Nantes | Hamburg | Helsinki

mySMART Life

Deliverable due date: M36 – November 2019

D2.9 DECISION AIDING TOOL

WP2, Task 2.5, Subtask 2.5.3

Transition of EU cities towards a new concept of Smart Life and Economy

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

Project Acronym		mySMARTLife
Project Title		**Transition of EU cities towards a new concept of Smart Life and Economy**
Project Duration		1st December 2016 – 30th November 2021 (60 Months)
Deliverable		D2.9 Decision Aiding Tool
Diss. Level		Public
Status		Working
		Verified by other WPs
	■	Final version
Due date		30/11/2019
Work Package		WP2
Lead beneficiary		ARMINES [ARM]
Contributing beneficiary(ies)		IMT Atlantique (ARMINES' linked third party)
Task description		A Multicriteria Decision Aiding (MCDA) technique will be designed and deployed in the Urban platform by ARM to provide decision makers with tools to advance in solving a decision problem where several, often contradictory, points of view (criteria) must be taken into account. Adaptation of MCDA algorithms available via the diviz ecosystem (http://www.diviz.org) and the XMCDA web-services (http://www.decision-deck.org/ws) will be part of this subtask.

Date	Version	Author	Comment
16/10/2019	1.0	Patrick Meyer (ARM)	First version
19/11/2019	1.1	Suvi Takala (HEN)	Review
22/11/2019	2.0	Patrick Meyer (ARM)	Integrating review comments and monitoring section
22/11/2019	2.1	Guillaume Chanson (NM)	Comments on TRL and Monitoring
25/11/2019	3.0	Patrick Meyer (ARM)	Integration of final comments from NM

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

Copyright notices

Table of Content

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

Table of Figures

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 731297.

Table of Tables

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

Abbreviations and Acronyms

Acronym	Description
mySMARTLife	Transition of EU cities towards a new concept of Smart Life and Economy
MCDA	Multi-Criteria Decision Aiding
DM	Decision Maker
MAVT	Multi-Attribute Value Theory
XML	A general-purpose syntax for defining markup languages
XMCDA	A data standard to share MCDA concepts and data
DH	District Heating

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

1. Executive Summary

The goal of task “Decision Aiding Tool” (D2.9) is to develop a Multi-Criteria Decision Aiding (MCDA) tool to support decision-makers in smart cities.

To this end, we propose a two-part approach:

First a generic state of the art decision aiding process, guided by a so-called analyst, which when instantiated to real–world decision problems, allows to structure and therefore to make intelligible very complex decision problems, involving multiple decision-makers and criteria,

Second a set of software tools to support the actual decision aiding process. The first one, diviz, is a software workbench which allows the analyst to design complex MCDA algorithms, executed as web-services in the cloud. The second one, my.diviz, plays the role of a human-friendly interface between the decision-makers and diviz, and allows to be tuned and designed according to the decision problem’s characteristics, and the decision-makers’ requirements.

In this deliverable we detail both elements of the proposed approach. First we introduce the reader to Multi-Criteria Decision Aiding, its main challenges, as well as the multiple high-level steps of the decision aiding process. Then, we present the two software support tools which can be used by the analyst and the decision-maker to simplify complex decisions. Finally, to illustrate the usefulness of our approach, we apply it on a use-case of the mySMARTLife project, namely the optimization of the district heating network of Nantes. We show how this optimization problem can be seen as an MCDA problem, and how the decision aiding process, as well as the software support tools, allow the 2 involved decision-makers to reach a compromise solution for improving part of the district heating network.

2. Introduction

2.1 Purpose and target group

The goal of this task is to develop Multi-Criteria Decision Aiding (MCDA) techniques to support decision-makers facing complex decision problems in smart cities.

We propose to implement a state of the art *decision aiding process* to help decision-makers to make more informed decisions, by taking into account human, organizational and social factors, the preferences of different stakeholders, and by managing the constraints and requirements of multiple decision-makers.

Very generally, in the smart cities context, the purpose of this decision aiding process is to solve decision problems in which:

different solutions (technical choices, design scenarios, …) have to be assessed or ranked,

- it is important to give arguments for the choice of one of the solutions, or explain their overall assessment,
- multiple actors and criteria have to be taken into account (energy saving, cost, benefit, citizen's comfort and acceptability, …)

The target group of this deliverable is therefore very large and contains decision-makers from private companies, public institutions or agencies, and even final customers.

Typical decision problems in the smart cities context that we can target are:

- evaluation of technical solutions by taking into account various criteria and actors (solar panel potential of roofs, wind potential, …),
- improvement of complex systems through the evaluation of multiple modification scenarios, by taking into account multiple criteria (economical, ecological, social) (improvement of a district heating network, a bus network / line, …),
- assessment of locations for the setting up of a new technology (charging stations for cars / busses / bicycles, urban windmills, …) with respect to their potential and the acceptability by customers,
- urban planning assistance by evaluating various scenarios involving multiple criteria and stakeholders,
- territorial risk assessment w.r.t. various indicators / points of views / regulations
- …

This decision aiding process needs to be supported by *decision aiding algorithms* and consequently *decision support software tools*. We propose in this task to support the proposed decision process with two tools:

- the diviz software workbench, which allows to design complex decision aiding algorithms and to deploy them in institutions and companies,
- the my.diviz software, which allows to generate high level dashboards, tuned to the needs of the decision-makers, and the underlying decision problem, and which calls diviz for the calculation of the decision recommendation.

We therefore propose a two-part solution, made first of the state of the art decision aiding process, and second supported by the two previously mentioned software tools.

2.2 Contributions of partners

The following table depicts the main contributions from participant partners in the development of this deliverable.

Table 1: Contribution of partners

Participant short name	Contributions
ARM / IMT Atlantique	Decision aiding process, development of new XMCDA web-services, customization of my.diviz for the district heating optimization use case

2.3 Relation to other activities in the project

The following table depicts the main relationship of this deliverable to other activities (or deliverables) developed within the mySMARTLife project and that should be considered along with this document for further understanding of its contents.

Table 2: Relation to other activities in the project

Deliverable Number	Contributions
D2.10	This deliverable provides the overall description of the district heating optimization use-case which is presented in Section 6 of the present deliverable.

3. Analysis of the problem

The migration from a traditional city to a smart city, or simply the daily management of a smart city, generate many complex decision problems, involving a large number of actors and factors to be taken into account. Such decision problems cannot solely take into account more or less objective indicators, but require often social or economic perspectives, as well as the difficult to model points of view of the human stakeholders of the problem.

In smart cities, the citizen plays a central role. The city must be at his / her service, and the services he / she uses must contribute to his / her well-being. It is therefore important that decisions made to make a city smarter are made in consultation with the actors involved.

An important objective of smart cities is to reduce their carbon footprint. Therefore, it is also important to take into account the ecological impacts of different choices in decision-making. However, these impacts are often in conflict with economic or even welfare criteria.

All in all, considering multiple conflicting factors, while taking into account various stake-holders' perspectives, make decision problems in smart cities complex and challenging. If one adds to that the fact that decision aiding tools should not be black boxes, in order to allow for some transparency of the decision aiding process, and to maintain a high level of interpretability of the decision recommendation, then these decision problems require techniques from a sub-discipline of Operations Research, namely Multi-Criteria Decision Aiding.

4. Multi-Criteria Decision Aiding process for decision problems in smart cities

4.1 General concepts and vocabulary

Helping to make more informed decisions, taking into account human, organizational and social factors in decision-making issues, formalizing the preferences of different stakeholders and being able to deal with multiple points of view in the decision-making process, managing the constraints and requirements of multiple decision-makers, etc. are all subjects that are part of the concerns of the field of Multi-Criteria Decision Aiding (MCDA).

MCDA is a branch of operational research, and aims to help one or more decision-makers to prepare and make decisions, when several points of view, most often conflicting, must be taken into account. Since the objective of this assistance is not to force decision-making at any cost or to replace the decision-maker(s), MCDA can in certain situations only be used to structure the decision problem for a better understanding, or go as far as the calculation of a decision recommendation.

The decision issues for which such multi-criteria support can be considered relate to complex day-to-day decisions, such as managerial decisions in organizations, risk management in industry, supplier selection, tender response assessment, hospital emergency triage decisions, complex system design, choice of investment strategies, etc.

The idea that mathematics should be used to serve knowledge in its broadest sense is widespread among scientists. It is therefore very tempting to model real-life situations using strong logical principles in order to describe them, explain them, or even predict future events that might result from them. However, when it comes to representing decisions, mathematical models often encounter human irrationality.

This necessary human intervention in decision-making thus represents one of the major difficulties encountered by a researcher trying to provide the decision-maker with recommendation tools. MCDA can therefore be seen as a preparatory *process* for decision-making that attempts to ensure a certain level of objectivity.

This MCDA process generally requires at least 2 actors: on the one hand the *decision-maker* (who can in the most general case be a group of decision-makers), who will be responsible for decision-making and its consequences, and who is governed by a certain *values system* that will have to be clarified and modelled; and on the other hand the *analyst* who has the task of facilitating the decision-making by guiding the decision-maker through the *decision aiding process*. This process is defined by a series of tasks that range from defining and structuring the problem to building a decision recommendation and using an aggregation algorithm. The complexity of this process therefore depends in particular on the context of the decision-making problem and the actors who have an influence on the decision. We will come back to this process in Section 4.5, where we will detail how it can be used to solve decision and assessment problems in a smart cities context.

Traditional operational research methods tend to formulate a decision problem in analytical terms (constraints, objective function …) and propose a solution by seeking an optimal solution. In that case, the decision-maker is involved in the delimitation and formulation of the problem as well as in the validation of the solution. In MCDA, on the other hand, the focus is on the central role of the decision-maker, and particular attention is paid to the inclusion of a representation of his / her value system in the mathematical models. He / she thus intervenes at all stages of the decision aiding process and the resulting recommendation is generally fairly faithful to his / her initial objectives.

Next to that, a lot of problems linked to decision-making often arise from the multidimensional nature of decision alternatives. Indeed, even the seemingly simple decision problems we regularly face are often subject to multiple preference dimensions (cost, quality, ecology, …), which tend to be conflicting (a low cost is generally not compatible with high manufacturing quality). When it comes to choosing the "best" option, MCDA proposes to seek a good compromise, which takes into account the decision-maker's preferences, in order to facilitate the adoption of this recommendation.

4.2 Two methodological schools

From a historical point of view, the origins of MCDA go back at least to the 18th century, when the Marquis de Condorcet was the first to systematically apply mathematical theories to social sciences. In 1785 he wrote his book “Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix”[1], which deals with decision-making in the presence of several voters. However, the foundations of modern MCDA were laid in the mid-20th century with the revealed preference theory of (Samuelson, 1938), the beginnings of game theory by (von Neumann & Morgenstern, 1944), the emergence of social choice theory by (Arrow, 1951) and research on the psychological and mathematical aspects of decisions by (Luce & Raiffa, 1957) and (Fishburn, 1970).

In the 1950s, an important step was taken by (Simon, 1957) towards a pragmatic foundation of decision theories with the concept of limited rationality. It states that in a real life decision, various factors limit a decision-maker's ability to make a fully rational choice. As a result, the decision-maker is subject to limited rationality and will choose an option that takes into account both his limitations on knowledge and those on his cognitive abilities. From a mathematical point of view, this decision will not necessarily be optimal, but will satisfy the decision-maker.

At the end of the 1960s, the first methods for solving multidimensional decision-making problems began to appear. In 1968 (Roy, Classement et choix en présence de points de vue multiples (la méthode ELECTRE), 1968) created the branch of *outranking* methods while in 1976 (Keeney & Raiffa, 1976) extended the theory of *utility* (or *value*) to the multidimensional case. These two schools of thought will generate two different methodological conceptions for MCDA: the *European school* and the *Anglo-Saxon school*.

[1] Essay on the Application of Analysis to the Probability of Majority Decisions

This project has received funding from the European Union‘s Horizon 2020 research and innovation programme under grant agreement No 731297.

From a methodological point of view, the mathematical tools generated by the two currents differ greatly. On the one hand, the European school has developed around the concept of *outranking relation*, where the decision recommendation is built on the basis of *pairwise comparisons* of the different decision alternatives. On the other hand, the Anglo-Saxon school is based on the notion of *utility* or *value* in a general framework called Multi-Attribute Value Theory (MAVT), which seeks to obtain, by aggregation, a *total comparability* of alternatives.

4.3 Alternatives, criteria and preferences

To understand the different MCDA techniques which we deploy in this task, and the decision aiding process that we promote, let us be a little more formal. Let *X* be the finite set of *n* decision *alternatives*. They represent the possible options on which a decision-maker must make a decision (for example, investment options, designs for a complex system, projects which need to be assessed, etc.). The definition of this set is generally quite delicate, and should allow the decision-maker to better understand the decision problem, by clarifying its contours. (Vincke, 1989) observes that “X does not generally impose itself as an objective reality that is easy to define”. It follows that the modelling choices for *X* have a significant influence on all stages of the decision aiding process.

(Roy, Méthodologie multicritère d’aide à la décision, 1985) proposes that the objective of an MCDA process is to solve one of the following three issues: determine an alternative that can be considered the best (*choice* problem), assign alternatives to predefined categories (*sorting* problem), rank alternatives from the best to the worst (*ranking* problem). These decision problems are mainly distinguished by the fact that judgments are made in an absolute (sorting) or relative manner (choice and ranking).

The *consequences* of implementing an alternative can be multiple. They reflect the diversity of characteristics involved in comparing alternatives (cost, quality of service, ecological impact, …). A *criterion* is mainly a function that summarizes the evaluations of an alternative on various relatively homogeneous consequences that relate to the same general point of view. (Roy & Bouyssou, Aide multicritère à la décision: Méthodes et cas, 1993) thus define a *criterion* as a real value function *g* defined on the *X* set of alternatives and which allows to compare two alternatives *x* and *y* so that :

$g(x) \leq g(y) \Rightarrow x$ is considered at least as good as y by the decision-maker on the considered point of view.

The numerical value taken by an alternative on a criterion is called its *performance* or *evaluation*, and the table containing all the evaluations of the alternatives on all the considered criteria is called *performance table*.

The existence of multiple criteria implies that the comparison of alternatives has to be made with respect to the decision-maker's point of view (it is in general impossible to “optimize” simultaneously multiple criteria, without introducing some trade-offs between them). It is therefore essential to take into account his / her value system and objectives through what is called the modeling of his / her *preferences*. To do this, binary relationships are usually used to represent how two alternatives compare. The classical theory based on the concept of utility or value (Fishburn, 1970) distinguishes two different relations for this comparison, *indifference* and *preference*, both of which are considered *transitive*. The European School of MCDA has enriched this dichotomy by introducing a relations of *incomparability* (impossibility for the decision-maker to decide) and *weak preference* (hesitation between indifference and preference) (Roy & Vanderpooten, The European school of MCDA: Emergence, basic features and current works, 1996), none of the relationships being supposed to be transitive.

4.4 Aggregation methods

As already mentioned, an MCDA process guided by the analyst is generally intended to provide a recommendation to the decision-maker. In the multidimensional context that we introduced earlier, this implies that an *aggregation procedure* or *method* must be implemented in order to synthesize the preferences expressed by the decision-maker on each of the criteria. (Mousseau, 2003) defines such a *multi-criteria aggregation procedure* by a rule or process that makes it possible to establish, on the basis of the performance of alternatives on the criteria and preferential parameters, one or more *relational systems* to compare the alternatives of *X*. The use of this relational system then makes it possible to generate a decision recommendation.

The values of the *preferential parameters* are determined via a so-called *preference elicitation process*. It consists of an interaction between the analyst and the decision-maker and leads the latter to express information on his preferences as part of the aggregation procedure that has been chosen to model them. This information is translated into a set of plausible values for the preferential parameters of the aggregation procedure. The preferential information that an analyst can expect from the decision-maker can generally take two forms. On the one hand, this information that can be described as *direct* when it concerns, e.g., on non-significant differences of evaluations, coalitions of criteria sufficient to make a comparison between two alternatives or substitution rates between two criteria. On the other hand, preference information is referred to as *indirect* when the decision-maker refers to the expected results of the decision model by comparing two alternatives with each other (ranking or choice problem) or by giving assignment examples to categories (sorting problem).

The two main methodological schools of MCDA provide different methods for multi-criteria aggregation. On the European school side, the general principle of the *outranking* methods is to compare alternatives through a *majority rule* based on differences in assessments that are compared to *discrimination thresholds* (indifference, preference, veto) for each of the criteria. An aggregation of these local comparisons is then carried out, in which each of the criteria has a *voting power* that is expressed by an importance coefficient (or weight). The operationalization of these pairwise comparisons can be done in different ways (Roy & Bouyssou, Aide multicritère à la décision: Méthodes et cas, 1993), (Brans & Mareschal, 2002), (Bisdorff, Meyer, & Roubens, 2008) and leads to the concept of *outranking relation*. As this relation is generally not transitive, a second stage is necessary in order to reach a recommendation for a decision.

On the Anglo-Saxon school side, the goal is to build a numerical representation of the decision-maker's preferences on *X* through a *value function* $U: X \rightarrow \mathbb{R}$ such that:

$$x \text{ is at least as good as } y \iff U(x) \geq U(y), \forall x, y \in X.$$

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

Again, the definition of *U* can take different forms, such as a weighted sum, the Choquet integral (Grabisch, 1997) or the additive value function (von Winterfeldt & Edwards, 1986). The preference relation induced by this value function is necessarily a preorder, and its exploitation for recommendation is thus quite easy (choice, ranking, sorting).

This introduction defines the main concepts of Multi-Criteria Decision Aiding and illustrates the complexity of solving real-world decision problems.

4.5 Decision aiding process in a smart cities context

As mentioned earlier in this section, decisions and the objects they are concerned with may be very diverse. In fact, each of us are faced with a multitude of decisions every day, from which route to take in order to get to work in the morning, to selecting what to have for lunch. There are numerous factors which influence these decisions, such as our preferences, our prior experiences, different constraints, etc. In certain cases, balancing these factors can be difficult.

The previously mentioned decision aiding process therefore serves as an interface between decision makers and the analyst, guiding them in reaching a decision when multiple and often conflicting criteria are involved.

The process generally starts with the analyst and the decision-maker focusing on *defining* the problem, their goals and how the final decision should be reached (Bouyssou D. , Marchant, Pirlot, Tsoukias, & Vincke, 2006). One key aspect of MCDA is that the final decision may not need to be the best possible one, but one that is acceptable by all the stakeholders. Hence, when multiple decision-makers are involved, conflicts need to be handled in order to reach a consensus on the final decision. The term ``decision frame'', used by Tversky in 1981, supports the fact that decision-makers often base their decision on subjective judgments. Furthermore, nowadays real-world decision problems have become increasingly complex.

The following steps have been identified in (Bouyssou D. , Marchant, Pirlot, Tsoukias, & Vincke, 2006) to structure an MCDA process:

- identify the problem
- formulate the problem
- construct the evaluation model
- reach a final recommendation.

Each of these steps contains additional sub-steps, which depend on the many factors that define a decision problem. Figure 1 illustrates a fictive decision aiding process and the complexity of the various steps involved.

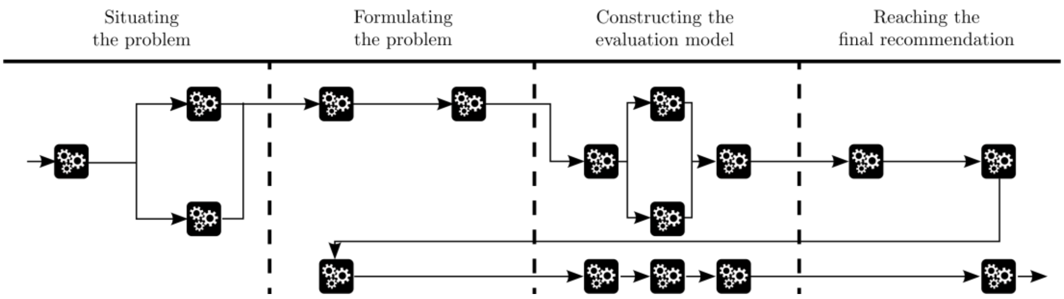

Situating the problem
Formulating the problem
Constructing the evaluation model
Reaching the final recommendation

Figure 1: Fictive MCDA process, with the 4 main steps

More precisely, the step of *structuring* the problem includes sub-steps such as identifying the stakeholders (or actors), identifying the context of the problem, the objectives of the decision and its respective properties. The second step of *formulating* the problem, involves identifying the decision alternatives and their criteria, the type of decision problem, as well as managing multiple decision makers and their different perspectives.

The third step of *constructing the evaluation model* involves the choice of a mathematical representation of the problem and its tuning so that it reflects the perspective of the DM. Furthermore, a resolution method also needs to be selected in order to provide a recommendation to the decision problem. Finally, in the last step, *reaching the final recommendation*, this decision recommendation is presented to the DM, who then either validates the result, asks for additional supporting analyses or revisits previous steps in order to refine the solution.

It is important to underline that the structure of the process is nonlinear, complex and iterative. This means that at any point in the process we may choose different paths to follow, in some cases going back to previous ones. We highlight this, and the fact that the decision aiding process is decomposed into multiple sub-steps, as shown in

. At each of these steps we may have an interaction between the decision-maker and the analyst, the extraction of an important piece of information, the use of an algorithm, or a visual representation of alternative, etc.

In our smart cities context, this decision aiding process can be applied to problems in which:

- different solutions (technical choices, design scenarios, …) have to be evaluated or ranked,
- it is important to give arguments for the choice of one of the solutions, or explain their overall assessment,
- multiple actors and criteria have to be taken into account (energy saving, cost, benefit, comfort of citizens, …)

Typical applications therefore are:

- evaluation of the potential of roofs for solar panels (by taking into account multiple points of view and actors),

- improve the district heating network by evaluating multiple modifications on the network according to multiple criteria (economical, ecological, social), and by taking into account the preferences of various actors,
- evaluate locations for charging stations (cars, busses, cars), with respect to their potential and the acceptability by customers,
- assess various scenarios for urban planning, by taking into account multiple perspectives
- evaluate territorial risks w.r.t. various indicators / points of views / regulations
- multi-criteria evaluation of companies w.r.t. their "greenness", and generate recommendations to become “greener”
- help the customer to choose the best service provider for his need (solar panels on the roofs, ...), according to his constraints (budget, …) and point of view (ecological, social, …)
- …

This list is clearly not exhaustive, and as we have shown in this section, MCDA techniques can be applied to a lot of assessment and decision problems, when multiple, conflicting points of view have to be taken into account simultaneously.

To apply the MCDA process to such real-life problems, the analyst usually uses a *decision aiding software*. The next section deals with this topic, and with our my.diviz solution, which generates software allowing the decision-maker to understand the underlying decision problem and to interact with a decision aiding algorithm, while having a software totally configured to his / her needs and the specificities of the decision problem.

5. Decision aiding software for supporting decision problems in smart cities

5.1 State of the art and presentation of the diviz workbench

Many so-called MCDA methods have been proposed in the literature and are very often available as software programs (see (Weistroffer, Smith, & Narula, 2005) for a study on existing MCDA software).

In other scientific research fields, as, e.g., statistics or machine learning, there exist software platforms which allow to easily compare different analysis methods and to test them on a given data set inside a common framework. Among the most famous ones, one can cite platforms such as the GNU R statistical system or the Weka suite of machine learning software. Both of these workbenches are open-source and independent from the operating system, which has certainly contributed to their large dissemination and acceptance among many researchers and users.

Unfortunately, at least four major difficulties arise when it comes to using MCDA programs in practice in a complex decision aiding process as the one which might occur in a smart cities context:

- different MCDA techniques are generally implemented in separate software products, with heterogeneous user interfaces;
- testing multiple MCDA algorithms on one problem instance is not always easy, because of the various input data formats required by the multiple software applications,
- a lot of MCDA algorithms which are presented and published in scientific articles are not easily available and consequently often only used by their authors, which blocks their use in industrial applications,
- several MCDA software products are not *free* (neither from a financial, nor from an open-source point of view), which can be considered as a weakness for their large dissemination, in particular in industrial projects.

To overcome these difficulties, the *Decision Deck Consortium*, in collaboration with *IMT Atlantique*, has developed a coherent ecosystem of tools called *diviz*.

It is a software workbench which facilitates the use of algorithmic resources from the field of Multi-Criteria Decision Aiding and can be used to support the complex MCDA process presented earlier in Section 4.5. As such, diviz's purpose is to provide effective tools for at least three types of users:

- *MCDA consultants* who wish to use MCDA tools to support actual decision makers involved in real world decision problems;
- *teachers* who present MCDA algorithms in courses;

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

- *researchers* who want to test, disseminate and compare algorithms or to develop new ones.

The diviz tool is an easy to use software to build, execute and share complex workflows of MCDA algorithms. One of the main features of diviz is that it facilitates the construction of the previously described classical MCDA methods, as well as derivatives and original ones, by combining various elementary calculation components in so-called *algorithmic workflows* (shown in the upper part of Figure 2) via an intuitive graphical user interface. This flexibility and adaptability are great advantages when it comes to answering the complex challenges raised by industrial applications, which justifies its use as an algorithmic engine for the mySMARTLife project.

The diviz workbench is integrated in an ecosystem of other initiatives:

- XMCDA: a standardized XML recommendation to represent objects and data structures coming from the field of MCDA. Its main objective is to allow different MCDA algorithms to interact and to analyze a problem instance stored in XMCDA by various MCDA algorithms;
- XMCDA web-services: distributed open-source computational MCDA resources.

Figure 2: Screenshot of the diviz workbench

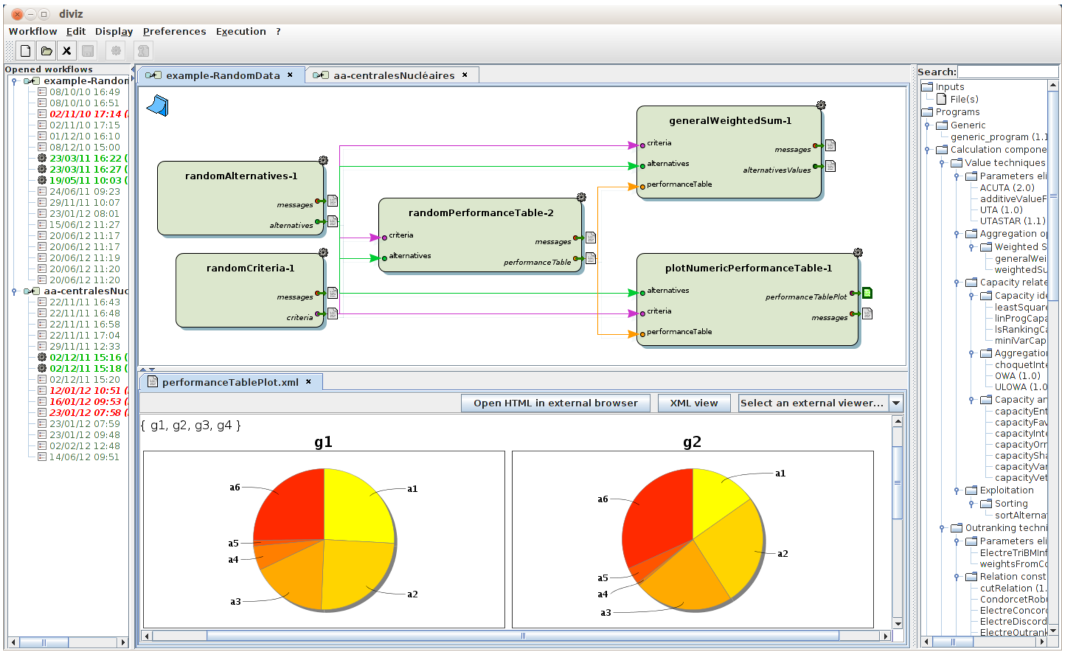


Figure 2 shows the diviz workbench. On the left side, a tree presents the list of the opened workflows, along with current and past execution results. The upper-middle panel shows the currently selected workflow, while the lower-middle panel shows the results of one of the components. On the right side, all available programs are organized by themes (e.g. value-based, outranking, …).

MCDA workflows are built by dragging and dropping the needed calculation components from the right onto the middle panel. These workflows can then be easily executed or shared with other diviz users.

As already mentioned above, in order to be interoperable, the various algorithmic components available in diviz use a common data standard, XMCDA, which allows to represent concepts and data structures coming from the field of MCDA. XMCDA is written in XML, a general-purpose syntax for defining markup languages whose purpose is to aid information systems in sharing structured data, especially via the Internet and to encode documents. XMCDA is defined via an XML Schema, a set of syntax rules and constraints which define its structure.

Each tag of an XMCDA file describes data related to a decision aiding problem. To summarize, these tags can be put in five general categories:

- description of the current decision aiding problem;
- description of the standard MCDA concepts like criteria, alternatives or categories (described in Section 4.3);
- the evaluations of the alternatives on the criteria in the so-called performance table;
- preference information related to criteria, alternatives, or categories (either provided as input by a decision maker or produced as the output of an algorithm);
- output messages from algorithms (log or error messages) and input information for algorithms (parameters).

The various MCDA calculation elements which are available in diviz are the XMCDA web-services proposed by Decision Deck and maintained by IMT Atlantique. From a general point of view, a web-service is an application which can be accessed via the Internet and is executed on a remote system. One of the great advantages of such online programs is their availability to anyone at any time and any place and on any computer which is connected to the Internet. From a practical point of view, these web-services propose MCDA algorithms, which, if properly combined (in diviz for example), can rebuild MCDA methods. To be interoperable, the inputs and outputs of the XMCDA web-services are formatted according to the XMCDA standard.

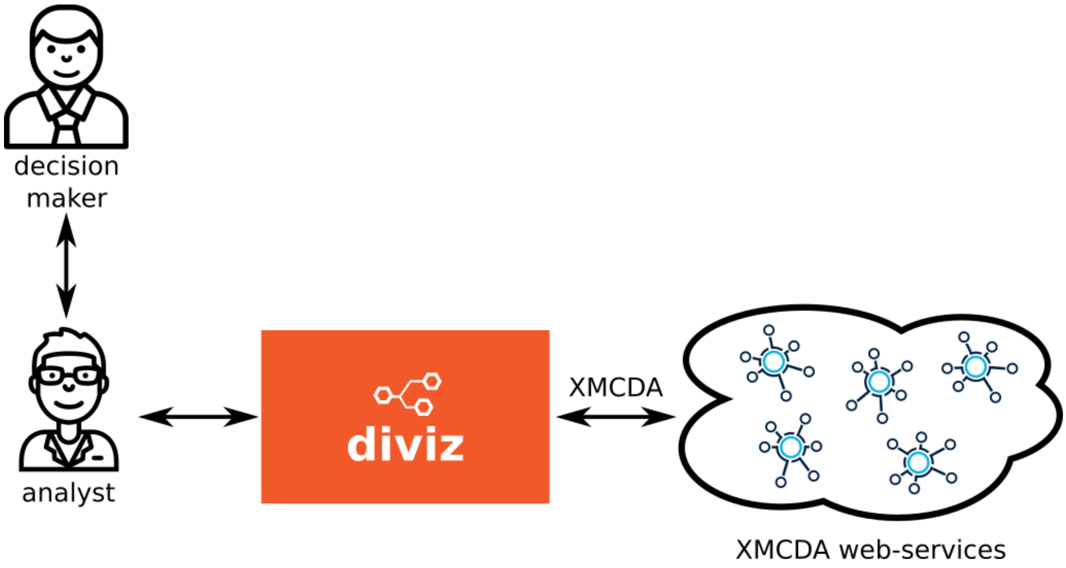
In diviz, once the design of the MCDA workflow is finished, the user (usually the analyst) can execute it in order to obtain a recommendation for his decision problem. Furthermore, after the execution of the workflow, the outputs of each of the components can be viewed and analyzed by the user. Some of these outputs might represent results of intermediate calculation steps of the workflow, which facilitates the tuning of the parameters of the algorithms.

The history of the past executions is also kept in the software and can at any moment be viewed by the user. More precisely, if a workflow is modified, the former executions' results and their associated workflows are still available. This contributes to the good understanding of the constructed chain of algorithms and to a proper calibration of the preferential parameters of the decision situation.

Next to designing and executing MCDA workflows, diviz can also be a convenient tool to compare the outputs of various methods and algorithms on the same input data, or to test the influence of variations in the data on the output recommendation. Indeed, it is easy to connect a data set linked to a specific decision problem to various workflows in a single workspace, each of them representing a different MCDA method, and to compare their outputs. Similarly, variations of the same data can be connected to multiple copies of the same MCDA technique in order to check the robustness of the output recommendation with respect to different input situations.

The diviz software also enables to export any workflow, with or without the data, as an archive. The latter can then be shared with any other diviz user, who can import it into his software and continue the development of the workflow or execute it on the original data. This allows, in a practical context, for the MCDA treatment to be shared among the various stakeholders of the process. Figure 3 summarizes diviz's ecosystem, with the decision-maker and the analyst. The analyst implements the decision aiding process described in Section 4 (arrow representing the dialogue between the decision-maker and the analyst). The analyst uses the diviz workbench to support this decision aiding process. Once the algorithmic workflow is designed, diviz calls the XMCDA web-services to make the calculations in the cloud. The decision recommendation is then presented to the analyst, who, in the decision aiding process discusses this result with the decision maker.

Figure 3: diviz's ecosystem



All in all, these features show that diviz is a very flexible tool and that it can be adapted to various practical decision situations. It is therefore a powerful tool in the context of smart cities, where the possible application domains requiring decision aiding techniques are quite heterogeneous, and might consequently require the use of different MCDA algorithms.

Unfortunately, diviz is clearly not a tool intended for the decision-makers as end users (as shown in Figure **3**, where we have not drawn any arrow between the decision-maker and diviz). It can support the decision aiding process, it can be easily deployed in a company or an institution to make decision aiding calculations, but it is far from a user-friendly software for non-specialists of decision aiding techniques (as the decision-makers).

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 731297.

As the goal of our task is to propose the various stakeholders of smart cities a decision aiding tool, we have identified the need to use a layer above diviz, which would allow decision makers to interact easily, in a human-friendly way, with the decision aiding algorithms. This analysis, performed in the beginning of the mySmartLife project, motivated us to use a supplementary tool developed in IMT Atlantique, namely my.diviz, which we present hereafter.

5.2 my.diviz to ease the interaction between decision-makers and decision algorithms

In a decision problem, a decision-maker requires to understand the underlying problem, in order to be able to make an informed decision. It is therefore important that the decision recommendations provided by a decision aiding algorithm are interpretable, and any black box effect be avoided.

To contribute to this transparency, the decision maker should be able to interact with a software tool which

- summarizes the key elements of the underlying decision problem,
- presents the available decision alternatives and decision recommendation in a synthesized and understandable way,
- if necessary allows him / her to add or modify the data related to the decision problem,
- possibly allows him / her to interact with the algorithm to tune the recommendation to his preferences.

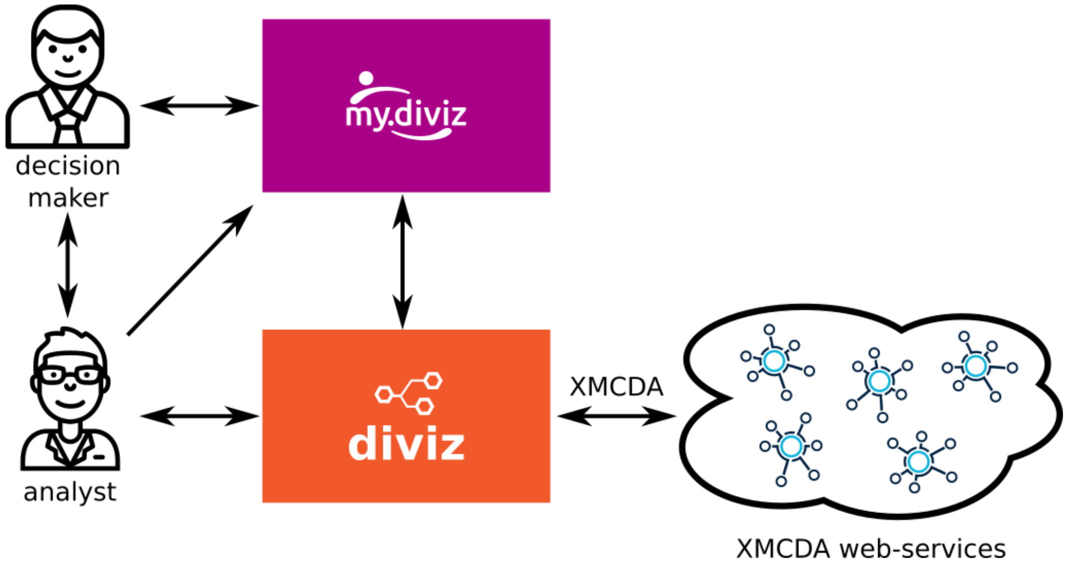
The previously introduced diviz workbench does not answer these requirements (or only partly). We therefore propose to use the *my.diviz* platform developed by IMT Atlantique, as a complement to diviz, which allows to generate human-friendly interfaces to interact with the decision algorithms designed in diviz.

As the extent of the possible decision problems linked to the management of smart cities is possibly huge and their nature very diverse, it is important that the tool with which the decision maker interacts is adapted to his / her specific domain of expertise and application. Also, for the result of our action to be easily replicable in other cities, it is important that the decision support software remains generic and adaptable to many specific needs.

As we will explain in the following sections, with my.diviz the analyst can build a *dashboard* tuned to the decision-maker's needs and the decision problem's specificities. As such it is a very generic tool, whose main goal is to make the decision recommendations more interpretable, while allowing the decision-maker to modify the data or the parameters of the algorithms designed in diviz by the analyst.

The my.diviz software can therefore be considered as a human friendly layer above diviz. Figure 4 summarizes how both tools interact, and how they are positioned with respect to the analyst, the decision-maker and the decision aiding process.

Figure 4: my.diviz in the diviz ecosystem



In the sequel we will detail the software architecture and the technical choices that were made for my.diviz. Upstream, the very nature of the software requires to be established, i.e., to identify the end users and the context in which the software is used. As already mentioned, this software is dedicated to the decision-maker, who is not an expert of decision aiding and the algorithms.

However, as shown on Figure 4, the my.diviz software is linked to the analyst and his expertise.

The role of the analyst is in this case to create the design of the dashboard that the decision-maker has to use to make his decision. Therefore, the analyst, next to his decision facilitator role, plays here also the role of a publisher. He decides on the publication of the data to be displayed for the decision-maker, as well as various graphical components which should facilitate the decision-maker’s understanding of the problem and the decision recommendation. The decision-maker, for his part, interprets the results obtained on the basis of the input data which may, if the analyst authorizes him, be modified.

In any case, both users interacting with my.diviz (the analyst and the decision-maker) do not necessarily come from the IT world. The my.diviz software must therefore be *simple to use* and *easy to generate*.

Moreover, the multi-user aspect is not without problems because the analyst's work determines the decision-maker's work. In order to avoid untimely deliveries to take into account each analyst's new publication, it was decided that my.diviz would be a web platform capable of being updated in real time. The choice of software architecture and technologies is therefore made accordingly.

5.3 Architecture

The software architecture is a three-tier architecture, i.e. it is divided into three layers.

The first one concerns the view part. It refers to the presentation of data. The second layer corresponds to the processing of the said data and finally the third layer is associated with their persistence.

For simplicity's sake, no database is used. Data storage uses xml (XMCDA) and ini files (where each data can be associated with a key).

The use of a 3-layer architecture has many advantages, including speed of development, scalability, performance and availability. As mentioned above, modularizing the different levels of an application allows development teams to develop and improve a product faster than developing a single code base because a specific layer can be upgraded with minimal impact on the other layers. It can also help improve development effectiveness by allowing teams to focus on their core competencies. Many development teams have separate developers who specialize in front-end, back-end server and back-end data development. By modularizing these parts of an application, you no longer need to rely on complete stack developers and can better use the specialties of each team.

Scalability is another great advantage of a 3-layer architecture. By separating the different layers, you can scale them independently of each other according to the needs at a given time. For example, if you receive many web requests but not many requests that affect your application layer, you can scale your web servers without touching your application servers. Similarly, if you receive many large application requests from only a handful of web users, you can expand your application and data layers to meet these requests without affecting your web servers. This allows you to balance the load of each layer independently, improving overall performance with minimal resources. In addition, the independence created by the modularization of the different levels offers you many deployment options. For example, you can choose to host your Web servers in a public or private cloud while your applications and data layers are hosted on site. Or you can host your application and data layers in the cloud while your web servers can be hosted locally, or any combination of the two.

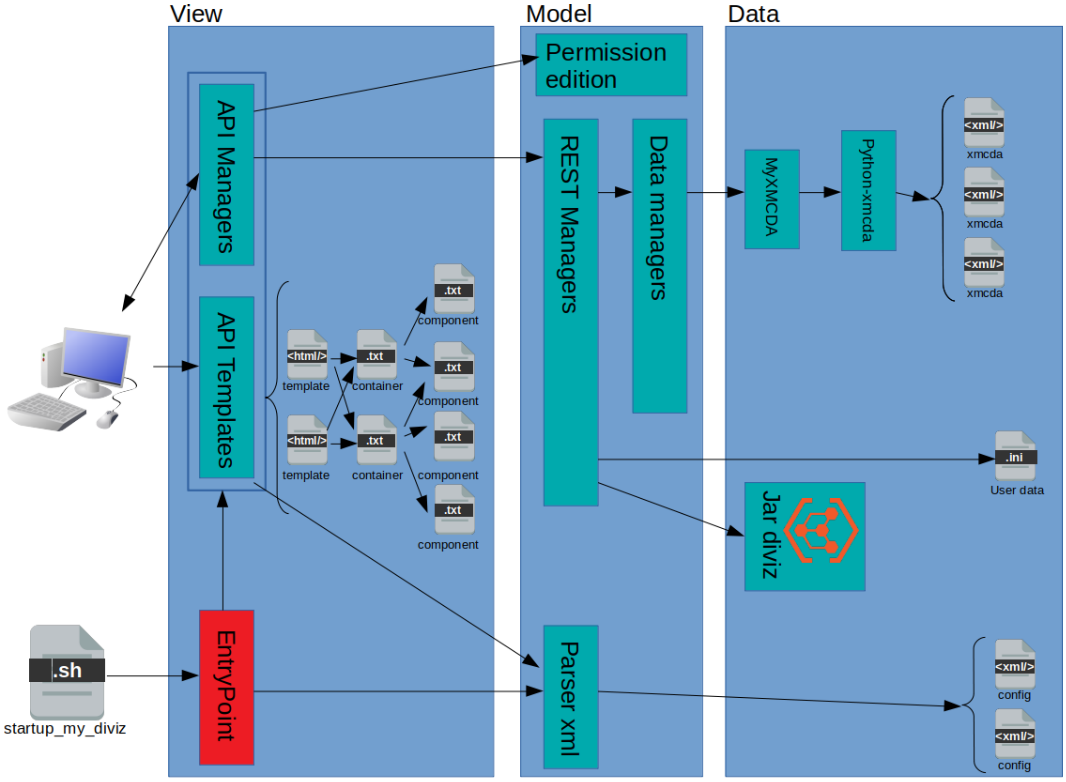
By having disparate layers, you can also increase reliability and availability by hosting different parts of your application on different servers and using cached results. With a full stack system, you need to worry about the failure of a server and the significant impact on the performance of your entire system, but with a 3-layer application, the increased independence created by physically separating the different parts of an application minimizes performance problems when a server fails.

Also, the modular properties allow several developers to work together without interfering with each other's work.

The architecture chosen is therefore in accordance with the initial condition: simplicity. This architecture is presented in Figure 5.

This project has received funding from the European Union‘s Horizon 2020
research and innovation programme under grant agreement No 731297.

Figure 5: 3-tier architecture of my.diviz



On the bottom right of Figure 5 are shown multiple *config.xml* files. They describe the design of the project, based on the requirements of the use-case and the needs of the decision-maker. Each file describes therefore one project.

The starting point of my.diviz is the *EntryPoint.py* class that is triggered via the shell script *startup_my_diviz*. *EntryPoint* creates the different REST APIs that allow the user to interact with the application. There are two types of APIs.

The *API Templates* handle the rendering for the display. Via a parsing of the *config.xml* configuration file of the concerned project, they restore the display with the right template and its containers where the components to be displayed are included.

The *API Managers* are also created on the fly by EntryPoint via a parsing of all config.xml files (one per project) in which the components to be displayed, the templates, etc are indicated. These APIs, do not handle the display but allow the user to interact with the application. They can therefore be used to add or delete data, for example.

When the user decides to add a data element and validates its input fields, an Ajax HTTP request is sent to the server. Thanks to the REST architecture, the HTTP URL indicates which API is being used. The HTTP method (POST, PUT, GET, DELETE...) refers to the function of the "REST manager" used. In other words, the request solicits a specific API that calls a class and its well-defined method (here it is called POST).

In general, the *REST Managers* are used to edit user data. These concern the choices made by the user, such as the positioning of components. They also allow you to launch diviz (which contains the MCDA algorithms to be executed).

The main activity carried out by the *API Managers* concerns mainly the editing of XMCDA data. To do this, it goes through an intermediary: the *Data Managers* who carry out the various data processing operations. For example, the deletion of an alternative must be applied to all the files concerned in both the list of alternatives and the performance table. To do this, the Data Managers use interfaces such as MyXMCDA, which processes data using the python-xmcda library.

Last but not least, an API Manager (still experimental) allows to manage multi editing between several users (decision-makers). It is based on the *websocket* technology, a new standard on the web. When a user modifies a data, he sends a signal to the server, which takes care of temporarily removing the write permissions (the duration is indicated in the project config.xml) from all the others.

5.4 Technological choices

The my.diviz software is free and open source. It will be distributed under the EUPL 1.2 license and is therefore not dependent on any proprietary technology. More generally, the relevance of a technology is established on several criteria:

- Independence: the risk with a technology is that it may become obsolete, which ultimately requires a costly migration to a newer technology. With a web application like this one, the problem is exacerbated because software compatibility depends on the user's technical choices (browser and version used, plug-in installed, etc.). When the user updates his working environment, there is a risk of version incompatibility. In this case, it is necessary: either to update the technology urgently or if it is dead (so no more updates possible) to migrate to a new one. The best strategy is therefore to rely upstream, as much as possible on timeless standards that are understood by all.
- Sustainability: Following on from what has been mentioned above, the technology must have the longest possible lifespan, i.e. have a large community as a support.
- Ease of development: technology must not be an obstacle to development. On the contrary, it must accelerate developments, i.e. be known and understood by developers.
- Execution efficiency: technology must not hinder the proper execution of the tool. The latter must be as light and fast as possible.

Thus, the choice to do open-source software does not consist in systematically privileging ethics over efficiency, quite the contrary. Short-term efficiency is replaced here by a more sustainable development; even if it sometimes means reinventing the wheel, it does not impose the cost of changing it regularly.

Web developments are based on the standards of the World Wide Web Consortium (W3C). The my.diviz software being a web application, the choice of technologies is therefore made in two places: on the server side and on the client side.

5.4.1 Server side

- python 3.6: the python language is chosen for its simplicity and its importance in the world of research. At the beginning of the developments the most advanced python version was 3.6.
- Flask: the Flask framework is open source (BSD license). It's a real Swiss army knife. It incorporates the REST API, SocketIO (Websocket),... It allows a quick and simple development of my.diviz.
- Jinja2 : the template framework Jinja2 is free and above all it has the particularity to be integrated in Flask. It allows you to create templates for the (X)HTML rendering of the web page and the components that accompany it.
- Websocket: on the server side, the websocket is created using the flask_socketio.SocketIO library compatible with the Flask framework.
- xml, xsd: there is no database in my.diviz. Data persistence is achieved by reading and writing xml files (XMCDA, config.xml). Nevertheless, user-specific data, i. e. data concerning the presentation of data such as the layout of components after a drag & drop, are saved in an .ini file and not in an xml file (the same applies to labels associated with a key). An XSD schema defines the structure of the config.xml file, as it allows for greater flexibility than a DTD.

5.4.2 Client side

- XHTML5 : This language is the latest standard of the W3C. The new features make it possible to free oneself from certain technologies such as flash adobe.
- CSS 3: it is a web standard that allows you to influence the graphic rendering of web pages. It is well suited for Responsive Web Design which allows to show the same application consistently on multiple devices (smartphone, tablet, computer, …).
- JavaScript: a standard web language
- jQuery: a JavaScript library that has established itself as web standard by allowing you to develop Javascript faster without having to worry about cross-platform issues, especially for ajax calls. It requires only very few upgrades.
- jQuery-ui: a jQuery user interface library that requires little upgrading. It allows you to quickly achieve effects such as drag & drop.
- Bootstrap 4: a library to create a web interface quickly. No update required. It accelerates the development of Responsive Web Design.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

- chartjs : a free JavaScript library for creating graphics. It is the most successful one on the market.
- Websocket: javascript library to exploit the websocket technology of (X)HTML5.

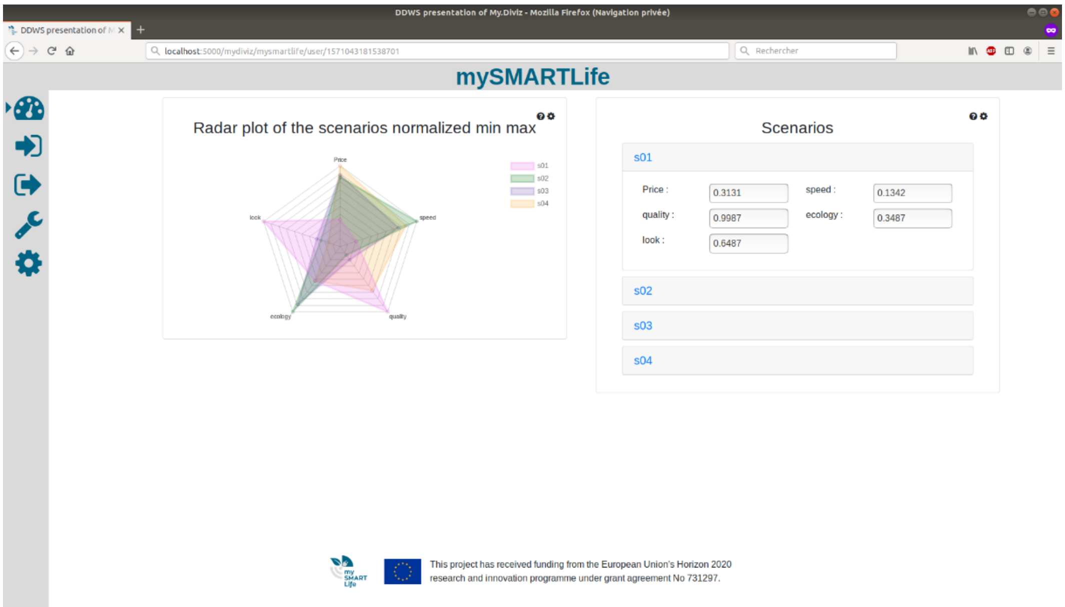
5.5 Design of a my.diviz project

As already mentionned in Section 5.3, the design of a my.diviz project is totally guided by a unique configuration file called *config.xml*. Its content depends merely on two elements :

- The required design of the project, w.r.t. to the decision-maker and the underlying decision problem,
- The algorithmic workflow which was designed in diviz.

Figure 6 shows a screenshot of a my.diviz project.

Figure 6: Screenshot of the dashboard of a my.diviz project


DDWS presentation of My.Diviz - Mozilla Firefox (Navigation privée)

localhost:5000/mydiviz/mysmartlife/user/1571043181538701

mySMARTLife

Radar plot of the scenarios normalized min max

Scenarios

s01

Price : 0.3131
speed : 0.1342
quality : 0.9987
ecology : 0.3487
look : 0.6487

s02

s03

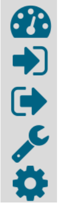
s04

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

On Figure 6 one can see the dashboard of a project. It can contain a summary of the key elements of the project, which are required by the decision-maker to understand the decision problem. These key elements are called “components” in my.diviz. They usually are linked to data of the decision problem.

On the left, the menu of the application is shown. It allows to navigate between the various views of the project. Typically, a my.diviz project contains 4 views, which can be accessed from that menu, also shown in the following figure:

Figure 7: Menu of a my.diviz project



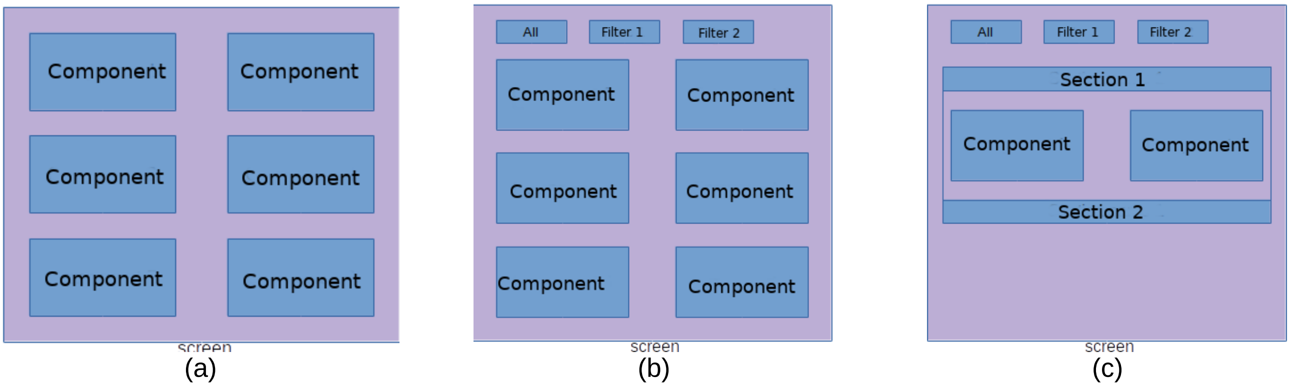
The first icon on the top allows to switch to the *dashboard*. A click on the second icon shows the components related to *input* data. The third menu entry shows the *output* data. The fourth one is used to *configure* the algorithms. And the last one is to *run* diviz in order to update the outputs and to calculate the decision recommendation. To illustrate the design of a project via the *config.xml* file, let us suppose here that an algorithmic workflow has been designed in diviz.

The config.xml file specifies the title (<title>) and the name (<name>) of the application, which template (<template>) should be used, the domain of the application (<domain>), which diviz should be used (<diviz>), the path of the project (<data-path>) and the underlying diviz workflow (<workflow>).

The <edition> and <duration> tags specify the duration of the lock when multiple users want to write some data « simultaneously » in the same application.

The <options> tag contains the elements <dashboard>, <input>, <output> <configuration> and <run> (recall the menu items). Except for <run>, we can specify for each of these menu items the components which should be shown.

At the moment, three ways to show components exist. The first is to put directly <component> tags in the <components> tag. The picture in Figure 8 (a) below shows how the components appear in that case:



(a) (b) (c)

Figure 8: Configuration options for components

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

Second, we can insert filters to mask components when necessary. Instead of writing directly <components> in <screen>, we write <sections filter= " True "><section><component>. We can add as many sections as necessary. In that case, the components look as in Figure 8 (b).

At last, we can use an accordion presentation as shown in the picture below Figure 8 (c). For that, we have just to specify mode=" accordion " in the <sections> tag.

Below, we give an example of a config.xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<gaudi xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="../../templates/config.xsd">
    <title>ddws presentation of my.diviz</title>
    <page-name>ddws presentation of my.diviz</page-name>
    <template>my_smart_life/mySmartLife.html</template>
    <domain>http://localhost:5000/</domain>
    <diviz>diviz-client.test-v3.2019-09-10.jar</diviz>
    <data-path>ddws/</data-path>
    <workflow>zipped_workflow.dvz</workflow>
    <edition>
        <duration>10</duration>
    </edition>
    <options>
        <dashboard>
            <display-name>Dashboard</display-name>
            <screen>
                <components>
                    <!-- add here your components-->
                </components>
            </screen>
            <visibility>True</visibility>
        </dashboard>
        <input>
            <display-name>Entries</display-name>
            <screen>
                <sections>
                    <section ref="entry_basic">
                        <display-name>Section 1</display-name>
                        <components>
                            <!-- add here your components-->
                       </components>
                    </section>
                    <section ref="entry_secondary">
                        <display-name>Section 2</display-name>
                        <components>
                            <!-- add here your components-->
                        </components>
                    </section>
                </sections>
            </screen>
            <visibility>True</visibility>
        </input>
        <output>
            <display-name>Results</display-name>
            <screen>
                <sections>
                    <section ref="output_basic">
                        <display-name>Plots</display-name>
                        <components>
                            <!-- add here your components-->
                        </components>
                    </section>
                    <section ref="output_secondary">
                        <display-name>Data</display-name>
                        <components>
                            <!-- add here your components-->
                        </components>
                    </section>
                </sections>
            </screen>
            <visibility>True</visibility>
        </output>
        <configuration>
            <display-name>Configurations</display-name>
            <screen>
```

This project has received funding from the European Union‘s Horizon 2020 research and innovation programme under grant agreement No 731297.

```
                <sections>
                    <section ref="Config1">
                        <display-name>Configuration 1</display-name>
                        <components>
                <!-- add here your components-->
                        </components>
                    </section>
                    <section  ref="Config2">
                        <display-name>Configuration 2</display-name>
                        <components>
                            <!-- add here your components-->
                        </components>
                    </section>
                </sections>
            </screen>
            <visibility>True</visibility>
        </configuration>
        <run>
            <display-name>Execute</display-name>
            <visibility>True</visibility>
        </run>
    </options>
</gaudi>
```

As shown by the comments *<!-- add here your components-->* no component has yet been added in this config.xml file.

To create the dashboard of Figure 6, we have to add two components: a radar chart of the performance table, and an accordion presentation of the performance table.

For that, the following code has to be inserted into the config.xml file in the <components> section of <dashboard>:

```
<component ref="mcda_component1">
    <display-name>Radar plot of the scenarios normalized min max</display-name>
    <modality>read</modality>
    <authorized>all</authorized>
    <display>get_radar_widget/</display>
    <template>resultats/radar-widget-real-2.txt</template>
    <panel-configuration>configuration/configuration-component.txt</panel-configuration>
    <panel-help>header-help.txt</panel-help>
    <parameters>
        <parameter name="datas_alternative_json" module="MyXMCDA">inputs/alternatives.xml</parameter>
        <parameter name="datas_criteria_json" module="MyXMCDA">inputs/criteria_name.xml</parameter>
        <parameter name="datas_criteria_values_scale_json" module="MyXMCDA">inputs/criteria.xml</parameter>
        <parameter name="datas_performances_table" module="MyXMCDA">inputs/performanceTable.xml</parameter>
    </parameters>
</component>
<component ref="mcda_component2">
    <display-name>Scenarios</display-name>
    <modality>write</modality>
    <authorized>all</authorized>
    <display>get_alternative_criteria/</display>
    <template>resultats/accordion-alternatives-criterias.txt</template>
    <panel-configuration>configuration/configuration-component.txt</panel-configuration>
    <panel-help>header-help.txt</panel-help>
    <manager>
        <manager-name>alternative-manager</manager-name>
        <manager-api>alternative</manager-api>
    </manager>
    <parameters>
        <parameter name="datas_alternative" module="MyXMCDA">inputs/alternatives.xml</parameter>
        <parameter name="datas_criteria" module="MyXMCDA">inputs/criteria_name.xml</parameter>
        <parameter name="datas_performances_table" module="MyXMCDA">inputs/performanceTable.xml</parameter>
    </parameters>
</component>
```

After saving the file, restarting my.diviz and refreshing the browser, the dashboard of the current project changes to look like the one of Figure 6.

Similarly, components are added to the input and output sections of the config.xml file, to create a my.diviz project, and fill the input and output elements of the project. An example of such a project is presented in Section 6 on a district heating optimization use-case.

This small illustration shows the flexibility of my.diviz and the ease to design a project in my.diviz. First of all, the analyst has to design an algorithmic workflow in diviz. Then, once this is done, he / she must identify what inputs and outputs are required by that workflow. In close collaboration with the decision-maker, the analyst then decides what data elements should be shown in the project, how they should be presented to the end-user, which ones should be editable, and which ones not. Once these choices are made, the analyst just has to edit the config.xml, guided by an XML Schema, to design the project in my.diviz. Once the design is ended, the corresponding project can be accessed from a web browser.

5.6 Link to the urban platform

To allow for an easy replicability and to keep the advantage of flexibility, it has been decided in the beginning of the mySMARTLife project to keep the decision aiding tools my.diviz and diviz separated from the urban platform. Both decision aiding software tools rely however on standards, as for example XMCDA, which simplifies interoperability with the urban platform. Next to that, my.diviz is able to import and export data in JSON format.

JSON (JavaScript Object Notation) is a lightweight data-interchange format which is easy for machines to parse and generate. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

This means that for certain criteria which need to be taken into account in the decision support, the urban platform can provide indicators (in JSON format) which contribute to the calculation of these criteria. However, as a certain number of these criteria necessarily rely on external data (social, geographical or demographical data, results of surveys and studies, …), a total integration in the urban platform would have been uninteresting, and even worse, would have contributed to decreasing the flexibility and the genericity of the tools.

6. Application to the optimization of the district heating network

Among the possible decision problems in a smart cities context, we have chosen to work on the optimization of the district heating network, by helping in identifying the “best” improvement scenarios with respect to multiple criteria and multiple stakeholders.

More specifically, in this work we aim to identify the potential of using the heat from the return pipe of existing district heating (DH), by combining a DH simulation tool and a multi-criteria decision aiding algorithm. The methodology is based on the analysis of the measured return temperatures at each sub-station in order to identify those with high potential of efficiency gains by using the return flow (with support of the supply pipe when needed). The candidate substations are determined based on the compatibility between the heat resource time series and the load profiles. The set of eligible substations is used to define scenarios (the alternatives in the decision aiding context) of connection to the return pipe.

The impacts of each scenario on the DH operational performance and the energy savings are evaluated with a dynamic hydro-thermal model. In this simulation tool, all the substations are detailed regarding the temperatures and mass flow rates at the heat exchangers as well as the inertia of heat transport in the pipes due to heat losses.

The technical parameters and the energy efficiency are not the only points of view in the decision making process for the selection of the scenarios “of best compromise” for the improvement of the DH. We propose to implement a complex decision aiding process, involving multiple criteria, dealing with different points of view (economic, energy, technical…) and different decision makers. The evaluations of the scenarios on the criteria are summed up in a so-called performance table and aggregated by an outranking model (MR-Sort) to identify relevant scenarios.

We illustrate this study by the example of a part of the DH in Nantes. The baseline of the actual configuration of the DH is analyzed and compared with the selected scenarios. Note that the modelling of the DH is presented in deliverable D2.10 (District Heating Monitoring and Optimization). We focus here only on the decision aiding part. The interested reader can refer to deliverable D2.10 to understand how the various criteria have been obtained by modelling and simulation.

6.1 The decision aiding process to select improvement scenarios for the district heating in Nantes

In Section 4.5 we presented the general framework of the decision aiding process which can be applied to solve decision problems in a smart cities context. Let us detail each of the steps of this process subsequently.

6.1.1 Identify the problem

Identifying the problem consists in determining the boundaries of the problem, the main actors involved in the problem, as well as first indications on the desired output of the decision aiding task.

We identify two distinct decision makers who are involved in the decision process of this case study: the *user* (the company that operates the secondary side of the heating network) and the *operator* (the company that operates the primary side of the network).

After discussions with them, we determine that the output of the decision aiding task should be a qualitative evaluation of various improvement scenarios of the DH. Their goal is to evaluate each of the scenarios on a two -level qualitative scale (“bad”, “good”). In the MCDA vocabulary, this translates to the fact that each scenario should be assigned to one of the two ordered classes “bad” or “good” (which will restrict the algorithms to “sorting” algorithms).

Nantes’ district heating has been recently extended to link more than 380 substations with about 85 km of pipes in 2017 for only 22 km in 2012. It supplies heat to more than 16 000 dwellings and many public facilities (swimming pools, hospital, schools, museums…). The total capacity of heat production units hits 202 MW with an energy mix achieving 84 % of renewable and waste heat (Table 3).

Table 3: Nantes DH main production units

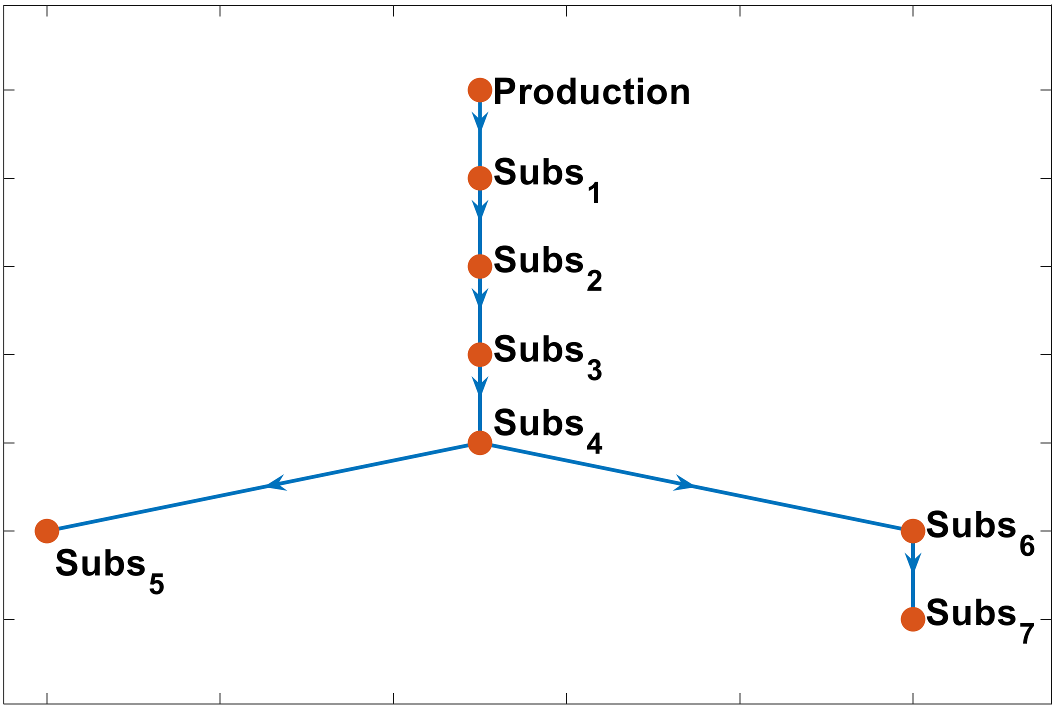
Site	Number of units	Energy source	Capacity (MWth per unit)
Malakoff	2	Waste inceneration	15
	2	Biomass (Wood)	15
	3	Gas	29
La Californie	3	Gas	13
	2	CHP	8
	1	Biomass (Wood)	8 (May 2019)

6.1.2 Formulate the problem

In this step of the decision aiding process, we aim a transforming the decision problem into a formalized version of it, using the vocabulary and concepts from MCDA.

The aim of this decision aiding use-case is to evaluate the ability of improving the efficiency of a network by implementing temperature cascades. In that view, a specific branch (Figure 9) of the network located in the East of Nantes Island has been selected since the return temperatures of most of its seven substations are usually over 60 °C: we can observe that the median, first and last quartile of return temperatures at Subs$_6$, Subs$_4$, Subs$_3$, Subs$_2$ and Subs$_1$ are between 62°C and 75 °C (as shown in the study of deliverable D2.10).

Figure 9: Topology of the network



These substations present high heat sources in their return pipe to be valorised, and particularly at $Subs_6$. We define the heat recovery potential as the difference between the power in the return pipe of a given substation at its functioning temperature and the power at a temperature of 55 °C. We can observe on figure 9 that for the $Subs_6$ the power varies from 80 kW to 200 kW in winter months, while the consumptions of $Subs_4$, $Subs_3$, $Subs_2$ and $Subs_1$ are always below 100 kW. It suggests that the high amount of heat available on the return pipe after $Subs_6$ could be used by the other substations by connecting them both to the supply and the return pipe.

For each scenario, the input data used in the simulation are demand powers in the substations and the outdoor temperature which is needed to calculate the secondary side supply temperatures using control curves. These data cover a period of 13 days in December/January 2018. The fraction of mass flow rate injected from the return pipe to the substation is optimized at each time step.

The 2 decision-makers identified beforehand agree on a set of 4 evaluation criteria of the scenarios (which represent energetic, economic and social perspectives of the problem):

- Criterion g_1: Energy savings [MWh]. It is calculated as the difference between the energy produced by the production unit for each scenario and a reference case where there is not any substation connected to the return pipe.
- Criterion g_2: Energy diagnostic of the substation consisting in the yearly energy consumption per heated surface. This criterion expresses the level of priority between connecting the substation to the return pipe

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

and to retrofit the buildings connected to the given substation. For this criteria the results are presented with letters (inspired from the Energy Performance Certificate labels)

- Criterion g_3: the connection cost. We do the assumption of a constant cost whatever the considered substation so that it is directly linked to the number of substations in the scenario.
- Criterion g_4: the potential acceptation by the owners. This criterion formalizes the level of difficulty to convince the owners to accept modification in the infrastructure. In that view, we consider a scale from "–" for strong resistance of the owners in the negotiations (e.g. multi-ownership buildings with many persons to convince), to "++" for low resistance (e.g. buildings owned by the municipality).

The evaluation of each scenario regarding the criteria are summed up in the Table 4:

Table 4: Performance table of the scenarios

Scenario	Connected substations	g1	g2	g3	g4
1	$Subs_4$	3.66	C	1	-
2	$Subs_4$, $Subs_2$, $Subs_1$	3.84	B	3	-
3	$Subs_4$, $Subs_3$	3.88	B	2	-
4	$Subs_2$	2.76	A	1	-
5	$Subs_1$	0.14	B	1	+
6	$Subs_3$	0.45	C	1	++

6.1.3 Construct the evaluation model

In this step of the decision aiding process, the goal is to select an MCDA algorithm, and to tune it according to the preferences of the decision-makers.

As the decision-maker wish to perform a qualitative (ordinal) evaluation of the scenarios, the class of possible MCDA algorithms is restricted to those who are able to solve the *sorting* problem (assignment of alternatives to pre-defined categories).

The data contained in the performance table clearly underlines the heterogeneity of the evaluation scales of the criteria. This speaks in favor of outranking methods (as mentioned in Section 4.2).

Among the possible outranking sorting methods, we choose Majority-Rule Sorting (MR-Sort) (Bouyssou & Marchant, An axiomatic approach to noncompensatory sorting methods in MCDM, I: The case of two categories, 2007) (Bouyssou & Marchant, An axiomatic approach to noncompensatory sorting methods in MCDM, II: More than two categories, 2007).

The method allows us to build an overall qualitative scale for the evaluation of the scenarios, while presenting a very readable and operational model. It handles very easily a mix of quantitative and qualitative evaluation scales on the criteria.

The preferences of the decision-makers which are used in MR-Sort are represented here through criteria weights, which give the relative importance of criteria, a majority threshold which indicates the weight of a coalition of criteria in order to be considered sufficient and category limits, which are used to delimit the various categories. In more complex versions, veto and dictator profiles are also considered, which allow to represent more precisely the preferential behavior of the DMs.

The basic version of MR-Sort is formalized as follows.

Consider a finite set *A* of decision alternatives (here the scenarios), a finite set of evaluation criteria indexed by J and a set of category limits $\mathrm{B} = \{B_1, .., B_{k-1}\}$. Each alternative and each category limit is a vector of evaluations with respect to all criteria. The evaluation with respect to criterion can be viewed as a function $g_j: A \cup B \to \mathbb{R}$, where $g_j(a)$ denotes the evaluation of alternative $a \in A$ on criterion j and $g_j(b_h)$ denotes the evaluation of category limit $b_h, \forall h \in \{1, .., k-1\}$, on criterion j .

The set of category limits are used to define a set of k categories $\{C_1, .., C_k\}$, ordered by their desirability, from C_1 being the worst category to C_k being the best one. Each category C_h is defined through its upper limit, b_h, and its lower limit, b_{h-1}, with the exception of the worst and best categories, which have only one limit. These categories represent the various levels of the qualitative scale on which the scenarios have to be evaluated.

To simplify the presentation, we assume, without loss of generality, that the performances are supposed to be such that a higher value denotes a better performance. This will obviously not be the case in the study presented later in this section. Furthermore the performances of the category limits are non-decreasing, i.e. $\forall j \in J, 1 < h < k: g_j(b_{h-1}) \leq g_j(b_h)$.

An alternative a is said to outrank a category limit b_{h-1} if and only if there is a sufficient coalition of criteria supporting the assertion "a is at least as good as b_{h-1}". A coalition of criteria corresponds to a subset of criteria which "agree" on how an alternative compare to a category limit, either being at least as good or strictly worse. To measure this, we define for each criterion j a function $C_j: A \times B \to \{0,1\}$ which assesses whether criterion j supports that statement or not:

$$\forall j \in J, a \in A, 1 \leq h \leq k: C_j(a, b_{h-1}) = \begin{cases} 1, if\ g_j(a) \geq g_j(b_{h-1}), \\ 0, \text{otherwise} \end{cases} \qquad (1)$$

To assess whether a coalition of criteria is in favor of the outranking or not, $\forall a \in A, 1 \leq h \leq k$, we first define the overall concordance as:

$$C(a, b_{h-1}) = \sum_{j \in J} w_j C_j(a, b_{h-1}) \qquad (2)$$

where w_j is the weight of criterion j. The weights are defined so that they are positive and sum up to one. This overall concordance is then compared to a majority threshold λ extracted from the decision-maker's preferences along with the weights. As in this basic version we do not consider any veto rule here, the outranking relation S is then defined as:

$$aSb_{h-1} \Leftrightarrow C(a, b_{h-1}) \geq \lambda \qquad (3)$$

If $C(a, b_{h-1}) < \lambda$, the coalition of criteria is not sufficient, the alternative does not outrank the frontier b_{h-1} and will therefore be assigned in a category lower than c_h.

Alternative a is then assigned to the highest category it outranks, hence this rule can be written as:

$$a \in c_h \Leftrightarrow \ aSb_{h-1} \text{ and } aSb_h. \qquad (4)$$

In order to model more accurately the preferential behavior of the DMs, several extensions of the MR-Sort model have been proposed in the literature. For example, some work extended this approach in order to handle large performance differences (resulting in what is called *dictator* and *veto* effects). These extensions lead to more flexible and more complex models as their number of parameters increases.

Several techniques have been proposed in the literature to determine the parameters of outranking-based multi-criteria sorting models. The decision-maker could provide them directly, which is what we propose to do for the district heating problem presented here. As an alternative, several authors have proposed to find the parameters of the model through the use of assignment examples.

Regarding this direct preference elicitation, our interviews with the decision-makers provide us with the following information.

The *user* decision-maker considers that a good value should be above 3MWh on criterion g_1, below 2 on criterion g_2, and below 3 on criterion g_3. As he is not taking into account criterion g_4 when he evaluates the various scenarios, these evaluations are not important to him. This defines the separation profile between the categories “good” and “bad” for him. Besides, he cannot make a decision based on only one of the remaining three criteria. But if a scenario is good on criterion g_2, then it is enough for him that it is good on either g_1 or g_3 to be considered on the overall as a good scenario. Finally, being good on g_1 and g_3 alone is not sufficient for him to evaluate the scenario as good. This defines the weights of the criteria, together with the majority threshold λ for this decision-maker.

The *operator*'s profile is a bit different. A good value on g_1 is above 2MWh, on g_2 below 2, on g_3 below 1, and on g_4 above 0. Again, this gives us the separation profile between the “good” and the “bad” classes. For him g_1 is clearly the most important criterion, and if a scenario is good on g1 and either g_2 or g_4, the scenario can be considered as good on the overall. If, however the scenario is not good on g_1, then it should be good on the remaining 3 criteria to be evaluated as good. And finally, if a scenario is below 0.4 on g_1, then it definitely cannot

This project has received funding from the European Union‘s Horizon 2020 research and innovation programme under grant agreement No 731297.

be considered as good on the overall, however good it has been evaluated on the remaining criteria. From these statements, we can deduce the weights of the criteria together with the majority threshold for this decision-maker.

All in all, these preferential information lead to the following preferential parameters of Table 5 and Table 6.

Table 5: Weights, veto and majority threshold

Decision-maker	g_1	g_2	g_3	g_4	λ
User	0.25	0.50	0.25	0	0.75
Operator	0.40	0.20	0.10	0.30	0.60
Operator veto thresholds	0.40	NA	NA	NA	

Table 6: Separation profiles

Decision-maker	g_1	g_2	g_3	g_4
User	3	2	3	0
Operator	2	2	2	0

For the user, the simple MR-Sort model described in this section is applied to evaluate the scenarios. For the operator however, as the notion of veto has to be taken into account (no good scenario should be below 0.4 on g_1). Therefore a classical extension of MR-Sort is used including this notion of veto

6.1.4 Reach a final recommendation.

In this fourth step of the decision aiding process, the goal is to reach a compromise recommendation for both decision-makers. Therefore the results of the separate assignments are first given in Table 7.

Table 7: Results of the assignments

Decision-maker / Scenario	1	2	3	4	5	6
User	"Bad"	"Good"	"Good"	"Good"	"Good"	"Bad"
Operator	"Bad"	"Good"	"Good"	"Good"	"Bad"	"Bad"

Consequently, a compromise decision would be to consider that scenarios 2, 3 and 4 are evaluated as "Good", as both decision makers agree.

The proposed decision aiding process should simplify their decision on which of these 3 scenarios should be implemented in practice, in order to improve the district heating of this part of the city of Nantes.

It is interesting to notice that the reasons why these scenarios have been evaluated as “good” by the 2 DMs may not be the same for both of them. For example, scenario 4 has been evaluated as “good” for the *user* decision-maker, because it is considered as good on g_2 and g_3, which is sufficient for this decision-maker. However, this

This project has received funding from the European Union‘s Horizon 2020 research and innovation programme under grant agreement No 731297.

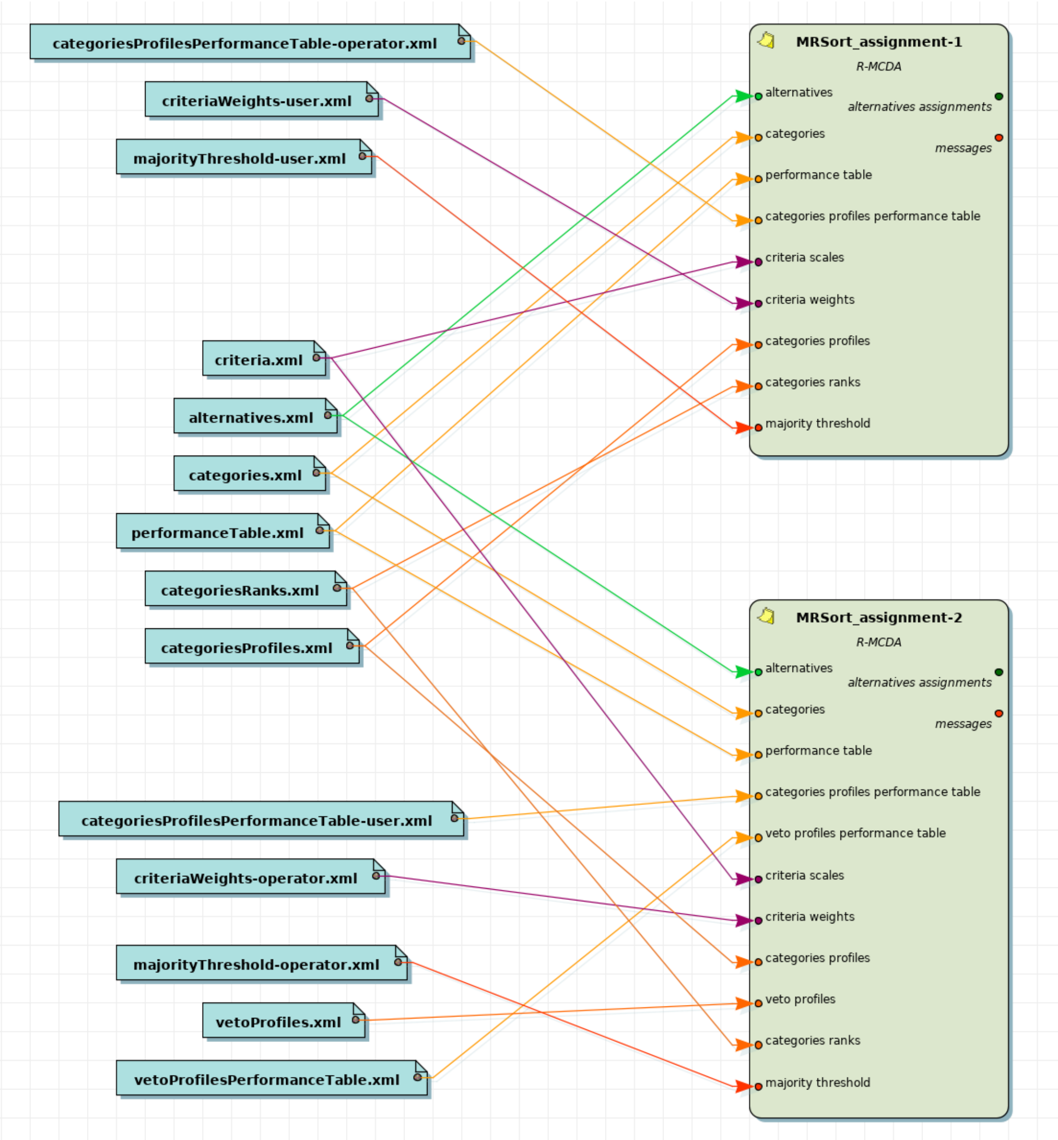
same scenario is evaluated as “good” for the *operator* decision-maker, because it is considered as good on the first 3 criteria, which again is sufficient for this decision-maker.

This clearly underlines the importance of modeling the preferences of the decision-maker in such a process accurately, as the results depend a lot on the preferential parameters.

6.2 The diviz and my.diviz decision aiding tools to support the decision aiding process

During the third step (construction of the evaluation model) of the previous decision aiding process, we have started designing the algorithmic workflow in diviz. This workflow is shown in Figure 10.

Figure 10: diviz workflow designed for solving the district heating use-case



This project has received funding from the European Union‘s Horizon 2020 research and innovation programme under grant agreement No 731297.

Two “MRSort_assignment” algorithms are inserted in this workflow (green rounded boxes on the right), one for each of the two decision-makers. On the left the data, in XMCDA format is shown. There are 3 groups of data files. In the middle, the data common to the two decision-makers (performance table of the six scenarios, labels and ranks of the categories, …). On the top and on the bottom two other groups of files can be seen. They correspond to the preference parameters of the *operator* decision-maker (top), and the preferences of the *user* decision-maker (bottom). One can also see that the bottom algorithm differs slightly from the top one, as it takes as input the veto thresholds.

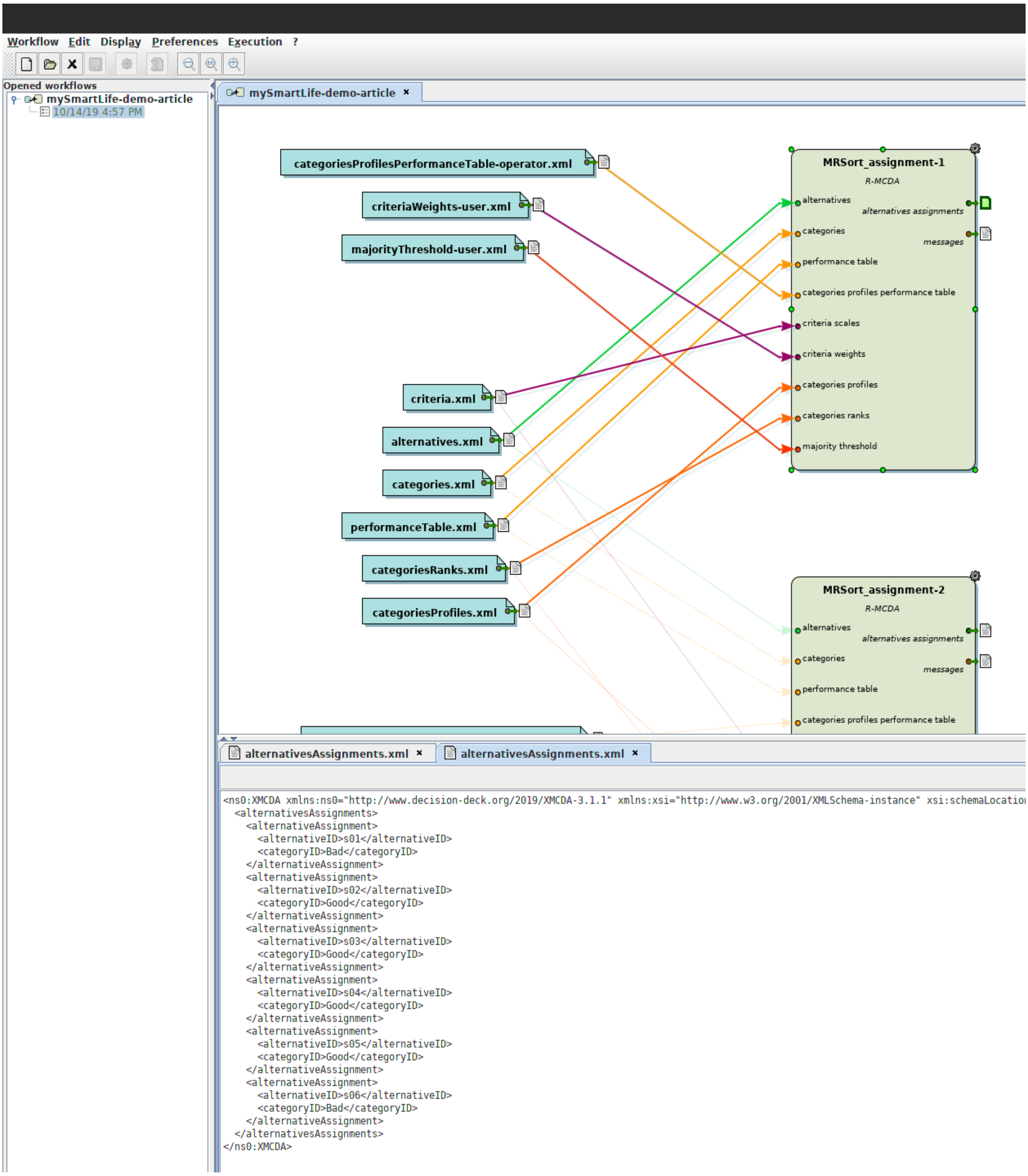
The output of both “MRSort_assignment” algorithms are the assignments of the 6 scenarios according to the two decision-makers’ preferences (as summarized in Table 7). This output is in XMCDA format, which means that it is not in a very human-readable compact form (see the output of the first algorithm in Figure 11).

To facilitate the interactions between both decision-makers and these algorithms, their input data, as well as the output assessments, we created a my.diviz project based on this workflow and these data. This project is structured around the following requirements (obtained from our discussions with them):

1. the decision-makers should be able to see both sets of recommendations,
2. the decision-makers wish to have a summarized view of the scenarios which allows to compare them easily
3. the decision-makers should be able to adjust the various input parameters of the decision models (the preferences)
4. the decision-makers wish to have a transparent view of the decision model in order to better understand how it works (to avoid a black box effect of the recommendations).

To answer requirements 1. and 2. we have designed a dashboard in my.diviz containing solely the decision recommendations of both decision-makers, as well as a radar-plot of the alternatives (which allows to compare their performances pairwisely). For requirement 3. we are providing both decision-makers in the input tab with editable input fields of the various preference parameters. To simplify this editing, we also add some filters to show only the preferences of each of the decision-makers separately. Finally, for requirement 4., in the output tab we present a compact version of MR-Sort models, where each scenario is compared to the separation profile. This allows to understand why a scenario is evaluated “good” or “bad”.

Figure 6 shows the dashboard generated for this project. In the top components, both decision-makers can see the assignments of the scenarios to the two classes “good” and “bad”. On the bottom, a radar plot of the scenarios is presented. It allows the decision-makers to focus on the various scenarios to understand why they have obtained a certain overall evaluation. In Figure 13 we show that one of the decision-makers has decided to compare scenarios 5 and 6 to understand why they have been assigned differently for the *user* decision-maker.



```
<ns0:XMCDA xmlns:ns0="http://www.decision-deck.org/2019/XMCDA-3.1.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocatio
  <alternativesAssignments>
    <alternativeAssignment>
      <alternativeID>s01</alternativeID>
      <categoryID>Bad</categoryID>
    </alternativeAssignment>
    <alternativeAssignment>
      <alternativeID>s02</alternativeID>
      <categoryID>Good</categoryID>
    </alternativeAssignment>
    <alternativeAssignment>
      <alternativeID>s03</alternativeID>
      <categoryID>Good</categoryID>
    </alternativeAssignment>
    <alternativeAssignment>
      <alternativeID>s04</alternativeID>
      <categoryID>Good</categoryID>
    </alternativeAssignment>
    <alternativeAssignment>
      <alternativeID>s05</alternativeID>
      <categoryID>Good</categoryID>
    </alternativeAssignment>
    <alternativeAssignment>
      <alternativeID>s06</alternativeID>
      <categoryID>Bad</categoryID>
    </alternativeAssignment>
  </alternativesAssignments>
</ns0:XMCDA>
```

Figure 11: Execution of diviz and XMCDA output

For this decision-maker criterion “Acceptance” is not important (weight equal to 0), which explains partly why this great advantage of scenario 6 over scenario 5 is not sufficient for scenario 6 to be “good”.

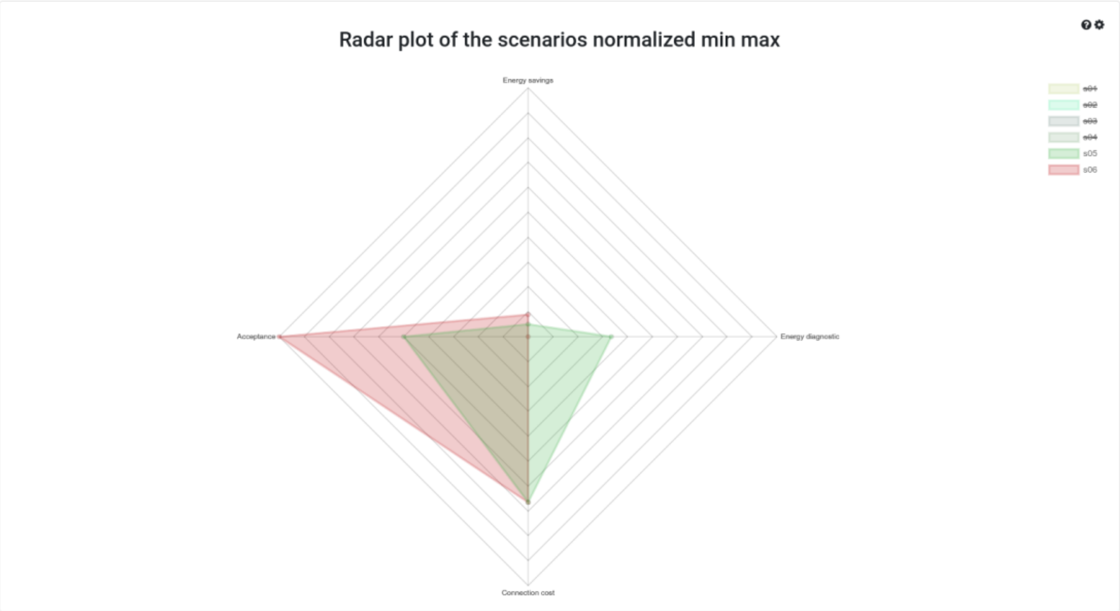
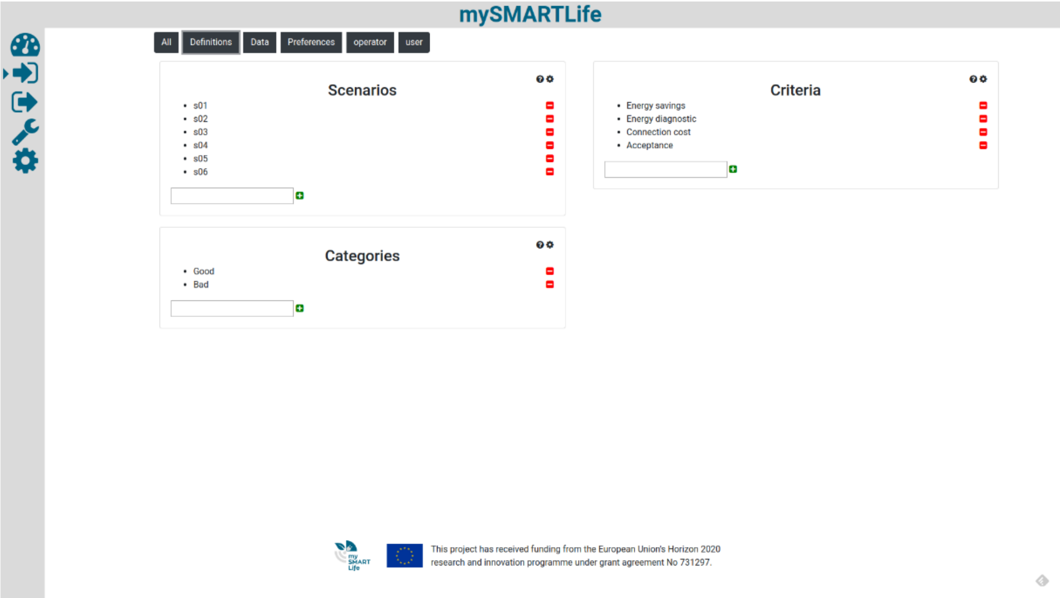


Figure 12: Radar plot of 2 scenarios

Let us now switch to the input screen of this project. It is composed of 3 sections which can be shown separately by clicking on the black "sections" buttons in the top of the screen, Figure 13, Figure 14 and Figure 15 show the definitions, data, and preferences sections separately. It is obviously possible to show them at once by clicking on "All", but we prefer to focus on separate views for this report.



Scenarios

- s01
- s02
- s03
- s04
- s05
- s06

Criteria

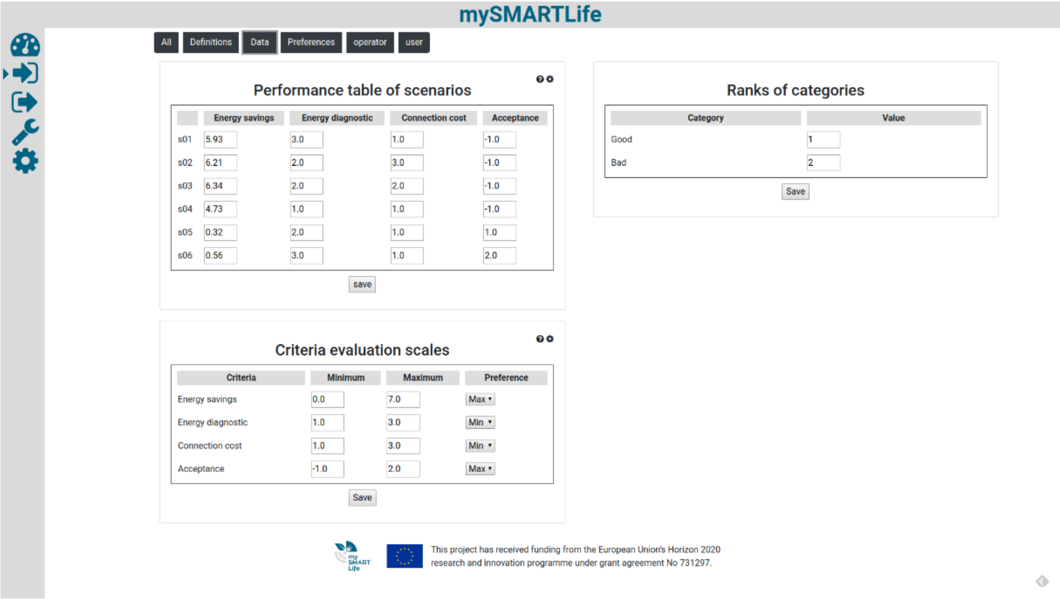
- Energy savings
- Energy diagnostic
- Connection cost
- Acceptance

Categories

- Good
- Bad

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

Figure 13: Definitions of alternatives, criteria and categories



Performance table of scenarios

	Energy savings	Energy diagnostic	Connection cost	Acceptance
s01	5.93	3.0	1.0	-1.0
s02	6.21	2.0	3.0	-1.0
s03	6.34	2.0	2.0	-1.0
s04	4.73	1.0	1.0	-1.0
s05	0.32	2.0	1.0	1.0
s06	0.56	3.0	1.0	2.0

Ranks of categories

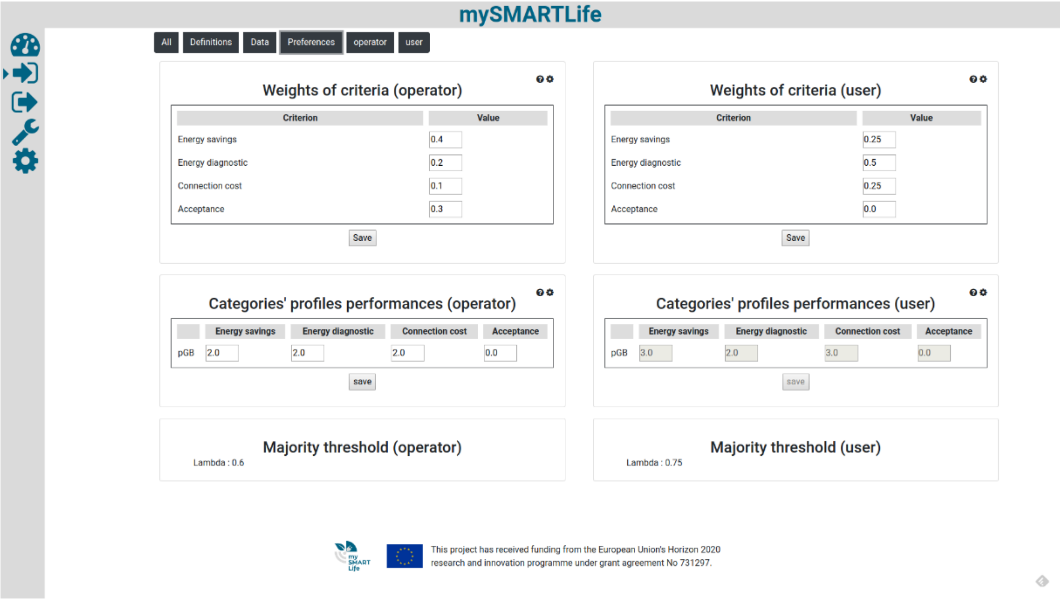
Category	Value
Good	1
Bad	2

Criteria evaluation scales

Criteria	Minimum	Maximum	Preference
Energy savings	0.0	7.0	Max
Energy diagnostic	1.0	3.0	Min
Connection cost	1.0	3.0	Min
Acceptance	-1.0	2.0	Max

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

Figure 14: Input data



Weights of criteria (operator)

Criterion	Value
Energy savings	0.4
Energy diagnostic	0.2
Connection cost	0.1
Acceptance	0.3

Weights of criteria (user)

Criterion	Value
Energy savings	0.25
Energy diagnostic	0.5
Connection cost	0.25
Acceptance	0.0

Categories' profiles performances (operator)

	Energy savings	Energy diagnostic	Connection cost	Acceptance
pGB	2.0	2.0	2.0	0.0

Categories' profiles performances (user)

	Energy savings	Energy diagnostic	Connection cost	Acceptance
pGB	3.0	2.0	3.0	0.0

Majority threshold (operator)

Lambda : 0.6

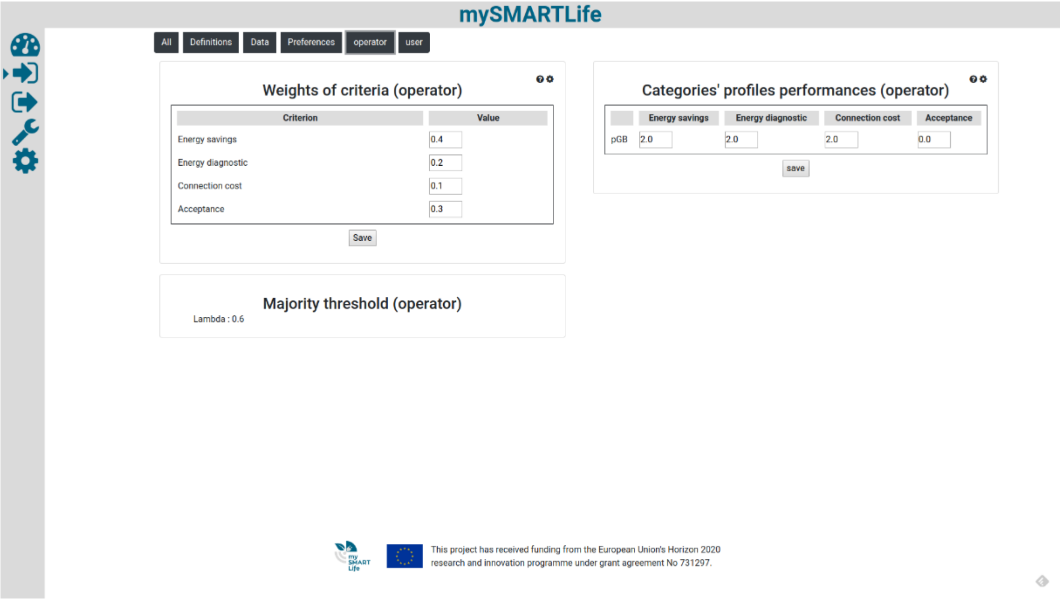
Majority threshold (user)

Lambda : 0.75

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

Figure 15: Preferences of the 2 decision-makers

To focus on either one of the decision-maker’s preferences, one can click on the name of the decision-maker. This generates the two screens of Figure 16 and Figure 17. These screens allow each decision-maker to update the weights of the criteria, the categories’ profile, or the majority threshold.



mySMARTLife

Weights of criteria (operator)

Criterion	Value
Energy savings	0.4
Energy diagnostic	0.2
Connection cost	0.1
Acceptance	0.3

Categories' profiles performances (operator)

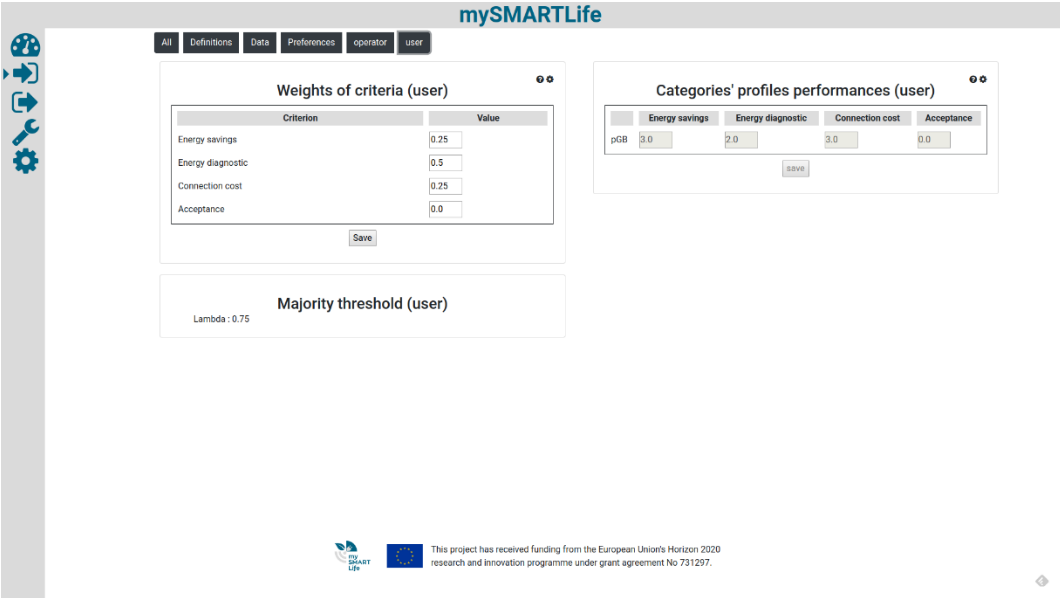
	Energy savings	Energy diagnostic	Connection cost	Acceptance
pGB	2.0	2.0	2.0	0.0

Majority threshold (operator)

Lambda : 0.6

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

Figure 16: Operator's preferences



mySMARTLife

Weights of criteria (user)

Criterion	Value
Energy savings	0.25
Energy diagnostic	0.5
Connection cost	0.25
Acceptance	0.0

Categories' profiles performances (user)

	Energy savings	Energy diagnostic	Connection cost	Acceptance
pGB	3.0	2.0	3.0	0.0

Majority threshold (user)

Lambda : 0.75

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

Figure 17: User's preferences

Regarding the output screen, we have defined 2 sections there. The first one (Figure 18) shows some plots summarizing the MR-Sort model and the assignments of the alternatives to their corresponding categories. The second one (Figure 19) shows the output of the MR-Sort model again, similarly to the component of the dashboard.

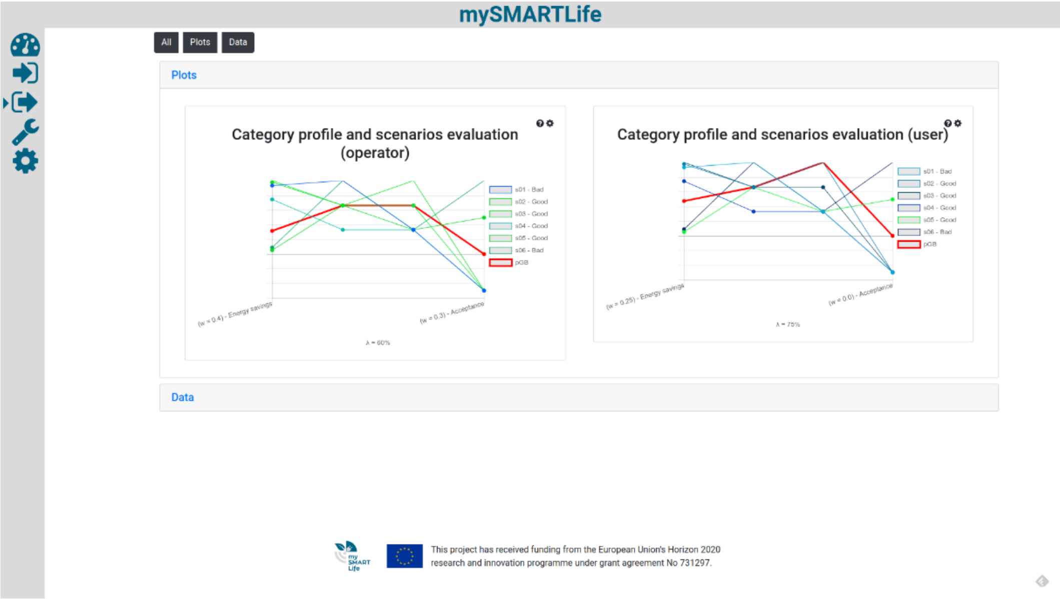
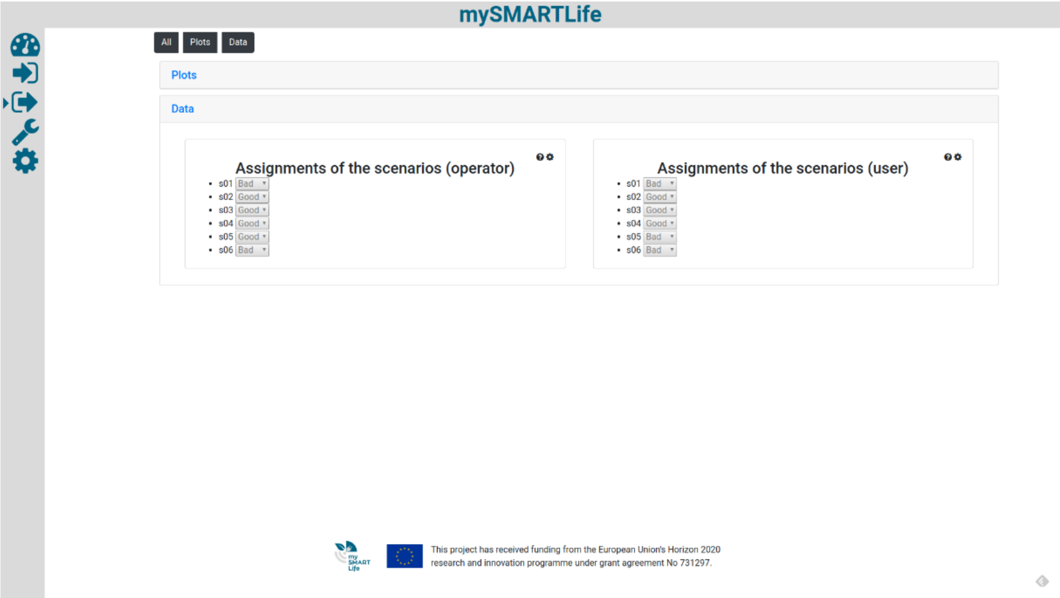

mySMARTLife
All
Plots
Data
Plots
Category profile and scenarios evaluation (operator)
s01 - Bad
s02 - Good
s03 - Good
s04 - Good
s05 - Good
s06 - Bad
pGB
(w = 0.4) - Energy savings
(w = 0.3) - Acceptance
λ = 60%
Category profile and scenarios evaluation (user)
s01 - Bad
s02 - Good
s03 - Good
s04 - Good
s05 - Good
s06 - Bad
pGB
(w = 0.25) - Energy savings
(w = 0.0) - Acceptance
λ = 75%
Data
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

Figure 18: Category profiles


mySMARTLife
All
Plots
Data
Plots
Data

Assignments of the scenarios (operator)

- s01 Bad
- s02 Good
- s03 Good
- s04 Good
- s05 Good
- s06 Bad

Assignments of the scenarios (user)

- s01 Bad
- s02 Good
- s03 Good
- s04 Good
- s05 Bad
- s06 Bad

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731297.

Figure 19: Assignments of the scenarios

If either one of the decision-makers wishes to modify some inputs (in the input screen), the preferences or the data, he / she then has to click on the “gear” icon to execute the underlying diviz workflow and to update the output correspondingly.

6.3 Discussion

This work presents the interest of coupling MCDA methods with modelling tools for improving the existing DH by connecting relevant substations to the return pipe.

The MCDA method combines the information given by energy indicators by taking into account various criterion associated to diverse actors who can have different points of view on the importance to be allocated to each indicator. The results show the coupled methods enable to identify strategies not obvious at a first glance and provide useful information for discussion between the different stakeholders. The proposed methodology in the example of a single branch is replicable to the whole DH as well as the list of criteria and points of view can be broaden depending of the case studied and the actors involved.

We have also shown how easily my.diviz can be configured to create a project which is specific to

- the requirements of the 2 decision-makers,
- the underlying diviz workflow, designed by the analyst.

It should now be clear that my.diviz can easily be tuned to many diviz workflow and can therefore be used as an effective decision aiding tool in a smart cities context, containing a variety of different decision problems, decision-makers, and requiring therefore various resolution algorithms to support the decision aiding process.

In terms of TRL (Technology readiness level), this application to the DH problem in Nantes Métropole, demonstrate the relevancy of the decision aiding process and the decision support tools. Consequently we consider that this deliverable has a TRL of 7 (System prototype demonstration in operational environment), tending towards a level 8 (System complete and qualified). We estimate that by the end of the project, through the monitoring process proposed below, we will achieve a TRL of 9.

7. Monitoring

The monitoring process is an essential part of the project. It provides the necessary raw material for carrying a precise evaluation of this project, keeping track of performance, further helping to understand the impacts one intervention can bring.

As part of WP5 dedicated to the monitoring, some work has already been done and preparation was undertaken to be ready for M36, and to prepare for the monitoring.

The WP5 work is still ongoing, and projections are subjected to change; but first milestones have undoubtedly been reached upon the common work of WP5 T5.1, T5.2 and T5.3, such as the set of monitoring KPIs, shared between the three lighthouse cities and based upon reference sources ("SCIS" or "CITYKEYS" to name a few).

As for this deliverable, no physical action is meant to be implemented in the frame of the mySMARTLife project. Its purpose is rather to propose a general decision aiding process, which can be deployed in multiple application domains of smart cities, to help decision makers in complex decision making tasks. This process is illustrated on such a real-world application, namely the district heating optimization problem of Nantes Metropole (in collaboration with deliverable D2.10 (District Heating Monitoring and Optimization), and supported by the my.diviz tool.

In consequence, due to the particularity if the action, the monitoring proposed for the two next years will address:

- The objective of the mySMARTLife program is to make the actions operational. So, an evaluation of the usefulness of the present action applied to the DH optimization will be done. This evaluation will cover both the use of the action on the DH system and its coupling with the present action.
- Depending on the previous assessment, the my.diviz and diviz tools could be deployed for the user and the operator. It would in that case be interesting to evaluate the number of uses of the tool, and make a small study on the acceptancy of the tool by the end users.
- A second case study has also been discussed during the first 36 months of the project. Currently we could not converge on such a complementary illustration of the decision aiding process, but depending on the outcome of the future discussions, another such case study could be tackled by the current deliverable.
- Such a second case study, as well as a deployment for real decision problems would probably bring this deliverable to a TRL of 9.

8. Conclusions

We have shown in this deliverable that a formal state of the art decision aiding process can be applied to many decision problems linked to the migration of a traditional city to a smart city, or simply the daily management of a smart city. The specificities of these decision problems are linked to the necessity to take into account several conflicting points of view, simultaneously with the perspective of human stakeholders.

The software tools which are used to support this decision aiding process are twofold: on the one hand, we propose to use diviz, a workbench for the analyst, which helps designing complex decision aiding algorithms, tuned to the characteristics of the underlying decision problem. On the other hand, we have developed my.diviz, a human-friendly layer above diviz, which allows decision-makers to interact with decision aiding algorithms, and to edit the data of their decision problems.

The citizen has a central role in a smart city. It is therefore important that social factors are taken into account in decision problems, next to more technical indicators. Furthermore, another important goal of smart cities is to reduce their carbon footprint. Therefore, it is also important to take into account the ecological impacts of different choices in decision-making. However, these impacts are often in conflict with economic or even welfare criteria.

By taking into account these two major concerns in the decision problems that this deliverable is able to address, we think that we have contributed to a major concern of mySMARTLife.

Next to that, by maintaining a high level of genericity, both in the decision aiding process proposal, as well as the 2 software tools, we are contributing to a good level of replicability of this task. The proposed method and tools are independent of any practical application, and can therefore be deployed for many other decision problems linked to smart cities, as it has been successfully shown with the optimization of the district heating network in the city of Nantes.

9. References

Arrow, K. J. (1951). *Social choice and individual values.* New York: J. Wiley.

Bisdorff, R., Meyer, P., & Roubens, M. (2008). Rubis: a bipolar-valued outranking method for the best choice decision problem. *4OR, Quaterly Journal of the Belgian, French and Italian Operations Research Societies, 6*, pp. 143-165.

Bouyssou, D., & Marchant, T. (2007, April). An axiomatic approach to noncompensatory sorting methods in MCDM, I: The case of two categories. *European Journal of Operational Research, 178*(1), pp. 217-245.

Bouyssou, D., & Marchant, T. (2007). An axiomatic approach to noncompensatory sorting methods in MCDM, II: More than two categories. *European Journal of Operational Research, 178*(1), pp. 246-276.

Bouyssou, D., Marchant, T., Pirlot, M., Tsoukias, A., & Vincke, P. (2006). *Evaluation and decision models with multiple criteria, Stepping stones for the analyst.* New York: Springer.

Brans, J. P., & Mareschal, B. (2002). *PROMETHEE-GAIA. Une Méthodologie d'Aide à la Décision en Présence de Critères Multiples.* Paris, France: Ellipses.

Fishburn, P. C. (1970). *Utility Theory for Decision Making.* Wiley, New York.

Grabisch, M. (1997). Alternative representations of discrete fuzzy measures for decision making. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 5*(5), pp. 587-607.

Keeney, R. L., & Raiffa, H. (1976). *Decision with multiple objectives.* New-York: Wiley.

Luce, R. D., & Raiffa, H. (1957). *Games and Decisions.* J. Wiley, New York.

Mousseau, V. (2003). Elicitation des préférences pour l'aide multicritère à la décision. Mémoire présenté en vue de l'obtention de l'habilitation à diriger des recherches, Université Paris-Dauphine.

Roy, B. (1968). Classement et choix en présence de points de vue multiples (la méthode ELECTRE). *Revue française d'informatique et de recherche operationelle (RIRO), 2*, pp. 57-75.

Roy, B. (1985). *Méthodologie multicritère d'aide à la décision.* Ed. Economica, collection Gestion.

Roy, B., & Bouyssou, D. (1993). *Aide multicritère à la décision: Méthodes et cas.* Economica, Paris.

Roy, B., & Vanderpooten, D. (1996). The European school of MCDA: Emergence, basic features and current works. *Journal of Multi-Criteria Decision Analysis, 5*, pp. 22-38.

Samuelson, P. A. (1938). A note on the pure theory of consumer's behavior. *Economica, 5*(17), pp. 61-71.

Simon, H. A. (1957). A behavioural model of Rational Choice. In H. A. Simon (Ed.). J. Wiley, New York.

Vincke, P. (1989). *L'aide multicritère à la décision.* Paris: Ellipses.

von Neumann, J., & Morgenstern, O. (1944). *Theory of games and economic behavior.* Princeton University Press, Princeton.

von Winterfeldt, D., & Edwards, W. (1986). *Decision Analysis and Behavorial Research.* Cambridge: Cambridge University Press.

Weistroffer, H., Smith, C., & Narula, S. (2005). Multiple Criteria Decision Support Software. (S. G. J. Figueira, Ed.) *Multiple Criteria Decision Analysis: State of the Art Surveys*, 989-1018.