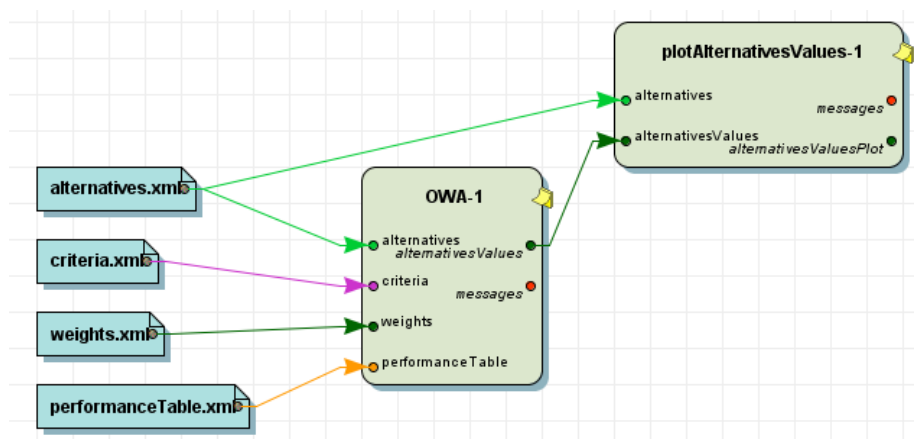
4.4.6 – Example

This example shows the results obtained by running the OWA on Diviz, with the following parameters:

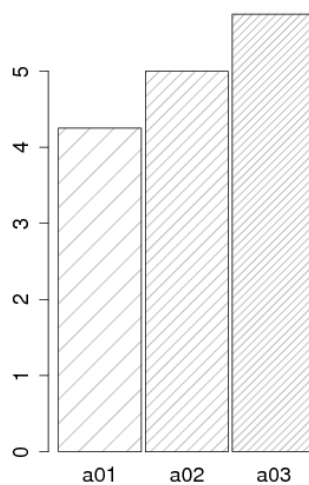
- We have 3 alternatives (a01, a02, a03).
- We have 4 criteria (c01, c02, c03, c04).
- The weights used are the following (0.25, 0.25, 0.25, 0.25).
- The performance table is:

	c01	c02	c03	c04
a01	1	5	2	9
a02	7	3	5	5
a03	8	6	6	3

Prepared the following workflow

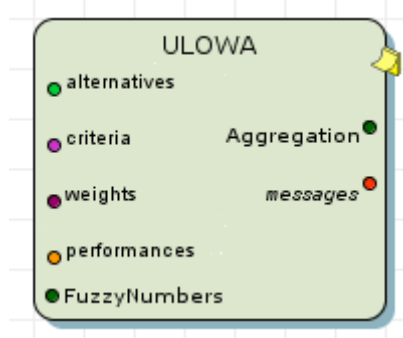


The result is:



4.5 – Unbalanced Linguistic Ordered Weighted Average

The second module that we have created is called ULOWA. It is an linguistic module. The view of a workflow including this module is given here:



The module tries to be compatible with the rest of modules in Diviz, but it introduces some novel concepts on Diviz: fuzzy sets and linguistic variables.

The input files Alternatives and Criteria are the same that use other modules, because they only define the names and identifiers. Performance Table differs from the kind of input files using non-numeric data. Instead it uses strings data.

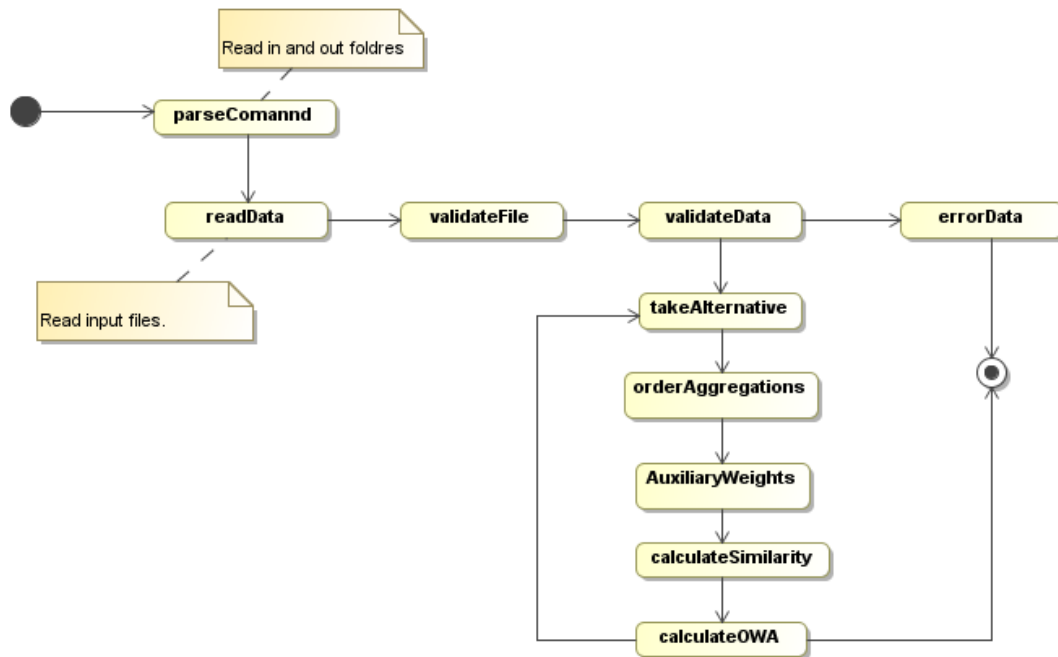
The file Weights is different from the ones used in other weighted operators, because the weights in ULOWA have a different role, as explained in section 4.4.3.4. However it follows the same files format as used in OWA module.

FuzzyNumbers is a new file that defines the vocabulary that is used in the Performance Table. It corresponds to the terms of a linguistic variable. Each term has a fuzzy set associated. To be able to introduce this sort of data we had to speak with the responsible of the XMCD language. We had to add fuzzyNumbers to the language. We avoided modifying as much as possible the language to maintain the uniformity with other versions. This modification let us add triangular or trapezoidal forms with different sizes, either heights or widths. As explained in section 4.5.3.1.

The output file now contains strings as result for each alternative. The structure is the same than the OWA module, but including strings instead of numbers.

4.5.1 – State diagram

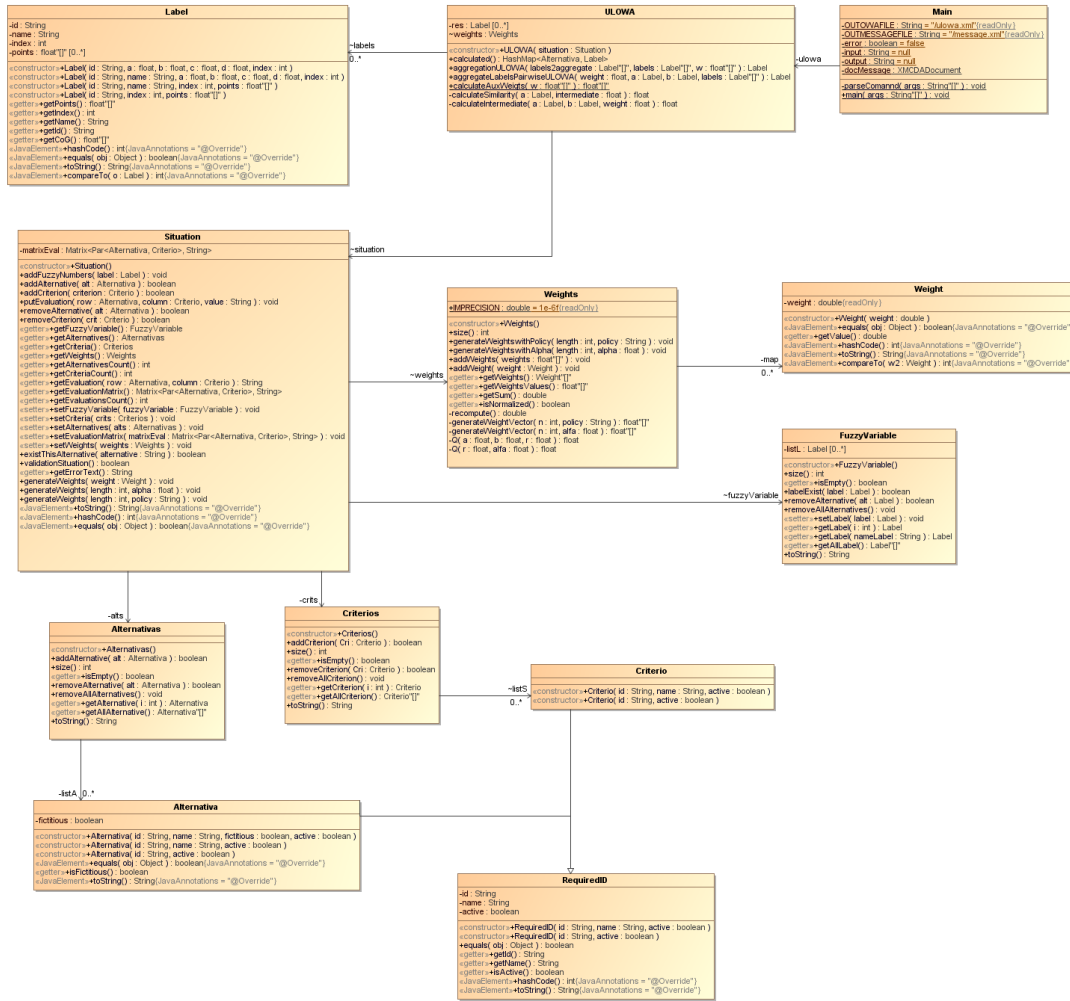
As we talked in paragraph 4.4.1 – State diagram, the state diagram of this second module differs from the first only in how to calculate the method. In this case the linguistic model needs to do more calculations.



This program covers the following errors:

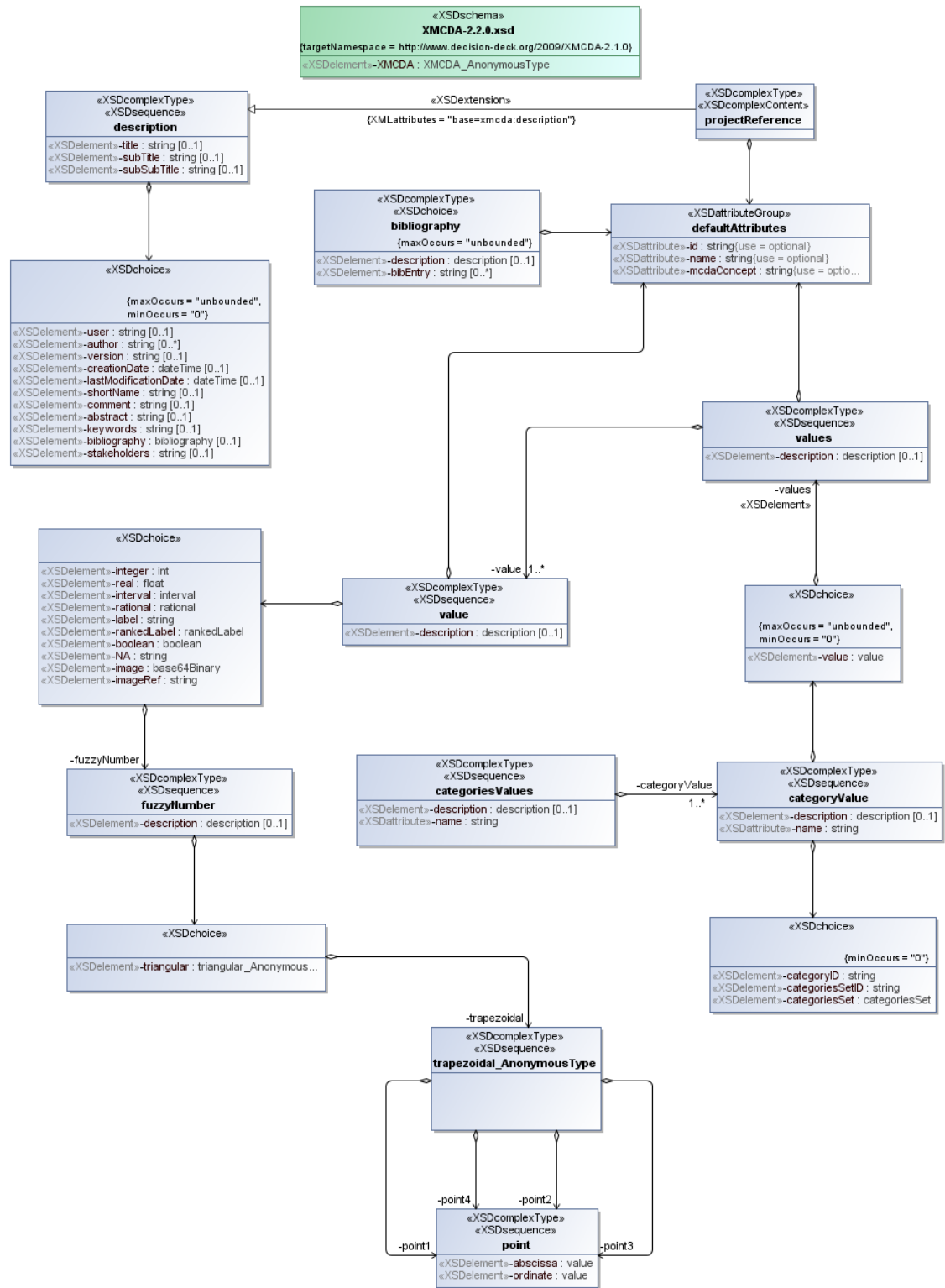
- Namespace incorrect, the program works with 2.2.
- Detects errors in the file structure.
- If everything went right till this point, evaluates the data entered:
 - That the sum of the weights is 1.
 - That the performance table have valid tags:
 - No null.
 - Have a value equal to fuzzyNumber.
 - Validate the values of the performance table with case insensitive.

4.5.2 – Class diagram



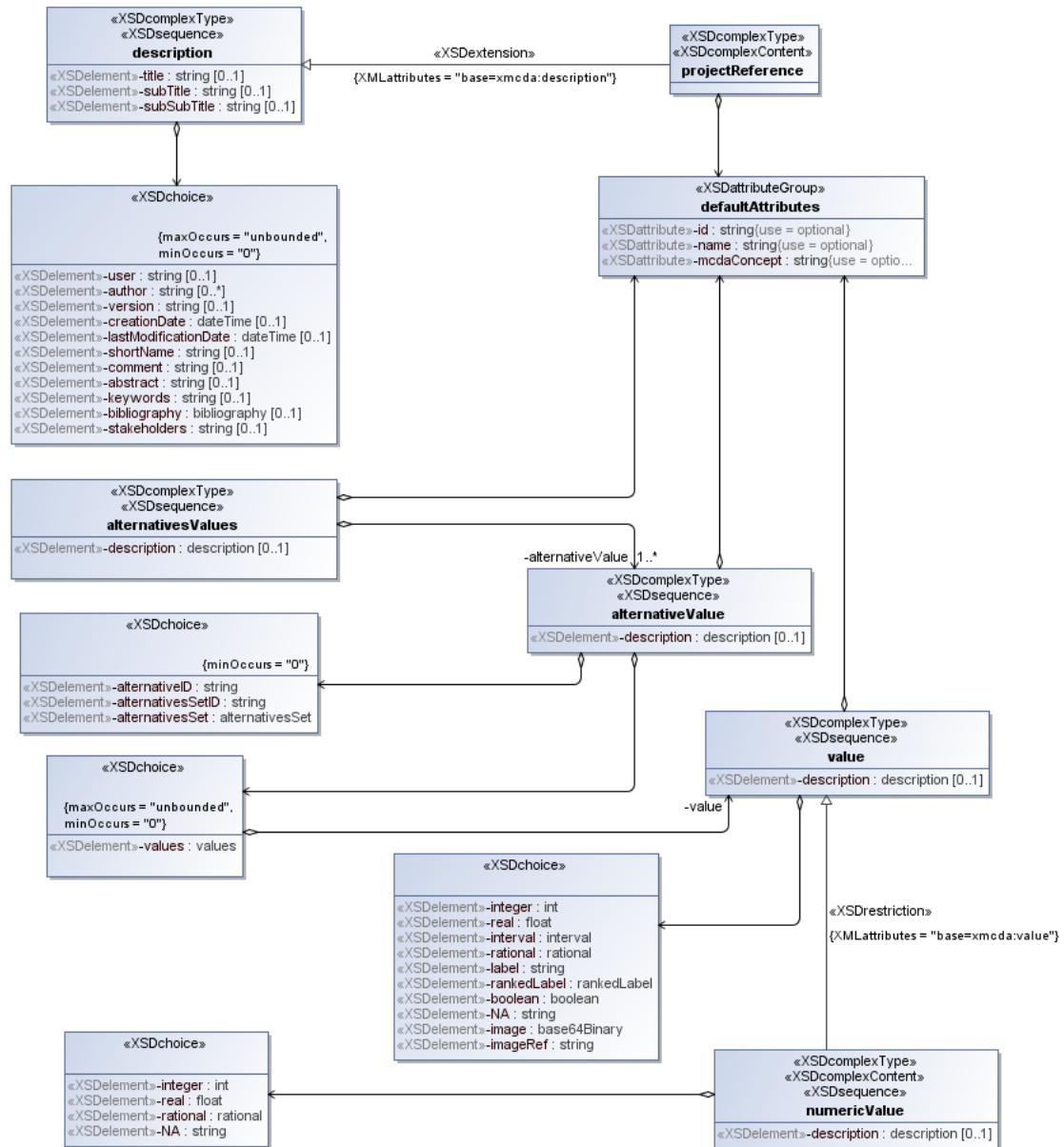
4.5.3 – XMCDa inputs diagram

4.5.3.1 – fuzzyNumber



4.5.4 – XMCDa output diagram

4.5.4.1 – ULOWA



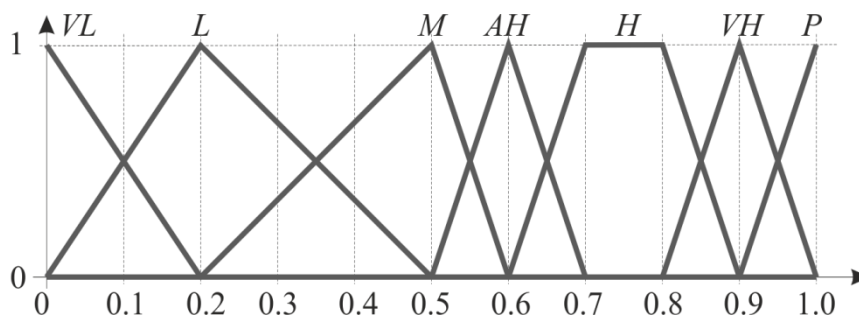
4.5.5 – ULOWA module

The ULOWA is not part of the next version of Diviz (1.6), because it requires changing the standard XMCDA. In the next version of this standard (2.2), it is planned to introduce the *fuzzyNumbers* necessary for the model ULOWA.

This module has been built on the basis of the OWA module, so the verification is the same. We have included some new tests kits. They are:

- Test 1: With 2 criterion and 5 alternatives. Weight = (0.5, 0.5).
- Test 2: With 5 criterions and 2 alternatives. Weight = (0.2, 0.2, 0.2, 0.2, 0.2).
- Test 3: With 5 criterions and 2 alternatives. Weight = (0.25, 0.25, 0.25, 0.25, 0).

All tests have used the following fuzzyNumber:



4.6 – How to document the program

All technical details concerning the program which you are submitting for inclusion into Decision Deck's web services infrastructure works is stored in a file called **specification.xml**. It also contains information on how it should be made accessible from within Diviz. This section explains how to write such a descriptive file.

The **specification.xml** file allows describing your program which you wish to integrate into Decision Deck's web-services framework (and into the Diviz software). We describe here its entries.

Once your program has been made available as a web service, it can be integrated into Diviz. However, in Diviz, there exists the possibility to modify manually the values of certain input elements via a dialog box related to your program.

It is therefore important to distinguish between 3 types of input and output data elements:

- Editable input parameters (which can be modified in a GUI).
- Input data files (using the XMCD standard), represented as an input connection point in Diviz.
- Output data files (using the XMCD standard), represented as an output connection point in Diviz.

Note that the grammar of **specification.xml** allows specifying that an editable input parameter should optionally become a standard input connection point (and vice-versa).

Finally remember that it is up to you to decide if you wish to allow Diviz users to edit certain parameters via a dialog box, or let them submit these data in a standard way as input data files to your program. This choice has no influence on the web service, its call via a SOAP request or your program.

Let us now switch to the detailed documentation of the **specification.xml** file. It is written in XML and structured as follows.

The root tag of **specification.xml** is given below:

```
<program_description>
  [...]
</program_description>
```

- The *program_description* has a certain number of children which are described hereafter.

Program name and technical information:

```
<program provider="URV"
  name="OWA"
  displayName="OWA"
  version="1.0" />
```

- The *provider* attribute specifies who is providing the web service;
- The *name* attribute specifies the name of the web service on the distant server;

- The *displayName* attribute specifies the name which should be displayed for the user (in software using the web service).
- The *version* attribute specifies which version of the software is available as a web service.

Description of the program:

```
<documentation>
  <description>[...]</description>
  <contact>[...]</contact>
  <url>[...]</url>
  <reference>[...]</reference>
</documentation>
```

- The *<description>* tag describes the purpose of the web service;
- The *<contact>* tag specifies a contact person which can help in case of problems with the web service (optional);
- The *<url>* tag specifies an URL where further information can be found on the web service (optional);
- The *<reference>* tag gives a bibliographical reference for the description of the web service (optional).

They specify the contents of the input data, output data and provide information on the editable input parameters in Diviz.

```
<parameters>
  <input [...]>
    [...]
  </input>
  [...]
  <output [...]>
    [...]
  </output>
  [...]
</parameters>
```

Detailed description of *<input>* and its children:

```
<input id="..."
  name="..."
  displayName="..."
  isoptional="0">
```

- The *id* attribute defines a unique identifier for this input element.
- The *name* attribute indicates the name of the parameter in the soap request.

- The *displayName* attribute indicates how this parameter should be displayed in Diviz.
- The *isoptional* attribute indicates if this parameter is optional (1) or if it is mandatory (0).

Let us now describe the children of `<input>`.

- The `<xmcda>` tag allows specifying what XMCDa code is associated with this parameter.

```
<xmcda tag="methodParameters">
  <![CDATA[
    <methodParameters>
      [...]
    </methodParameters>
  ]]>
</xmcda>
```

- The `<documentation>` tag has the same children as the documentation tag used to describe the program.

```
<documentation>
  <description>
    [...]
  </description>
</documentation>
```

Detailed description of `<output>` and its children:

```
<output id="[" [...] "
  name="[" [...] "
  displayName="[" [...] ">
```

- The *id* attribute defines a unique identifier for this output element.
- The *name* attribute indicates the name of the parameter in the soap request.
- The *displayName* attribute indicates how this parameter should be displayed in Diviz.

The children's of the `<output>` tag are the same of the `<input>` tag.

The **specification.xml** file has 3 main purposes:

- Information for the deployment of your program as a web service.
- Documentation of the web service and its use.

- Information elements for the generation of a GUI for the input of certain parameters. (Not explained here).

4.6.1 – OWA documentation file.

```
<?xml version="1.0" encoding="UTF-8"?>
<program_description
  xsi:noNamespaceSchemaLocation="http://sma.uni.lu/d2cms/ws/_downloads/description.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <program name = "OWA" provider = "URV" version = "1.0"
    displayName = "OWA"/>

  <documentation>
    <description>
      The Ordered Weighted Averaging operators, commonly called
      OWA operators, provide a parameterized class of mean type
      aggregation operators.
      OWA was defined by R.R. Yager in 1988. Here we provide an
      implementation of the OWA operator.
      The main characteristic of this aggregation operator is the
      fact that the weights are associated to the values instead
      to criteria.
      In this way, we can define different aggregation policies
      according to the importance that we associate to high and
      low performance scores.
    </description>
    <reference>
      R.R. Yager, On Ordered Weighted Averaging Aggregation, IEEE
      Transactions on Systems, Man and Cybernetics, 18, pp. 119-
      145 (1988)
    </reference>
    <contact>
      Aida Valls <aida.valls@urv.cat>;, Jordi Canals
      <jordi.canals@estudiants.urv.cat></contact>
    </documentation>
    <parameters>
      <input displayName="alternatives" name="alternatives"
        id="input0" isoptional="0">
        <documentation>
          <description>
            A list of alternatives. Alternatives can be activated
            or desactivated via the <active> tag (true or
            false). By default (no <active> tag),
            alternatives are considered as active.
          </description>
        </documentation>
        <xmcd a tag="alternatives">
          <![CDATA[
            <alternatives>
              <alternative>
                <active>[...]</active>
                [...]
              </alternative>
              [...]
            </alternatives>
          ]]>
        </xmcd a>
```

```

</input>
<input displayName="criteria" name="criteria"
  id="input1" isoptional="0">
  <documentation>
    <description>
      A list of criteria. Criteria are always
      active.
    </description>
  </documentation>
  <xmcda tag="criteria">
    <![CDATA[
      <criteria>
        <criterion>
          <active>[...]</active>
          [...]
        </criterion>
        [...]
      </criteria>
    ]]>
  </xmcda>
</input>
<input displayName="weights" name="weights"
  id="input2" isoptional="0">
  <documentation>
    <description>
      The weights are associated to the values
      of the alteranatives. Assuming that the
      values on the alternative will be sorted
      from the best to the worst (e.g. 7, 5, 5,
      3, 1), the list of weights must be
      ordered according to the importance that
      is given to the values, from the highest
      to the lowest. For example a list of
      weights as (0.5, 0.5, 0, 0, 0) is
      ignoring the 3 lowest values, and making
      an average of the two highest ones. A
      list like (0, 0, 1, 0, 0) is calculating
      the median, while (0, 0, 0, 0, 1) is
      taking the minimum. Notice that the sum
      of weights is required to be 1.
    </description>
  </documentation>
  <xmcda tag="alternativesValues">
    <![CDATA[
      <alternativesValues>
        <alternativeValue>
          <values>
            <value>
              <real>[...]</real>
            </value>
            <value>
              <integer>[...]</integer>
            </value>
            [...]
          </alternativeValue>
        </alternativesValues>
      ]]>
    </xmcda>
  </input>
  <input displayName = "performanceTable" name =
    "performanceTable" id="input3" isoptional="0">

```

```

        <documentation>
            <description>
                The performance table will contain only
                numerical values, all in the same scale
                of measurement.
            </description>
        </documentation>
        <xmcda tag="performanceTable">
            <![CDATA[
                <performanceTable>
                    <alternativePerformances>
                        <alternativeID>[...]</alternativeID>
                        <performance>
                            <criterionID>[...]</criterionID>
                            <value>
                                <real>[...]</real>
                            </value>
                        </performance>
                        [...]
                    </alternativePerformances>
                    [...]
                </performanceTable>
            ]]>
        </xmcda>
    </input>
    <output displayName="alternativesValues"
        name="alternativesValues" id="output0">
        <documentation>
            <description>
                Result obtained from the OWA aggregation
                on each alternative.
            </description>
        </documentation>
        <xmcda tag="alternativesValues"/>
    </output>
    <output displayName="messages" name="messages"
        id="output1">
        <documentation>
            <description>A status message.</description>
        </documentation>
        <xmcda tag="methodMessages"/>
    </output>
</parameters>
</program_description>

```

4.6.2 - ULOWA documentation file

```

<?xml version="1.0" encoding="UTF-8"?>
<program_description
    xsi:noNamespaceSchemaLocation="http://sma.uni.lu/d2cms/ws/_downl
oads/description.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <program name="ULOWA" provider="URV" version="1.0"
        displayName="ULOWA"/>

    <documentation>
        <description>
            This module implements the ULOWA aggregation
            operator: Unbalanced Linguistic Ordered Weighted
            Average. Aggregation operators for linguistic
            variables usually assume uniform and symmetrical
            distribution of the linguistic terms that define the

```

variable. However, there are some problems where an unbalanced set of linguistic terms is more appropriate to describe the objects. ULOWA accepts a set of linguistic labels defined with unbalanced fuzzy sets. The fuzzy sets must define a fuzzy partition on the set of reference values. They can be defined by trapezoidal or triangular membership functions.

</description>

<reference>

D. Isern, L. Marin, A. Valls, A. Moreno, The Unbalanced Linguistic Ordered Weighted Averaging Operator, In: Proc. IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2010, IEEE Computer Society, Barcelona, Catalonia, 2010, pp. 3063-3070.

</reference>

<contact>

Lucas Marin <lt;lucas.marin@urv.cat>, Aida Valls
<lt;aida.valls@urv.cat>, Jordi Canals
<lt;jordi.canals@estudiants.urv.cat>

</contact>

</documentation>

<parameters>

<input displayName="alternatives" name="alternatives"

id="input0" isoptional="0">

<documentation>

<description>

A list of alternatives. Alternatives can be activated or deactivated via the <lt;active> tag (true or false). By default (no <lt;active> tag), alternatives are considered as active.

</description>

</documentation>

<xmcd tag="alternatives">

<![CDATA[

<alternatives>

<alternative>

<active>[...]</active>

[...]

</alternative>

[...]

</alternatives>

]]>

</xmcd>

</input>

<input displayName="criteria" name="criteria" id="input1"

isoptional="0">

<documentation>

<description>

A list of linguistic criteria. Criteria are always active.

</description>

</documentation>

<xmcd tag="criteria">

<![CDATA[

<criteria>

<criterion>

<active>[...]</active>

[...]

</criterion>

```

        [...]
    </criteria>
]]>
</xmcd>
</input>
<input displayName="weights" name="weights" id="input2"
    isoptional="0">
    <documentation>
        <description>
            The weights operate in the same way as
            the classical OWA operator. They are
            associated to the values of the
            alternatives rather than to the criteria.
            In this way they can define different
            aggregation policies. Assuming that the
            values on the alternative will be sorted
            from the best to the worst (e.g. 7, 5, 5,
            3, 1), the list of weights must be
            ordered according to the importance that
            is given to the values, from the highest
            to the lowest. For example a list of
            weights as (0.5, 0.5, 0, 0, 0) is
            ignoring the 3 lowest values, and making
            an average of the two highest ones. A
            list like (0, 0, 1, 0, 0) is calculating
            the median, while (0, 0, 0, 0, 1) is
            taking the minimum. Notice that the sum
            of weights is required to be 1.
        </description>
    </documentation>
    <xmcd tag="alternativesValues">
    <![CDATA[
        <alternativesValues>
            <alternativeValue>
                <values>
                    <value>
                        <real>[...]</real>
                    </value>
                    <value>
                        <integer>[...]</integer>
                    </value>
                    [...]
                </alternativeValue>
            </alternativesValues>
        ]]>
    </xmcd>
</input>
<input displayName="performanceTable"
    name="performanceTable" id="input3" isoptional="0">
    <documentation>
        <description>
            The performance table will contain only
            string values, which correspond to the
            identifiers of the labels defined in the
            domain of the linguistic variable.
        </description>
    </documentation>
    <xmcd tag="performanceTable">
    <![CDATA[
        <performanceTable>
            <alternativePerformances>

```

```

        <alternativeID>[...]</alternativeID>
        <performance>
            <criterionID>[...]</criterionID>
            <value>
                <label>[...]</label>
            </value>
        </performance>
        [...]
    </alternativePerformances>
    [...]
</performanceTable>
]]>
</xmcd>
</input>
<input displayName="fuzzyNumbers" name="fuzzyNumbers"
id="input4" isoptional="0">
<documentation>
    <description>
        This file contains the definition of the
        fuzzy sets associated to the linguistic
        variable that is used for all the
        criteria. The semantics of the linguistic
        labels are given by a trapezoidal
        membership function, that is represented
        with a tuple of 4 pairs values
        ((x1,y1),(x2,y2),(x3,y3),(x4,y4)). The
        first value corresponds to the abscissa
        point x, and the second to the ordinate
        value, y. If x2=x3 the fuzzy set A is
        triangular. The values must be ordered on
        the X axis, so that x1<x2<x3<x4, whereas
        the ordinates correspond to the
        membership degree for each point, so they
        must be concave function. The maximum
        value of y is 1.
    </description>
</documentation>
<xmcd>
    <categoriesValues>
    <categoryValue>
    <values>
        <value id="...">
        <fuzzyNumber>
            <trapezoidal>
                <point1>
                <abscissa>
                    <real>...</real>
                </abscissa>
                <ordinate>
                    <real>...</real>
                </ordinate>
                </point1>
                [...]
            </trapezoidal>
        </fuzzyNumber>
        </value>
        [...]
    </values>
    </categoryValue>
    </categoriesValues>
</xmcd>

```

```

</input>

<output displayName="alternativesValues"
  name="alternativesValues" id="output0">
  <documentation>
    <description>
      Result obtained from the ULOWA
      aggregation on each alternative. It will
      be a linguistic label belonging to the
      domain defined in the fuzzyNumbers input
      file.
    </description>
  </documentation>
  <xmcda tag="alternativesValues"/>
</output>

<output displayName="messages" name="messages"
  id="output1">
  <documentation>
    <description>A status message.</description>
  </documentation>
  <xmcda tag="methodMessages"/>
</output>

  </parameters>
</program_description>

```

5 – Conclusions

The development of this project has been an adventure full of experience and where the acquisition of new knowledge has been a continuous act.

This project has two distinct parts, the first was based on the development of a manual for students of the subject SAD, and the second was to collaborate with another project to improve it. Each part made it more interesting.

When you spend a year working on this type of work, you expect your work to have a utility in the future, and in this project we have done it twice.

This project has gone beyond my prospects. It has been a big deal to develop a project with such features in a foreign language.

Another important aspect has been to develop a project at European level. I am very pleased to have collaborated in crucial aspects of the project, not just at the development of the two modules to be integrated but also in helping to generate a standard language such as XMCDA.

On the one hand, this work has been useful to apply the knowledge acquired during the degree, especially in related subjects such as: IA, PR and ENS. On the other hand, the acquired knowledge during the project will be useful to me in the Artificial Intelligence Master (that I want to start next year), as well as being capable of carrying out such a difficult project.

To conduct this work, several goals were formulated and now we can say that they have been completely accomplished:

- Initial phase of learning concepts needed to the correct development of the project such as:
 - Weighted Sum.
 - ELECTRE.
 - OWA.

- Generating a research report (Canals & Valls, Including ordered weighted averaging operators in Decision Deck, 2011) which will be used as a practical manual for the course: SAD. In fact, it has been already used current course as well as the examples prepared to use in the laboratories.
- Development of two modules for Diviz software that were presented at the 8th Decision Deck Workshop at the University of Corsica - Pasquale Paoli, Corte, France, on April 13th, 2011.
- Extension of the XMCDa language.

During the development of this job have raised new ideas to consider for further projects about Diviz:

- A weight generator that generates a file from a policy (average, most, worst, etc.) or value. This file could be introduced into other modules.
- An evaluation system of weights that from a weights file evaluates its characteristics and gets some conclusions explained at section 3.1.2 – Characterize a weight vector.

Bibliography

- Beliakov, G., Pradera, A., & Calvo, T. (2007). *Aggregation Functions: A Guide for Practitioners*. Berlin: Springer-Verlag.
- Bigaret, S., & Meyer, P. (n.d.). Retrieved Noviembre 17, 2010, from http://www.cs.put.poznan.pl/ewgmcda/pdf/SW_diviz.pdf
- Bigaret, S., Meyer, P., & Venezi, T. (2009). *Decision Deck*. Retrieved Noviembre 15, 2010, from Decision Deck: <http://www.decision-deck.org/>
- Bisdorff, R., Meyer, P., & Roubens, M. (2008). *RUBIS: a bipolar-valued outranking method for the choice problem* (Vol. 6). A Quarterly Journal of Operations Research - Springer.
- Bisdorff, R., Pirlot, M., & Roubens, M. (2006). *Choies and kernels in bipolar valued digraphs* (Vol. 175). European Journal of Operational Research.
- C. Dias, L., & Clímaco, J. (2000). *Additive Aggregation with Variable Interdependent Parameters: the VIP Analysis Software* (Vol. 51). Journal of the Operational Research Society.
- C. Dias, L., & Mousseau, V. (2003). *IRIS: A DSS for Multiple Criteria Sorting Problems* (Vol. 12). Journal of Multi-Criteria Decision Analysis.
- C. Dias, L.; Mousseau, V.; Figueira, J.; Clímaco, J. (2002). *An aggregation/disaggregation approach to obtain robust conclusions with ELECTRE TRI* (Vol. 138). (E. J. Research, Ed.)
- Canals, J., & Valls, A. (2011). *Including ordered weighted averaging operators in Decision Deck*. Corte, Corsica, France: Book of abstracts.
- Canals, J., Valls, A., & Marín, L. (2011). *Introduction to Decision Deck-Diviz: Examples and User Guide*. Technical Report at Universitat Rovira i Virgili.
- Chen, S.-J. (2006). *A new similarity measure for generalized fuzzy numbers based on geometric-mean averaging operator*. In Proc. of 16th IEEE International conference on fuzzy Systems.
- Chen, S.-J., & Chen, S. (2003). *Fuzzy risk analysis based on similarity measures for generalized fuzzy numbers* (Vol. 11). IEEE Trans. Fuzzy Syst.
- Damart, S., C. Dias, L., & Mousseau, V. (2006, August). *Supporting groups in sorting decisions: Methodology and use of a multi-criteria aggregation/disaggregation DSS* (Vol. 43). Decision Support Systems.
- Domingo-Ferrer, J., & Torra, V. (2003). *Aggregation Operators for Prototype Construction in Ordinal Scale* (Vol. 18). Int. J. Intell. Syst.
- Figueira, J., Greco, S., & Ehrott, M. (2005). *Multiple criteria decision analysis: State of the art survey*. Springer.
- Figueira, J.; Greco, S.; Ehrott, M. (2005). *Multiple criteria decision analysis: State of the art survey*. United States of America: Springer.
- Greco, S., Matarazzo, B., & Slowinski, R. (2001). *Rough sets theory for multi-criteria decision analysis* (Vol. 129). European Journal of Operational Research.

- Greco, S., Mousseau, V., & Słowiński, R. (2008). *Ordinal regression revisited: multiple criteria ranking using a set of additive value functions* (Vol. 191). European Journal of Operational Research.
- Herrera, F., & Martínez, L. (2000). *A 2-Tuple fuzzy linguistic representation model for computing with words* (Vol. 8). IEEE Trans. Fuzzy Syst.
- Herrera, F., Herrera-Videman, E., & Verdegay, J. (1996). *Direct approach processes in group decision making using linguistic OWA operators* (Vol. 79). Fuzzy Sets and Systems.
- J. Dujmovic, J. (2007). *Continuous Preference Logic for System Evaluation* (Vol. 15). IEEE T. Fuzzy Systems.
- Merigó, J. (2008). *Nuevas extensiones a los operadores OWA y su aplicación en los métodos de decisión*.
- Merigó, J., & Gil-Lafuente, A. (2009). *The induced generalized OWA operator* (Vol. 179). Information Sciences.
- Meyer, P. (2007). *Progressive Methods in Multiple Criteria Decision Analysis*. PhD thesis. University of Luxembourg.
- Perny, P.; Roy, B. (1992). *The use of fuzzy outranking relations in preference modelling* (Vol. 49). Fuzzy sets and Systems.
- Roy, B. (1974). *Analyse et choix multicritère*.
- Roy, B. (1991). *The outranking approach and the foundations of ELECTRE methods. Theory and Decision*. Springer Netherlands.
- Roy, B. (1996). *Multicriteria Methodology for Decision Analysis*. Kluwer Academic Publisher.
- Rui Figueira, J.; Greco, S.; Słowiński, R. (2009). *Building a Set of Additive Value Functions Representing a Reference Preorder and Intensities of Preference: GRIP Method* (Vol. 195). European Journal of Operational Research.
- Torra, V., & Narukawa, Y. (2007). *Modeling decisions: information fusion and aggregation operators*. Springer.
- Xu, Z. (2006). Induced uncertain linguistic OWA operators applied to group decision making. *Information Fusion*, 7, 231-238.
- Yager, R. (1988). *Fusion of ordinal information using weighted median aggregation* (Vol. 18). Int.J. Approx. Reason.
- Yager, R. (1988). *On ordered weighted averaging aggregation operators in multi-criteria decision making* (Vol. 18). IEEE Trans. Syst., Man, Cybern.
- Yager, R. (1993). *Families of OWA operators* (Vol. 59). Fuzzy Sets and Systems.
- Yager, R. (1996). *Quantifier guided aggregation using OWA operators* (Vol. 11). International Journal of Intelligent Systems.
- Yager, R. (2002). *Heavy OWA Operators* (Vol. 1). Fuzzy Optimization and Decision Making.
- Yager, R. (2004). *OWA aggregation over a continuous interval argument with applications to decision making* (Vol. 34). IEEE Trans Syst Man Cybern B Cybern.