Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften
Department für Informatik

Decision Evaluation System – Towards Sustainable Decision-Making

Dissertation

Submitted in fulfilment of the requirements for the degree:
Doctor of Engineering (Dr.-Ing.)

Submitted by:

Dipl.-Wirt.-Inf. (FH) Abdelkerim Rezgui

Date: May 29, 2018

"If I were given one hour to solve the planet I will spend 59 minutes understanding the problem and one minute solving."

Albert Einstein

„It is beyond a doubt that all our knowledge begins with experience."

Immanuel Kant

"Each problem that I solved became a pattern which served afterwards to solve other problems."

René Descartes

Statement

I hereby declare that I completed this dissertation independently and used only the indicated ressources.

This dissertation has, neither as a whole, nor in part, been submitted for assessment in a doctoral procedure at another university.

I confirm that I am aware of the guidelines of good scientific practice of the Carl von Ossietzky University Oldenburg and that I observed them.

I confirm that I have not availed myself of any commercial placement or consulting services in connection with my promotion procedure.

Abstract

The industrial revolution, electronics, medicines and the internet are affecting our planet, causing pollution (air, soil and water), resource depletion and climate change. Governments and non-government organizations are pushing stakeholders to devote increasing attention to the environment by establishing rules and guidelines. In order to meet those rules and guidelines, decision-makers are using specific computer-based systems. Decision support systems (DSS) and environmental management information systems (EMIS) play a major role here in delivering information about processes and operations of organizations.

However, the role of these systems is limited to initiating the decision-making act without any support while making decisions. They neglect the archiving, tracking, recommendation and evaluation of decisions based on their sustainability impact (Rezgui and Marx Gómez, 2017).

A decision support system (DSS) is a set of tools, techniques and methodologies that supports and improves the decision-making process. It includes the use of available data, documents, models and knowledge by a computerized system to process the raw data and turn it into useful knowledge within the appropriate context. Many research papers written about DSS and EMIS were explored to recognize the status and the limitations in the field. The literature review process was based on the systematic literature review approach, showing that the aim of the early DSS in the 1960s was to make the transactional data (billing, payroll, inventory, etc.) available to the managers for decision-making purposes (Arnott and Pervan, 2005a). Subsequently, (Gorry and Morton, 1989) extended this aim to support managerial decision-making that is structured or unstructured. Fakeeh considered that the existing DSS can be categorized into seven families: data-driven DSS, communication-driven DSS, group DSS, document-driven DSS, model-driven DSS, knowledge-driven DSS and web-based DSS (Fakeeh, 2015). These families were mapped with the main expected capabilities from a DSS. In order to evaluate the actual situation in the domain of decision support systems, the recommendation of Phillips-Wren et al. (2009) in their work on intelligent DSS was used. They annotate that the evaluation of a DSS should be made among others based on the understanding and use of Simon's phases in the decision-making process (Herbert A Simon, 1960). None of the actuall DSS cover all outcomes and especially the ability to deliver information about the decision itself (decision-maker information, decision date, decision objectives, goals and targets, deadline to achieve the expected goals, etc.) in addition to the evaluation based on the sustainability impact. According to the Simon's decision-making process, the DSS should enable decision-makers to learn from their past choices. It is important to obtain information about the decision itself through storing, tracking, recommendation and evaluating decisions to enable this learning.

In this dissertation, the aim is to enhance the quality of decisions in terms of sustainability using a new concept called Decisions Evaluation System (DES). The design and implementation of a

decision evaluation system based on sustainability is planned. The three evaluations pillars : ecological, economic and social should be calculated for each decision. This system should enable stakeholders to track, evaluate, recommend and comment on decisions. Moreover, it should comply with the understanding and use of Simon's phases in the decision-making process (Herbert A Simon, 1960). This novel system should improve sustainable decision-making within organizations.

Zusammenfassung

Die industrielle Revolution, Elektronik, Medizin und das Internet beeinflussen unseren Planeten und tragen einen wesentlichen Teil zur Umweltverschmutzung (Luft, Boden und Wasser), zum Ressourcenverknappung und Klimawandel bei. Weltweit encouragieren Regierungen und Organisationen alle Aktoren, der Umwelt zunehmende Aufmerksamkeit zu schenken, indem sie Regeln und Richtlinien aufstellen. Um diese zu erfüllen, verwenden Entscheidungsträger spezifische computergestützte Systeme. Entscheidungsunterstützungssysteme (DSS) und Betriebliche Umweltinformationssysteme (EMIS) spielen hier eine zentrale Rolle bei der Bereitstellung von Informationen über Prozesse und Abläufe von Organisationen.

Die Hauptrolle dieser Systeme beschränkt sich jedoch darauf, den Entscheidungsakt vorzubereiten ohne jegliche Unterstützung während und nach der Entscheidung. Sie vernachlässigen die Archivierung, Verfolgung, Empfehlung und Bewertung von Entscheidungen basierend auf ihrer Auswirkungen auf die Nachhaltigkeit (Rezgui und Marx Gómez, 2017).

Ein Entscheidungsunterstützungssystem beinhaltet eine Sammlung von Tools, Techniken und Methoden, die den Entscheidungs- und Planungsprozess unterstützen und verbessern. Es umfasst die Verwendung verfügbaren Daten, Dokumente, Modelle und Kenntnisse durch ein computergestütztes System, um die Rohdaten zu verarbeiten und sie in nützlichem Wissen im geeigneten Kontext zu verarbeiten. Viele Forschungsarbeiten über DSS und EMIS wurden untersucht, um den Status und die Einschränkungen in diesem Bereich zu erkennen. Das Literatur-Review wurde basierend auf dem systematischen Literatur-Review Prozess durchgeführt. Es zeigte, dass das Ziel des frühen eines DSS in den 1960er Jahren darin bestand, die Transaktionsdaten (Abrechnung, Gehaltsabrechnung, Inventur usw.) den Managern zur Entscheidungsfindung zur Verfügung zu stellen (Arnott und Pervan, 2005a). In der Folge erweiterten Gorry und Morton (1989) dieses Ziel, um strukturierte und/oder unstrukturierte Managemententscheidungen zu unterstützen. Fakeeh war der Meinung, dass bestehende DSS in sieben Familien / Gruppen eingeteilt werden können: datengetriebenes DSS, kommunikationsgesteuertes DSS, Gruppe DSS, dokumentengetriebenes DSS, modellgetriebenes DSS, wissensgesteuertes DSS und webbasiertes DSS (Fakeeh, 2015). Diese Familien /Gruppen wurden mit den wichtigsten erwarteten Fähigkeiten eines DSS kartiert. Um die aktuelle Situation im Bereich der Entscheidungsunterstützungssysteme zu bewerten, wurde die Empfehlung von Phillips-Wren et al. (2009) in ihrer Arbeit an intelligenten DSS verwendet. Sie kommentieren, dass die Bewertung eines DSS unter anderem auf dem Verständnis und der Verwendung der Simon-Phasen im Entscheidungsprozess basieren sollte (Herbert A. Simon, 1960). Keines der aktuellen DSS deckt alle Phasen ab. Es mangelt ihnen an der Fähigkeit, Informationen über die Entscheidung selbst (Entscheidungsträgerinformationen, Datum, Ziele, Zielstellungen und Zielvorgaben, Frist zur Erreichung der erwarteten Ziele usw.) zu liefern und ignorieren die Bewertung im Hinblick auf die

Auswirkungen auf die Nachhaltigkeit. Nach dem Entscheidungsprozess von Simon sollte ein DSS den Entscheidungsträgern ermöglichen, aus ihren Entscheidungen in der Vergangenheit zu lernen. Es ist wichtig, Informationen über die Entscheidung selbst zu erhalten, indem Entscheidungen gespeichert, nachverfolgt, empfohlen und bewertet werden, um dieses Lernen zu ermöglichen.

Ziel dieser Dissertation ist es, die Entscheidungsqualität (Operationale und teilweise Taktische Entscheidungen) in Bezug auf die Nachhaltigkeit mit einem neuen Konzept, dem Entscheidungsbewertungssystem (DES), zu verbessern. Die Konzeption und Implementierung eines auf Nachhaltigkeit ausgerichteten Entscheidungssystems ist geplant. Die drei Bewertungssäulen, ökologisch - ökonomisch - sozial, sollten für jeden Entscheidungsprozess einkalkuliert werden. Stakeholdern soll dieses System ermöglichen, Entscheidungen zu verfolgen, zu bewerten, zu empfehlen und zu kommentieren. Es fußt auf dem Verständnis und der Verwendung von Simons Phasen im Entscheidungsprozess (Herbert A. Simon, 1960). Dieses neuartige System wird zu einer entscheidenden Verbesserung der nachhaltigen Entscheidungsfindung in Organisationen beitragen.

Table of Contents

List of Abbreviations and Acronyms

ABI	Adaptive Business Intelligence
ADR	Action Design Research
AI	Artificial Intelligence
BI	Business Intelligence
BPI	Business Process Intelligence
BPM	Business Process Management
BPMN	Business Process Model And Notation
BSC	Balanced Scorecards
CBD	Case-Based Reasoning
CDSS	Clinical Decision Support Systems
CO^1	Carbon Monoxide
CO^2	Carbon Dioxide
DAO	Data Access Object
DES	Decision Evaluation System
DSISR	Design Science In Information Systems Research
DSR	Design Science Research
DSRM	Design Science Research Methodology
DSS	Decision Support Systems
DTM	Data Mart
DWH	Data Warehouse
EDSS	Environmental Decision Support Systems
EIS	Executive Information Systems
ELPI	Ecological Performance Indicators
EMIS	Environmental Management Information Systems
ENPI	Economical Performance Indicators
EPI	Environmental Performance Indicator
ERP	Enterprise Resource Planning
ETL	Extract-Transform-Load
EVIS	Environmental Information Systems
FESLM	Framework Evaluating Sustainable Land Management
FR	Functional Requirement
GDMP	Green Decision-Making Process
GSS	Group Support Systems
GUI	Graphical User Interface
HOLAP	Hybrid Online Analytical Processing

ICT	Information And Communications Technologies
IDSS	Intelligent Decision Support Systems
IS	Information Science
ISRF	Information System Research Framework
KM	Knowledge Management
KMDSS	Knowledge Management-Based Decision Support Systems
KPI	Key Performance Indicator
MDA	Multi-Dimensional Analytical
MIS	Management Information Systems
MOLAP	Multidimensional Online Analytical Processing
NO	Nitrogen Oxide
NSS	Negotiation Support Systems
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
OpBI	Operational Business Intelligence
PDSS	Personal Decision Support Systems
PM	Process Mining
POC	Proof Of Concept
PWH	Process Warehouse
QMS	Quality Management System
ROLAP	Relational Online Analytical Processing
SD	Sustainability Development
SO^2	Sulfur Dioxide
SPI	Social Performance Indicators
VOC	Volatile Organic Compound

1 Introduction

1.1 Motivation

The current situation in any organization is the result of the taken decisions in the past, whereas the future situation will be based on the decisions taken today. Managers are facing the aforementioned two facts, whereby their main activity is making decisions. Despite the fact that it can be argued that management can be considered as synonymous for decision-making, half of the decisions made by managers within organizations ultimately fail (Ireland and Miller, 2004). Decision-making which can be defined as choosing among two or more alternative courses of action for the purpose of attaining one or more goals (Turban et al., 2011a), makes managers in uncertainty. Then taking the right decision is typically not a simple matter, as most decision problems are highly complex in nature (Grünig and Kühn, 2005) and they may intensely influence (directly and/or indirectly) economical, ecological and social organization´s objectives.

For this reason, before making choices it is very important to understand the problem and envisage the consequences. Albert Einstein uttered "if I were given one hour to solve the planet I will spend 59 minutes understanding the problem and one minute solving". For a better understanding, a huge amount of data should be collected, synchronized, aggregated, harmonized and presented in a user-friendly form. This enables decision-makers to build a "first aid" pool of information and gather more knowledge to understand the situation.

However, the production and consumption of most important appliances automobiles, computers, electricity, food, mobile telephones, and synthetic materials are made today without sufficient understanding of their full life-cycle effects or recognition of their full social costs. This induces that massive quantity of usable materials go unrecovered and unused because current policies (such as water subsidies) encourage over-consumption or make materials recovery or resource efficiencies uneconomical for many products (National Research Council, 2014).

Having to clear and enough information about the situation oblige individuals to consider a range of more and less sustainable options. Assuring a balance between government rules, organizations objectives and protecting the environment and natural resources makes managers' activities challenging. Unfortunately they are hampered with different biases, which can push them towards choices that are less sustainable. Managers face many decisions, but they often make unsustainable ones (Arvai et al., 2012).

Despite the importance of understanding the details of the problem situation, it is insufficient. Two other important elements are necessary to derive a complete view: the cause and effect relationship between the taken actions in the past and the current situation; while the responsibility of those actions should be available. Learning from historical decisions through measuring and presenting their impact is the most significant knowledge piece in understanding the decision issue.

Then once the information has been validated, it must be made available to those who need it in the organisation. In order to enable the access to the necessary and useful information, information must be available, shared, understood and easily accessible (Ministère de l'Économie et des Finances, 2014). Information availability also means that information is filtered in such a way that acute company information arrives quickly at the recipient despite the general information flow. Giving specific context to the raw material (the information) is necessary for the generation of knowledge. The capacity for action of organizations requires the availability of the necessary information and knowledge at the right time at the right place (North, 2016). Sharing knowledge involves active interaction between several actors who possess the necessary knowledge and it is not limited to single actor efforts (Kwahk and Park, 2016). It needs fundamental prerequisites: the ability that people collaborate with each other and share the accumulated knowledge in their minds. In addition people should be motivated to measure and communicate their decisions including sustainability evaluation. Sharing knowledge and collaboration can be seen as a pillar of Enterprise 2.0 because it is at the heart of the processes implemented by organizations for more efficiency and effectiveness.

In this context, the need for having such solution that supports the management, evaluation and sharing of decisions sustainability impact in an industrial context was detected successively by the companies Intercolor and Zollner. In order to respond to this need a systematic literature review was conducted. In comparison to literature on decision support and environmental systems, there is less available academic literature on the management of decisions and evaluating of the impact based on sustainability on an organization' environment.

These motivated the investigation of solutions to support managers while making decisions. This support should not be limited in the preparation step, like classical DSS, but also in the making and evaluation step. Subsequently, inadequate decisions taken in the past might repeatedly occur over time in the future. Such a solution should bring companies one step closer in solving the problem of decision evaluation, enhancing the chance of deriving benefits from their historical pitfalls and sharing knowledge. It should encourage more sustainable choices within organizations and provide transparency by allowing the tracking of decisions and by providing their sustainability impact to the decisions-makers and/or stakeholders.

1.2 Related Work

DSS are a set of tools, techniques and methodologies that support and improve the managerial decision-making process. It includes the use of available data, documents, models and knowledge by a computerized system to process the raw data and turn it into a useful knowledge within the appropriate context (Rezgui et al., 2017). We explored many research works written about DSS to ascertain the status and limitations in the field. We based our literature review process on a systematic literature review approach, showing that the aim of the early DSS in the 1960s was to make the transactional data (billing, payroll, inventory, etc.) available to the managers for decision-making purposes (Arnott and Pervan, 2005a). Subsequently, (Gorry and Morton, 1989) extended this aim to support managerial decision-making that is structured or unstructured. Fakeeh considered that we can categorize the existing DSS into families: data-driven DSS, communication-driven DSS, group DSS, document-driven DSS, model-driven DSS, knowledge-driven DSS and web-based DSS (Fakeeh, 2015). In this chapter, more details about these types, their application domains and limitation will be given.

Based on the classification by (Power et al., 2015), DSS can be divided into six types based on what are they driven by:

Data-driven DSS: A DSS of this type provides access to and manipulation of a time-series of internal company data and sometimes external and real-time data. Simple file systems accessed by query and retrieval tools provide the most elementary level of functionality, e.g. data warehousing, OLAP, data mining instruments (Fakeeh, 2015).

Document-driven DSS: This type of DSS uses computational storage and processing abilities to provide document retrieval and analysis. The documents could be in various forms like audio, images, video, texts, etc. Examples of such documents that a decision-maker might need are meetings videos, texts of policies and procedures, product specifications or images of catalogues. Fedorowicz utters that a search engine represents a powerful decision-aiding tool associated with a document-driven DSS (Fedorowicz, 1993).

Knowledge-driven DSS: DSS under this category are considered “smarter” as they can suggest or recommend actions to managers. It is the artificial intelligence (AI) contribution to the field of DSS. These systems have knowledge codified to provide specialized problem-solving assistance. These DSS are person-computer systems with specialized problem-solving expertise. The "expertise" comprises knowledge about a particular domain, the understanding of problems within that domain and the "skill" in solving some of these problems (e.g. intelligent DSS).

Model-driven DSS: A model-driven DSS emphasizes access to and manipulation of financial, optimization and/or simulation models (a quantitative model). Model-driven DSS use data and parameters provided by decision-makers to aid them in analysing a situation.

Communications-driven DSS: These are DSS in which we find the use of network and communications technologies to facilitate and enhance collaboration, information sharing, etc. This is vital for the collaborative decision-making process, e.g. GSS, CSCW, groupware.

Web-based DSS: This type of DSS appeared in 1995 and comprises passing on decision support-related information and/or instruments to a decision-maker after having the latter utilizing a "thin client" web program (Navigator), TCP/IP protocol, etc. Web-based DSS can be communications-driven, data-driven, document-driven, knowledge-driven, model-driven or a hybrid. This is why it is not considered as a “type” by several researchers and I could not find it in many classifications; rather, it is only considered as a delivery model by some.

(Holsapple and Whinston, 2001) proposed a slightly different classification identifying text-oriented DSS, database-oriented DSS, spreadsheet-oriented DSS, solver-oriented DSS, rule-oriented DSS and compound DSS. Other classifications are also present in the literature, regardless on which technologies they are based upon or driven by. They are classified by use and appearance (group DSS, data warehouse, negotiation DSS, etc.), whereby we can distinguish between the following types:

Personal Decision Support Systems (PDSS): (referred to today as “Analytics”) are normally developed for one manager or a small number of independent managers for one decision task. PDSS are the oldest form of DSS and it was the only form of DSS for a decade (Arnott and Pervan, 2008). Fortunately, PDSS proved more successful than traditional management information systems (MIS), which gave a head start for the other types of DSS to gain interest in the professional environment and among academics for further development.

Group Support Systems (GSS): GSS comprises a set of software, hardware and language components and procedures that support a group of people (decision-makers) engaged in a decision-related meeting. This means that the responsibility of decisions is shared among multiple decision-makers rather than one in the case of PDSS. The term GSS can be expanded to include communication and information processing and includes the sub-fields of groupware, EMS, CSCW and CMS. Most available GSS software is flexible and with general purpose, whereby a typical GSS would include generating and reviewing corporate strategic plans, negotiations, conducting focus group studies, event planning and unveiling new methods for cost reduction (Lewis, 2010).

Negotiation Support Systems (NSS): also operates in a group context, but – as the name suggests – they involve the application of computer technologies to facilitate negotiations between opposing interests. NSS are more complex systems far from simply providing and ordering information or facilitating communication or coordination; moreover, there exist the issues of negotiations, participants, opposing and similar objectives, negotiation strategies and rules (Hosack et al., 2012). These NSS are thus a mixture of technical and psychological characteristics. These issues and complexities have distinguished the NSS from the GSS branch and developed it as a new, interesting type of DSS that gained the interest of researchers from the 1990s and remains an interesting field even in the 2010s (Hosack et al., 2012).

Intelligent Decision Support Systems (IDSS): as one of the results of applying AI to operate in DSS, intelligent DSS can be classed into two generations: the first involves the use of rule-based expert systems; and the second generation uses neural networks, genetic algorithms and fuzzy logic (Turban et al., 2005). Unfortunately, this sub-field of DSS – which first appeared in the 1980s – was not meant to survive as there exists a fundamental tension between AI and DSS: AI has the ultimate objective of replacing humans in many aspects, while this is not the objective of DSS, which is rather supporting human decision-makers. As a result, the greatest impact of AI techniques in DSS has been embedded in the PDSS, GSS or EIS. However, there are still some research works in recent years attempting to revive this DSS type. For instance, (Ma et al., 2014) have developed an IDSS for residential energy consumption, while (Zhou et al., 2015) used a big data-based IDSS for sustainable regional development.

Knowledge Management-based Decision Support Systems (KMDSS): KM-based decision support technologies can aid knowledge storage, retrieval, transfer and application by supporting individual and organizational memory and inter-group knowledge access (Arnott and Pervan, 2005a). It is simply using what is deemed as "knowledge" for the organizations to ameliorate decisions. KMDSS and DW are trending at present and (Hosack et al., 2012) predicted that the two DSS types will merge in the future and the focus will incorporate better ways to allow organizational members to interact with available information, wherever and whenever it is available.

Executive Information Systems (EIS): referred to today as "business intelligence", executive information systems are data-oriented DSS that provide reporting about the nature of an organization to management (Arnott and Pervan, 2008). However, today many researchers argue the term "business intelligence" has replaced by itself the DSS and EIS. BI is poorly defined by organizations to make it cope with their needs but many definitions are somewhat similar as they define BI as "a category of applications and technologies to manage (gather, analyse, access, report, integrate…) data to help enterprise users make better decisions" (Turban et al., 2007). Business intelligence (or EIS) has known a growing interest in the professional environment in recent years and practical use despite the non-relevance of the research works in the period of 1990-2003, although that has changed in the last decade, as argued in (Hosack et al., 2012), claiming that all DSS research has recovered and it is alive and well. Gartner's (2014) most recent survey of over 2,500 CIOs shows that BI and business analytics is the number one priority technology investment for 2015 (Safwan et al., 2016).

DSS has proven very effective for businesses, a true guide for better choices and decisions and useful for all types of organizations and areas. Even after many decades since their appearance, researchers are still investigating the potential of the DSS through various fields and from different angles and aspects besides business and ICT engineering.

For instance, using the DSS for environment protection, environmental DSS (EDSS) have been developed to assess the impact of utilization of natural resources and evaluate the impact of agricultural and industrial activities on the environment (Kersten and Lo, 2002). Sustainable manufacturing means the production of goods in such a way that it utilizes minimum natural resources and aims for a cleaner, safer production (Vinodh et al., 2014), DSS was widely used in this area of sustainable manufacturing and preserving.

Water resource management is another facet of environmental decision-making where DSSs have been applied. Water quality management decision support was discussed by some researchers in the 1990s, regarding water delivery maintenance and planning, forestry preserving, calculating threat impacts on forests or natural resources. All of this proved applicable for DSS to interfere in and provide decision support for managers, governments and environmentalists.

Another application domain is clinical DSS (CDSS), reflecting the use of DSS in medicine. DCSS link health observations with health knowledge to influence health choices by clinicians for improved health care as defined by Dr. Robert Hayward, Centre of Health Evidence. (Kawamoto et al., 2005) proved that 61% of healthcare organizations have improved their performance with DSS and this success relates to the fact that CDSS is integrated in the clinical workflow rather than separate, that CDSS provides real-time decision support rather than prior or after the patient encounter and that CDSS is electronic rather than paper-based. However, (Black et al., 2011) conducted a more recent systematic review on e-health including CDSS and seemed less optimistic by questioning the cost-effectiveness of CDSS and the gap between what is promised by researchers and what happens in healthcare environments.

Additional field that DSS proved to be applicable in is higher education. A relatively recent research work still considers DSS as an important, non-disposable technology that has passed from a “deep-seated progress that altered the way information systems is perceived in higher education’s, to a main stream IT progress that all organizations take on” (Fakeeh, 2015). A DSS in an educational environment is expected to assemble information on every educational methodology, offer feedback towards their change and help with decision-making with high integration and direct association with all the zones of the issue, like deciding how many students should enrol, their scientific background and profiles, the nature of courses and methodologies of teaching regarding the professional market needs and the university budget, etc. All of these decisions vary from structured to unstructured and it would be useful to rely on a DSS.

Several other areas have included DSS as part of their assets, whereby the only difference is how much they consider the reliability and the integrity of DSS after all, and not all areas have reached maturity while dealing with decision support software and applications.

1.3 Problem Definition

As shown in the previous section DSS has been well investigated by academics in different fields and especially in the sustainability field. The services offered by EDSS can be considered as helpful yet insufficient: Assessment and monitoring performance indicators do indeed enhance visibility to the organization’s efforts towards sustainability. However, this does not guarantee the understanding of the sustainability impact of certain business decisions, there is no demonstration of the cause-and-effect relationship between decisions and the enhancement/deterioration of performance indicators. Thus, a decision’s sustainability can’t be evaluated based on evidence and logic to ameliorate future decision-making. In addition, in order to evaluate the current situation in the domain of DSS, the recommendation of (Phillips-Wren et al., 2009) in their work on IDSS was used. They annotate that the evaluation of a decision support system should be made among other based on the understanding and use of Simon’s phases in the decision-making process (Herbert A Simon, 1960).

DSS Driven by	Collects and processes data	Collaboration	Information sharing	Context information	Decision information
Data	X	--	--	--	--
Communication	X	X	X	X	--
Group	X	X	X	X	--
Document	X	--	X	--	--
Model	X	--	X	X	--
Knowledge	X	--	X	--	--
Web	X	X	X	--	--
DES[1]	X	X	X	X	X

Table 1.1 DSS comparative table

We noticed – as shown in the table 1.1 – that none of them cover all outcomes and especially the ability to deliver information about the decision itself (decision-maker information, decision date, decision objectives, goals and targets, deadline to achieve the expected goals, etc.). According to the understanding of Simon’s decision-making process, the DSS should enable decision-makers to learn

[1] Proposed Decision Evaluation System

from their past choices. It is important to obtain information about the decision itself through storing, tracking, recommendation and evaluating decisions to enable this learning (Rezgui et al., 2017).

1.4 Main Contributions

The management of decisions over time is one of the major contributions of current work. The evaluation of past decisions makes it helpful for companies to perform more optimized and efficient decisions in the future. In case a company relies on the experience of past decisions, it will significantly improve reduced uncertainty and confusions existing from a historical perspective when the first similar decisions were taken. For this purpose, we need to categorize the present situation to predict appropriate decisions and/or actions. In other words, the evaluation of past decisions is an essential activity to make future decisions more efficient and optimal.

Activities behind the management of past decisions of a company include storing, tracking, evaluating, ranking and recommending decisions. Such an evaluation presents gathered advanced knowledge in a way that a company can observe its decisions in a format that suits its expectations and requirements. For instance, well-adapted BI tools such as reports and dashboard can be used to visualize each single decision taken in the past, whereby they can be seen on a dashboard along with the reputation of a single decision based on its sustainability evaluation and total number of its occurrences over time.

The objective of this work is the design and implementation of a decision evaluation system to enhance sustainability within organizations. This novel system should:

- Improve sustainability within organizations.
- Enabling evidence-based decision-making.
- Empowering more decision-making transparency.
- Overcome the failings of classical DSS: human-machine interactivity, impact demonstration, tracking, evaluation and recommendation of decisions.

In order to achieve the aforementioned objectives, the following research question should be answered: How can decisions be evaluated based on their sustainability impact?

Sequentially, this main question is divided into three partial questions:

- What criteria should be taken into account in evaluating decisions?
- What are the main processes of a decision evaluation system?
- Which generic artifact has to be developed to support the decision evaluation concept?

1.5 Thesis Structure

This chapter introduces the work through representing the motivation, related work including a literature review of DSS with their application domains and the problem definition. In addition, the main contributions, the research questions and objectives are defined.

The second chapter handles related background concepts and technologies in relation with the research domain decisions support systems, environmental management systems and decision-making.

The third chapter shows the methodological approach followed by the establishment of this work. It gives an introduction about design science, the motivation of using this methodological approach and demonstrates how it was used to develop the decision evaluation system. An understanding of the information systems research framework and the design science research methodology mapped with the research activities and deliverables is presented.

The fourth chapter illustrates the conception and design of the decision evaluation system. It includes the functional and non-functional requirements, use cases, process interactions and data structure for the global system and for each component.

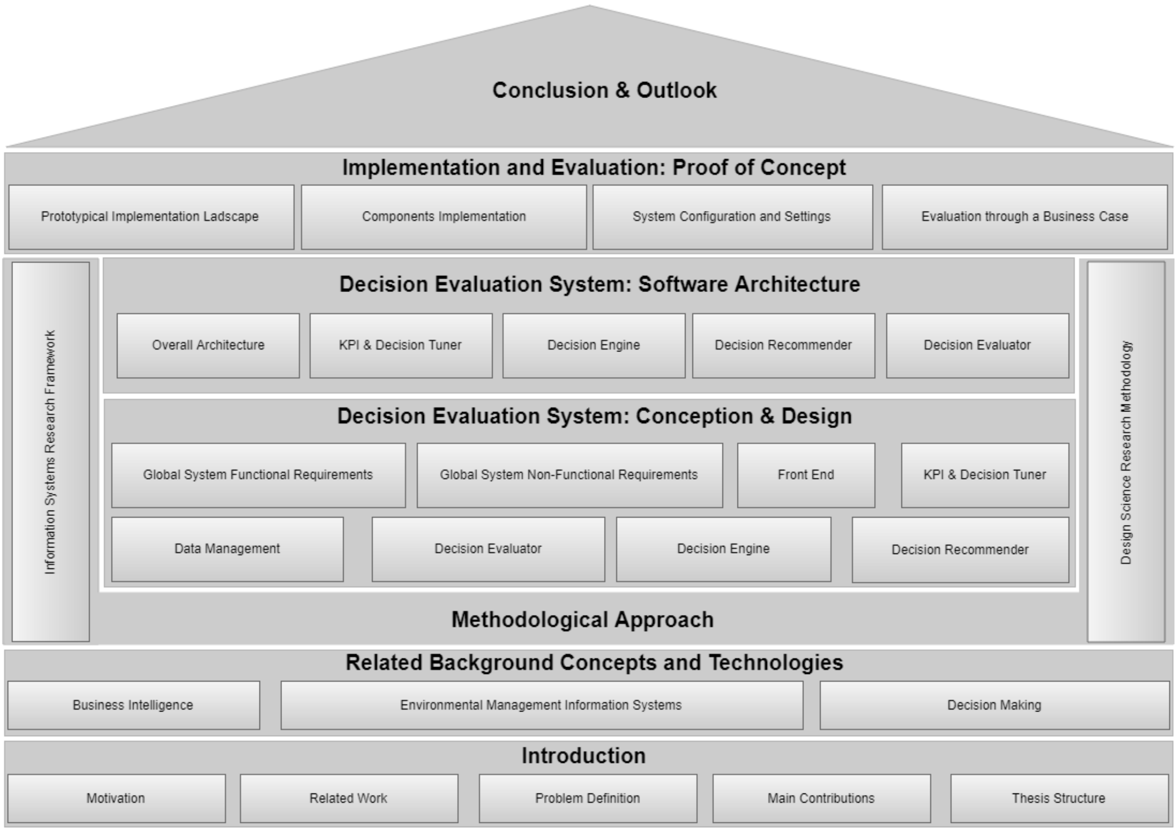

Figure 1.1 Thesis structure

The fifth chapter illuminates the software architecture of the decision evaluation system, all components, sub-components and the interactions between them in the form of component diagrams using the unified modelling language.

The sixth chapter demonstrates the prototypical implementation including the implementation landscape, technologies used and description of the components (classes, attributes, functions, etc.). Furthermore, it validates the idea of evaluating decisions based on their sustainability and proves its applicability in practice. It concludes an industrial business case.

The final chapter summarizes the work, discusses the limitations and offers an outlook for future directions.

2 Related Background Concepts and Technologies

2.1 Business Intelligence

Business intelligence (BI) is a kind of decision support system. In 1989, Howard Dressner from the Gartner Group introduced BI as an umbrella term to describe concepts and methods that are dedicated to improving business decision-making process by means of support systems, which is based on pure facts. A few years later, the definition of BI was updated by (Kemper et al., 2006), who collected and identified seven different definitions of BI. In particular, according to the work of Kemper et al., BI is equal to a data warehouse, alerting system, advanced management information system and list of other systems with intersecting functionality. (Felden et al., 2015) claimed that: “the amount of empirical work in BI has been substantial and diverse. A precise assignment of the term BI is challenging and there is a lack of a consistent definition in literature”.

However, it is very important not only to treat a particular BI system as an extension of a data warehouse and limit its role only to transferring information from an operative information system that is in charge of transactional data via online transaction processing (OLTP) into a system that supports OLAP analysis required by company’s information reporting policies (Gómez et al., 2008). According to (Turban et al., 2011a) and (Rezaie et al., 2011), a BI system can be defined as “an umbrella term that encompasses tools, architectures, databases, Data Warehouses, performance management, methodologies, and so forth, all of which are integrated into a unified software suite or package”.

The BI concepts are widely adopted in industry and well explored within academia. The main motivator to implement ideas behind BI into a software system is to enable decision-makers to possess the required knowledge to perform optimal decisions. Knowledge that is usually offered to decision-makers is based on extracted information from various structured, semi-structured or unstructured data sources like data warehouses (DWHs), ERP systems, web services, CRM systems, flat files, third party systems, etc. In order to provide a decision-maker with required knowledge, the data is first loaded into a dedicated central multidimensional data warehouse, and later on processed by means of online analytical processing (OLAP) analysis, reports or other types of data investigations techniques. After processing data in a desired way, the presentation part takes place. The presentation of knowledge to the user (namely the decision-maker) is required to facilitate a smoother decision-making process.

BI is considered in this work as a system that is dedicated to supporting a decision-making process. It comprises of many different methods and techniques. For instance, a typical process to be conducted to move data from various systems into a BI system is known as an extract-transform-load (ETL) process. Other tools that play a fundamental role in the BI concept are data quality assurance, data warehousing, master data management, web data management and many others. The usage of

BI can be helpful for any actor within a particular organization. It can influence decisions made by a particular employee in an organization, regardless of her/his position, responsibilities and assigned company department (e.g. human resources, sales, marketing, research and development, etc.). It helps them to have appropriate knowledge about the factors affecting their daily business routines and support them in decision-making process.

BI systems require a substantial degree of technical competence to deal with automization, data integration, and the fast, reliable availability of information (Felden et al., 2015). Figure 2.1 demonstrates a typical architecture of a BI system that performs the ETL process to extract, transform and load data into a particular centralized data warehouse. Later on, loaded data is processed and delivered to end users in the form of reports, OLAP cubes, ad-hoc queries, etc.

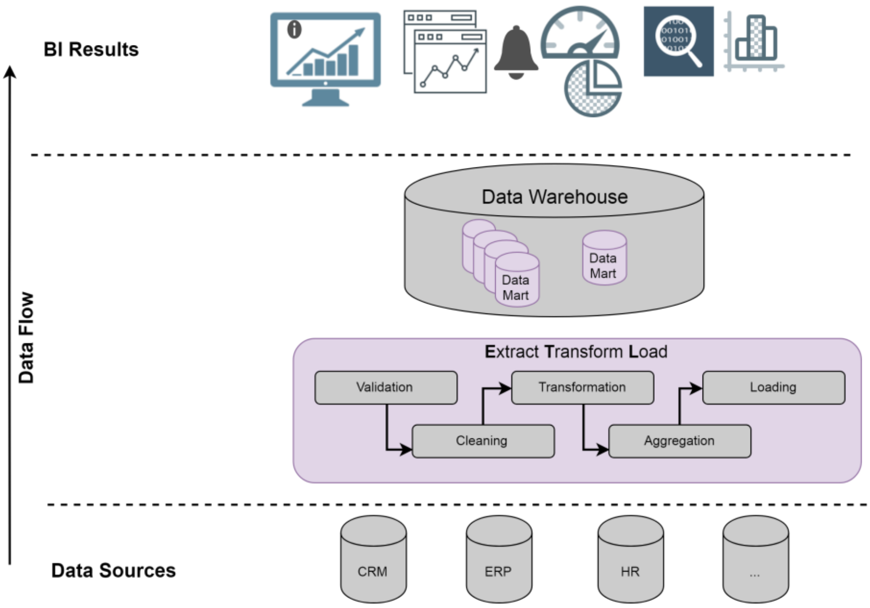

Figure 2.1 Business Intelligence Architecture

Based on (Kimball, Ralph, 1996)

In order to facilitate the analysis and discovery of knowledge within multidimensional data located inside a data warehouse – which should ultimately be delivered to business managers – (Codd et al., 1993) introduced the OLAP concept and defined it with twelve rules. OLAP offers the possibility to explore, aggregate and visualize data using its operators. While implementing BI concepts into a software solution, the ability to capture data required by business users is a necessary step. Indeed, usually providing appropriate information that should be directly integrated into a data warehouse is the most expensive phase in terms of time and resources. A manager who is in charge of implementing a BI solution first of all has to detect exact information that could be useful for stakeholders in the decision-making process. After identifying suitable information required for knowledge that influences a decision-making process, a manager usually should perform an extract step, which includes data acquisition from heterogeneous sources (e.g. R/DBMS, ERP, Excel files, flat files, etc.) in various formats. The next stages after the data extraction process can be either transformation and load or load and transformation. Moreover, the reality depends on a schema of data load chosen, namely whether it is ETL (extract-transform-load) or ELT (extract-load-transform). The transformation step includes all activities related to manipulations on data made after extraction (or load). It mostly involves transformations of operational data into specially-formed data, which can be interpreted in terms of business and economy. It is a composition of several sub-processes, i.e. filtering (eliminating redundancies and outliers), harmonization, aggregation and enrichment (Kemper et al., 2016). Loading steps are dedicated to bringing the data into a central multidimensional database and/or a data warehouse.

The terms used in the context of data warehouse are dimensions, facts, aggregations and hierarchies. Depending on business needs, the data storage schema can be implemented in the view of a star, snowflake or galaxy. Usually, data within a data warehouse is grouped into specialized data marts, which are defined in functional terms. Data marts may include subject-specific, previously-aggregated, historical, current and planned data (Rezgui and Naana, 2010). As previously mentioned, in order to bring data pulled from various data sources in a centralized data warehouse into a form that can be interpreted by a human, BI solutions play a major role. They help users to have different views on data, based on the needs analysis of a particular user. For instance, a BI user may want to interpret her/his data in the form of a pie chart to illustrate numerical proportion, or she/he can use a bar chart to show comparisons among categories. Presenting information in the right form plays a significant role in understaffing subjects, and as was mentioned by (Gluchowski et al., 2008): “finally, the value of the information [...] not only depend on the offered content, but also on the chosen form of presentation.” However, besides offering functionality, BI solutions also take care of the user experience as a whole. For instance, both data preparation and presentation in the form of reports are well presented in a number of software solutions. However, to improve the user’s performance in terms of data analysis, many BI tools offer a user-friendly graphical interface, drag-and-drop techniques for creating ad-hoc reports and other functionalities. These tools – which are suitable for employees lacking profound IT knowledge –thus enable them to perform evaluations without needing to forward their requests to IT specialists. Standard reports and dashboards are prebuilt functionality and are normally available out-of-the-box. They usually contain pre-calculated indicators and serve as a basis for decision-making. Another instance of information presentation is balanced scorecards (BSC), which describe various tasks such as activity planning, communication and the inspection of key performance indicators (KPIs). BSCs empower users with other possibilities to perform data analysis and reporting.

We can summarize that the main scope of BI is the internal and external data and information management without direct consideration of the link to the business processes. In BI-oriented literature, there exist several approaches that refer to process analysis models like Operational

Business Intelligence (OpBI) for instance. OpBI focuses on the analyses of business processes and their connections with further information. The results of the application of OpBI are suggestions for many users in order to improve the control of business processes during progress (Bauer and Schmid, 2009; Felden et al., 2010).

The difference between OpBI and classical BI is the focus on the process, OpBI is concerned with the process state while other BI approaches rather focus on the results of a process (Bauer and Schmid, 2009). Almost on the same period of the emergence of OpBI, another term has surfaced: Business Process Intelligence (BPI). Initially, OpBI and BPI were used as synonyms, but BPI is wider than OpBI since it aims at the improvement of processes (static and dynamic), which focus on process identification, process analyses, process simulation (not only process control like OpBI) (Felden et al., 2010). (Hosny, 2009) further claims that those operations are conducted at the time of construction and during the progress of a business process.

Thus BPI refers to the application of BI techniques to business processes (Grigori et al., 2004) and provides a better understanding of how there are executed (Castellanos et al., 2009). Based on (Weske, 2012) BPI is is a systematic approach for supporting business processes (sets of interrelated activities) using methods, techniques, and software to design, enact, control, and analyze operational processes. The convergence of this approach with BI was predictable according to (Felden et al., 2010) since their intersection deals with analysis and design of economic processes. He presented the characteristics of BPI in a form of a morphological box. The main findings is that BPI focuses on business-oriented process design and redesign rather than process execution, it supports the tactical and strategic decision-making, and it relies on the inspection of historic data that can be structured or unstructured and from an internal or external sources. Two technologies are related to BPI which are: (i) Process Warehouse (PWH) to analyze and store process logs (since data is historic) and, by this context, (ii) Process Mining (PM) to discover process structures (Van der Aalst et al., 2007). Those characteristics of were used to formulate a clear definition of BPI:

"*Business Process Intelligence (BPI) is the analytical process of identifying, defining, modeling and improving value creating business processes in order to support the tactic and strategic management.*" (Felden et al., 2010).

The next section provides more details about the main concepts and techniques used under the umbrella of the term BI.

2.1.1 Data Warehousing, Data Mining and OLAP

Data Warehouse

A data warehouse (DWH) is simply a set of databases created to provide information to decision-makers (Cooper et al., 2000). They are dedicated to storing all of the organization production data and they are reserved for the purpose of analysis and reporting.

DWHs play the role of a central repository for the company since they allow federating data often scattered in different databases. It offers a global vision and guide for decision-makers and enables them to deal with market and company changes through providing comprehensible, useful and fast access to information.

The need for such an approach comes from the massive amount of data collected by organizations, which need to keep an integrated view of their business. DWHs provide this with dimensional modelling of huge data, making the latter visible to managers in a meaningful view (decision-makers). DWH also represents an important field in academia and practice. Different research works relevant to industry have emerged since the 1990s. The major contribution of data warehousing to information science (IS) theory is dimensional modelling (Kimball and Ross, 1996). Using dimensional models enables very large data sets to be organized in ways that are meaningful to managers. The data warehouse is not a one-to-one copy of production data; rather, it is organized and structured based on rules. As the founder of the concept, Inmon (2002) lists four aspects of a DWH:

Subject-Oriented

This aspect means that the data should be structured based on its business nature (theme-specific: subject) rather than the transactional process. Products, suppliers, sales and customers are a representative example of subjects. Any subject is transversal to the functional structures of the organization. The integration of the different subjects is undertaken in a single structure, which enables having all of the related data for one subject in a single place.

Integrated

Loading the data from diverse heterogeneous sources represents a challenge. Subsequently, the data is stored in the transactional systems in a different way (nomenclature, encoding, taxonomy, etc.). In the DWH, it should be standardized and harmonized before being offered for use. It mainly involves giving them unique meaning, coding and description that is comprehensive for users.

Non-volatile

The first and main objective of storing data is to simplify the access and use of it by the user. It means that the data should be designed for read access. Data is considered volatile when it is frequently updated like in transactional systems. In a DWH, the data should not disappear and not changed over processing.

Time-variant

The non-volatile aspect natively induces the time-variant aspect. Therefore, the DWH stores the history of the values that the data will have taken over time. Accordingly, the data is also time-stamped. Consequently, it is possible to visualize the evolution in time of a given value. The degree of detail of the archiving is naturally related to the nature of the data. Not all data is worth archiving.

The quality of data before and after its integration within the DWH can be considered as one of the main factors determining the success of the decision-making system.

Data Mart

Rather than focusing on the universality of subjects, a data mart (DTM) focuses on a topic, subject or department. The DTM can be defined as the "lightweight" version of the DWH. For example, there are data marts for marketing, sales, production, human resources, etc. The objective of a data mart is to permit easily access to data classified by themes of use. It also allows the distribution of some or all of the data stored in a DWH. It offers optimal use of data in a structured business view to specific analysis tools; for instance, OLAP and data mining (DM).

Data Mining

Data mining can be viewed as a result of the natural evolution of information technology. The database and data management industry evolved in the development of several critical functionalities: data collection and database creation, data management (including data storage and retrieval and database transaction processing) and advanced data analysis involving data warehousing and data mining (Han et al., 2011). Data mining covers a set of tools using techniques and methods enabling the exploration and analysis of a large volume of data to discover knowledge and derive patterns from previously-anonymous elements. It allows finding original structures and informal correlations between data and discovering links between apparently distinct phenomena and anticipating trends that are not yet discernible. Data mining methods and techniques are applied on the data stored in the DWH or DTM. Choosing the appropriate technique is based on the data and the analysis to be undertaken. There are six different techniques of data mining:

Association: also known as the relation technique, looking for patterns in which an event is related to another event. A pattern is learned based on a relationship among objects in the same matter.

Sequence analysis: for patterns in which one event leads to another event later. It enables discovering similar patterns, regular events or trends in transaction data over a time or season.

Classification: is used to order and group each element in a pre-defined set of classes or groups. It enables the automatic assignment of data into groups.

Clustering: visually finding and documenting groups of previously-unknown facts. Data is grouped in relation to logical relationships or customer preferences. Unlike classification, the clustering technique automatically defines the classes (not pre-defined) and assigns the data accordingly.

Prediction: using historical data to discover patterns that can lead to reasonable predictions about the future. In other words, it discovers the relationship between variables in data. This type of data mining is also known as predictive analysis and it is used – for instance – in credit scoring.

We can recapitulate that data mining aims to identify possible correlations in a large volume of information system data to identify trends. It can help to create valuable information in the form of cubes. However, it is insufficient for businesses as it lacks the visualization features and the navigation into and through data at different levels of granularity. This motivated the use and application of OLAP technologies.

OLAP

Since the early 1980s and with an explosion in the mid-1990s, companies have been equipping themselves with management solutions and storing an increasing amount of information. In this way, decision-making informatics is developed with the purpose to collect, consolidate and synthesize information to help decision-making.

The OLAP concept was initially in the year 1993 by Codd et. al. introduced with the aim to answer multi-dimensional analytical (MDA) queries rapidly (Codd et al., 1993), usually described as "complex" user queries. Those queries are to retrieve and present measurable data related to multiple dimensions (e.g. time, product, customer, region etc.). This enables a non-technical user to easily and selectively extract and view data from different points-of-view by interactively generating ad-hoc queries without the intervention of IT professionals. An OLAP tool generates an n-dimensional matrix presentation of at least part of an m-dimensional database including data records and at least two key dimensions. Each key dimension includes data value fields. Each data value field has real data therein (Arras and Steinhoff, 2008).

Data mining and OLAP can be used and combined in several ways. Whereas data mining prepares the information analysis infrastructure, OLAP presents the results in the form of reports and dashboards at diverse altitudes. This infrastructure includes the exploration and discovery of hidden knowledge and patterns.

According to Codd there are twelve rules for selecting tools for OLAP (Beynon-Davies, 2004) (see OLAP.com[2]):

[2] http://www.olap.com/learn-bi-olap/codds-paper/

Multidimensional conceptual view

OLAP tools should provide a multidimensional model which corresponds with the user's views of the organization. Such a model should also be easy-to-use.

Transparency

The technology used, the underlying database architecture and the various data sources should be transparent to the user und consistent of an open system.

Accessibility

The OLAP tool should be able to access data from sources in different formats – relational, non-relational and in terms of legacy systems.

Consistent reporting performance

The user should not perceive any degradation in performance when the number of dimensions, aggregations or the size of the database increases.

Client/server architecture

The OLAP tool should be able to work effectively in a client–server environment.

Generic dimensionality

Every data dimension should be equivalent in its structure and operational capabilities.

Dynamic sparse matrix handling

The OLAP tool should be able to adapt its physical organization to optimize the handling of sparse matrix handling.

Multi-user support

OLAP tools must provide and handle concurrent retrieval and update access, integrity and security.

Unrestricted cross-dimensional operations

The OLAP tool must be able to support dimensional hierarchies and automatically perform consolidation calculations within and across dimensions.

Intuitive data manipulation

Pivoting, drill-down and consolidation should be accomplished using classic graphical user interface operations such as point-and-click and drag-and-drop... It should not require the use of a menu or multiple trips across the user interface.

Flexible reporting

Reporting facilities should present information in any way the user wants to view it. It must be possible for a user to arrange rows, columns and cells at will in a fashion that meets its needs.

Unlimited dimensions and aggregation levels

The OLAP tool should not impose any restriction on the number of dimensions or aggregation levels in an analytical model.

The analysis contexts (dimensions and dimension hierarchies) play a major role in the data aggregation and define the way in which the data should be stored in the OLAP database. This form of data storing is called a cube. A 3-dimensional data cube is displayed in this next figure to represent the emissions quantity of vehicles (referred to as measure) for 3 dimensions:

- Emission (e.g. Nitrogen Oxide [NO], Carbon Monoxide [CO1], Sulfur Dioxide [SO2])
- Cars (e.g. by brands [BMW, GM..] or by types [SUVs, pick-ups...etc.])
- Time (e.g. by years, quarters or months...)

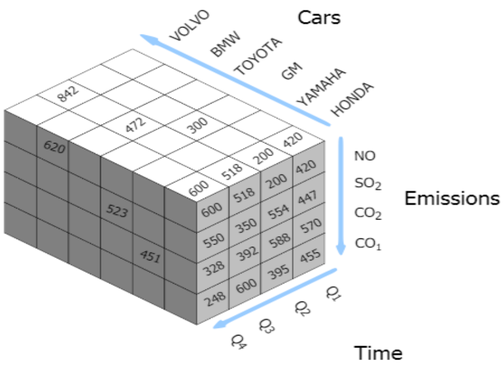

Figure 2.2 Environmental Data Cube

The association of the data in the cube is business-oriented and should be easy-to-use for non-technical users. Functions like year-to-date (YTD), year-to-month (YTM), what-if analysis, running sum, etc. are typically supported by OLAP. A classic example is the identification of potential annex products in a sales transaction. Thanks to data mining techniques, the user is informed about the annex products sold in relationship with other products. OLAP allows him/her to navigate within the information through different techniques like drill-down, drill-up, slice, dice, pivoting, etc. Among those many possible operations that can be applied on multidimensional data in OLAP, the following operations are the basic, most popular end-user operations:

Roll-up allows to create a data cube that contains instance data on a higher aggregation level. Roll-up is executed by replacing the inquired members of a dimension with members of a higher level (Kämpgen et al., 2012).

Figure 2.3 shows an example of this operation on a 3D data cube where the roll-up operation was on the cars dimension from brands (Yamaha, Honda, GM, Toyota...) to types (SUVs, Pick-ups, Vans).

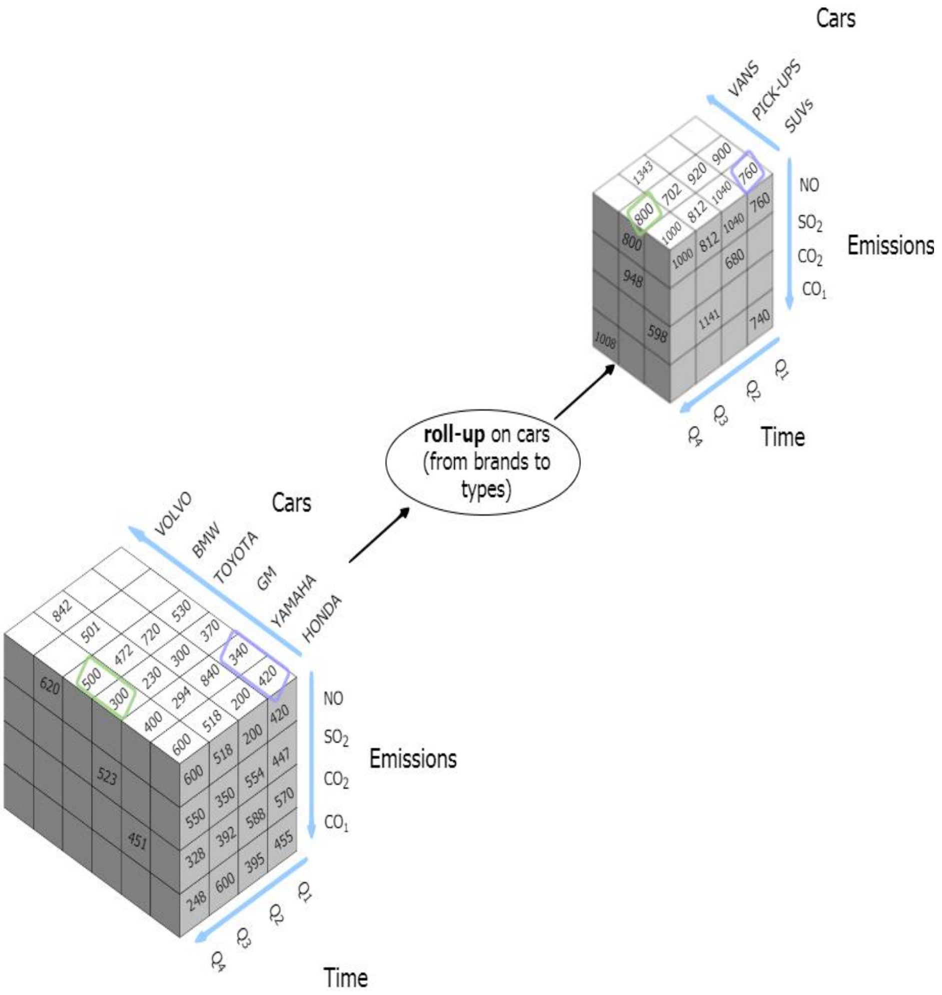

Figure 2.3 Roll-up operation

Drill-down (also called roll-down) this is the counterpart of Roll-up. Thus, it removes the effect of that operation by going down through an aggregation hierarchy, and results showing more detailed data (Abelló and Romero, 2009). Drill-down is executed by replacing the inquired members of a dimension with members of a lower level. An example of this operation is ta drill-down operation on the time dimension from year quarters [Q1…Q4] to a lower-level of Months [January... December].

Pivoting (also called rotation) it refers to inverting the view of cube by rotating data dimensions (or axes in 2D views) to enable views from different perspectives. Figure 2.4 shows an example of this operation on a 2D data matrix where the pivot operation rotated the dimension cars from the x-axis to the y-axis.

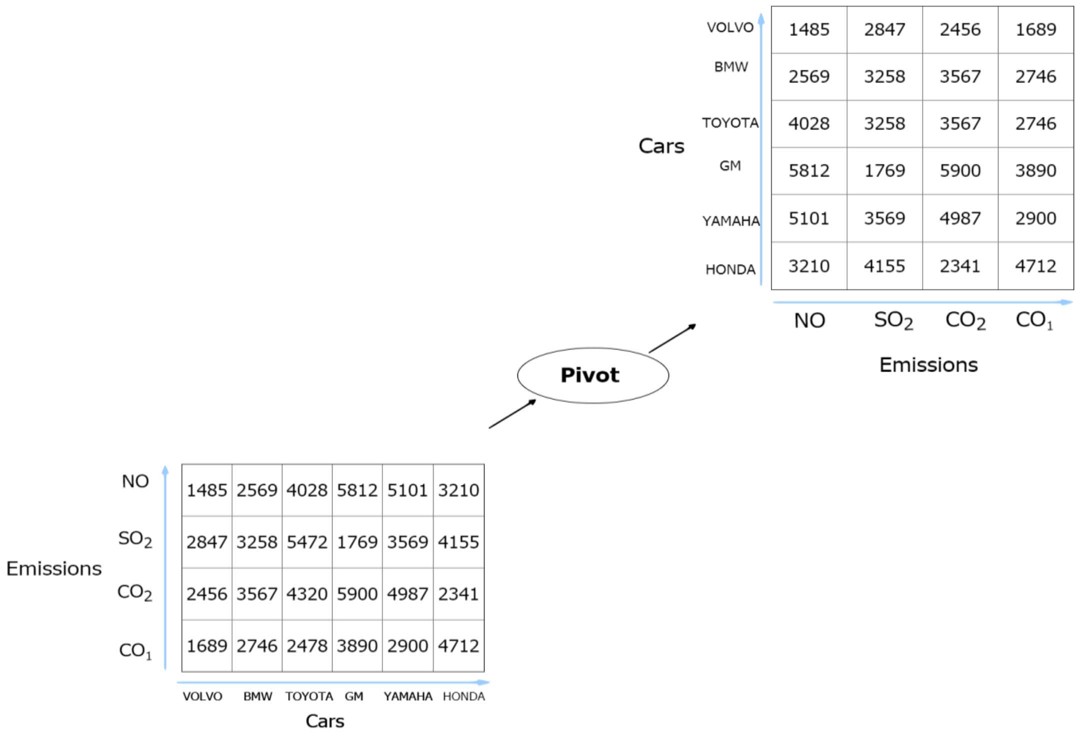

Figure 2.4 Pivoting operation

Slice This operation removes dimensions from the input cube and aggregates over the members of a single dimension to be viewed separately (Kämpgen et al., 2012). Reducing the dimensionality of the view could be for the importance of a particular dimension for the decision-maker. Figure 2.5 illustrates an example of this operation where the slice operation applied in a 3D data cube. The slice was on the criterion 'Q4' from the dimension time. This eliminates other members of this dimension (Q1, Q2, and Q3) and shows data of all members from the remaining dimensions emissions and cars respecting this criterion. The result is a new sub-cube with n-1 dimensions, n=3 in this 3D cube and 3-1=2 so the result is shown in a 2D matrix.

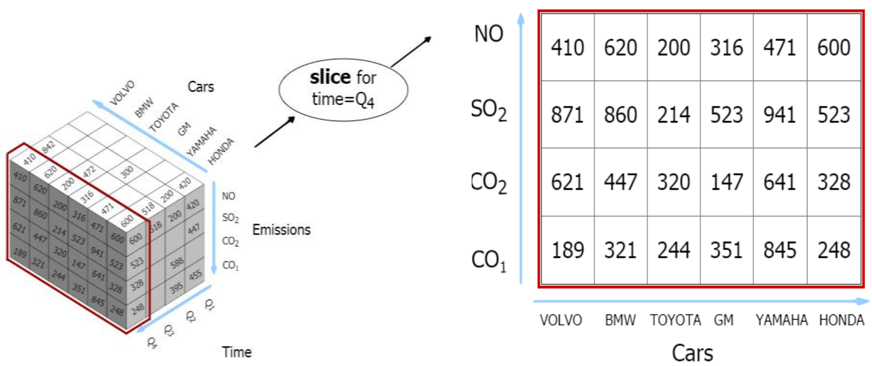

Figure 2.5 Slice operation

<u>Dice</u> (Or Selection) this operation allows users to choose the subset of points of interest out of the whole n-dimensional space (Abelló and Romero, 2009), this means the possibility to filter for ,and, aggregate over certain dimension members in order to obtain a sub-cube containing two or more dimensions (Kämpgen et al., 2012). Figure 2.6 shows an example of this operation applied on a 3D data cube. The dicing was for selecting specific members (criterions) from different dimensions to generate a new, ad-hoc selected sub-cube. The resulted sub-cube concerns the brands (VOLVO, BMW, and HONDA) from the cars dimension, the year quarters (Q1, Q2) for the time dimension and the compounds (NO, SO_2) in the emissions dimension.

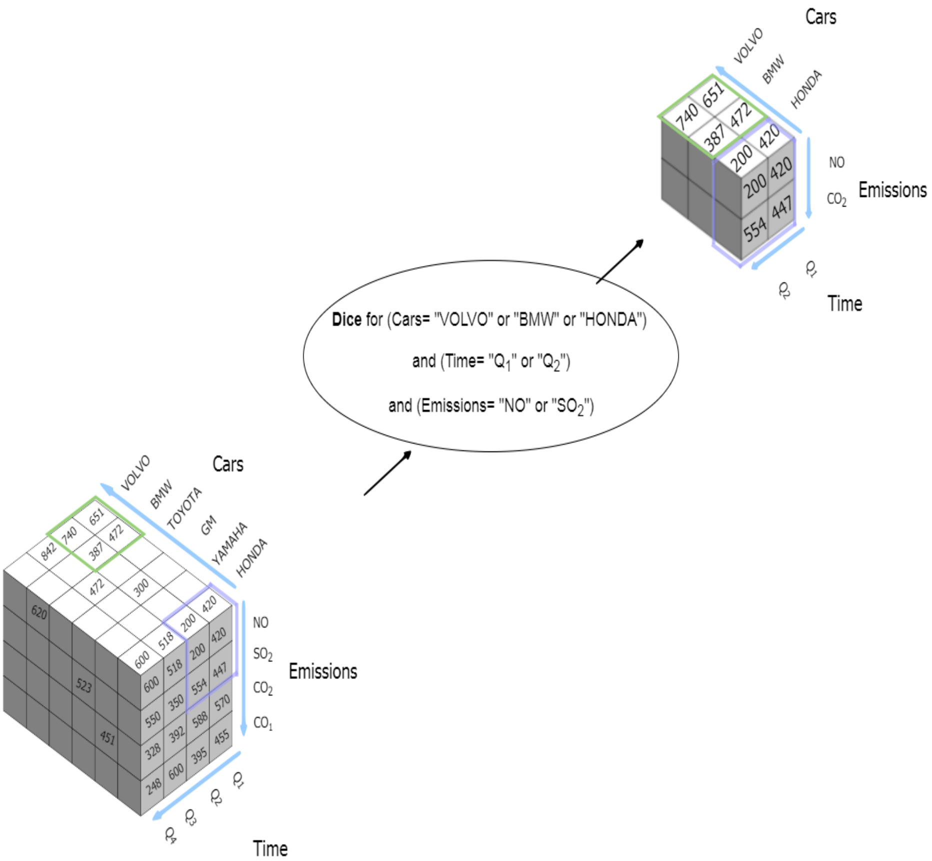

Figure 2.6 Dice operation

The OLAP model may have different physical representations, whereby we distinguish between relational online analytical processing (ROLAP), multidimensional online analytical processing (MOLAP) and hybrid online analytical processing (HOLAP).

OLAP can be categorized in three classes based on how the data in OLAP is stored and how servers are implemented (Chaudhuri et al., 2011; Pendse, 2006):

Multidimensional OLAP (MOLAP): MOLAP servers directly support the multidimensional view of data through a storage engine that uses the multidimensional array abstraction (n-dimensional cubes). They typically pre-compute large data cubes to speed up query processing (Chaudhuri et al., 2011).

This type offers fast queries performances due to optimized storage, indexing and caching and automated computation of higher level aggregates of the data.

However, MOLAP has poor storage utilization for sparse datasets and data redundancy in some cases (Grabova et al., 2010).

Relational OLAP (ROLAP): ROLAP servers support only classic relational databases that are organized to function as an OLAP database. The base data and the dimension tables are stored as relational tables and new tables are created to hold the aggregated information (**e.g.** the usage of star schemas to represent multidimensional data).

ROLAP have no limitation on data volume unlike MOLAP and can leverage functionalities inherent in relational databases (**e.g.** authorization control on row-level security) (Grabova et al., 2010), it is also considered more scalable in handling large datasets.

However, The ROLAP databases are much slower and significantly less efficient than the MOLAP databases. ROLAP servers may also need to implement functionality not supported in SQL, for example, extended aggregate functions such as median, mode, and time window based moving average (Chaudhuri et al., 2011).

Hybrid OLAP (HOLAP): To end the debate about using MOLAP or ROLAP and the compromises between slow query performance and poor storage utilization, the HOLAP model has surfaced as a combination between the two previous models. HOLAP is attempt to address the shortcomings of the previous two models by allowing the model designer to choose which data should be in multidimensional forms and which should be in relational structure (splitting data or partitioning). One method useful to have both of their advantages is to store the detailed data in a RDBMS as ROLAP servers do, and pre-computing aggregated data in MOLAP (Chaudhuri et al., 2011), there are also other methods of splitting.

HOLAP permits fast access at all levels of aggregation; compact aggregate storage; dynamically updated dimensions; easy aggregate maintenance.

However, designing HOLAP can be complex with the necessity to have both MOLAP's and ROLAP's engines and tools in the servers. A functionality overlap can occur between the engines of the two models (Grabova et al., 2010).

Other OLAP types have surfaced such as Web-Based OLAP, Desktop OLAP, Real-Time OLAP... Those types are not as widely used as the three common models above however.

All of the aforementioned techniques and concepts can be considered under the umbrella of the term business intelligence. In the next section, an extension of BI using prediction and optimization techniques to include adaptability in the decision support will be presented.

2.1.2 Adaptive BI

The complexity of software systems and the rapidly-changing environment prompted the software engineering community to look for a concept that allows systems to adapt themselves. Indeed, usually an adaption of particular system should be based on users' behaviours, their profiles or changes in requirements. This motivated an approach towards the creation of self-adaptive systems.

There are a couple of terms defining self-adaptive software systems. One of these definitions is introduced by (Oreizy et al., 1999), claiming that: "Self-adaptive software modifies its own behaviour in response to changes in its operating environment. Here operating environment means anything observable by the software system, such as end-user input, external hardware devices and sensors, or program instrumentation". Later, (Villegas Machado et al., 2011) extended the abovementioned definition of the self-adaptive systems to the following: "Such dynamic systems adapt in response to changes in their environments, either to ensure the continuous satisfaction of their functional and non-functional requirements, or to provide ubiquitous and context-dependent smart services".

As can be derived from the definitions of self-adaptive systems, one of the core properties that they bring to software components is flexibility-by-design. Indeed, a property such as flexibility is a requirement that influences the business intelligence domain and motivates researchers to change (or re-adapt) previously-applied BI concepts and integrate concepts behind self-adaptive systems into them. The next section provides more details about the new generation of BI motivated by the self-adaptive software.

The extension of BI by using adaptability based on prediction and optimization methods and techniques for forecasting and decision supporting is called adaptive business intelligence (ABI) (Rezgui and Ben Maaouia, 2016). It was first introduced by (Michalewicz et al., 2007), who define the term ABI as "the discipline that uses prediction and optimization techniques to build self-learning 'decisioning' systems" (Michalewicz et al., 2007, p. 5). Building such a system requires a good understanding of the techniques for optimization, prediction and adaptability, which are largely documented and viewed in the literature, although combining them into one system was never previously considered (Michalewicz et al., 2007) in his new idea of adaptive BI. It can be seen with clarity that ABI attempts to re-visit the application of AI in DSS with more maturity and experience after these two fields encountered tension in research in the last decade. ABI addresses the core issue of AI, which is how to make computers more useful and intelligent. To date, researches have rarely applied this ABI approach in their works.

ABI has been investigated by different researchers from different perspectives (e.g. adaptability in user interface, adaptability in models, automatic decision-making, and adaptive knowledge presentation) (Nenortaitė and Butleris, 2009) (Fabac, 2010) (Burmester, 2011) (Lau et al., 2012) and (Kim et al., 2013). Another application of ABI was presented by (Nenortaitė and Butleris, 2015), who used ABI and swarm intelligence to improve business rules management.

However, despite progress in research in various directions, most of these initiatives are isolated from each other and do not provide a general integrated overview. Indeed, one of the most crucial points that has been ignored is the adaptability of ABI in a content of gathered/generated knowledge (e.g. decisions), the human involvement in a decision-making process and the recommendation for particular better decisions, which are based on past experience.

In ABI systems, decisions are not evaluated in a periodic manner and the inappropriate decisions of the past might repeatedly occur over time in the future. Such system behaviour prevents companies from receiving benefits in decision-making process, which can be based on historical pitfalls. Furthermore, as previously mentioned, introducing past experience enhances the quality of decisions made within a company over time. The same applies for archiving such decisions considered as "best practices" or the most successful ones. Activities in storing and maintaining a catalog of such decisions and subsequently integrating them into a decisions recommendation system will optimize decisions for a specific issue in a company.

2.1.3 Knowledge Management

As seen previously the main motivator for the implementation of business intelligence systems is providing decision-makers with the required knowledge to perform optimal decisions. This gathered knowledge has to be managed and shared. Thus, knowledge sharing has been identified as a major focus area for knowledge management (KM). The relevance of this theme particularly derives from the fact that it provides a link between the level of the individual knowledge workers, where knowledge resides, and the level of the organization, where knowledge attains its (economic, competitive) value (Hendriks, 1999).

The organization's environment is ever-changing for most businesses and companies and the awareness and reactions of these variations can make or break the organizational survival, competitiveness and profitability (Omotayo, 2015). Organizations can be conscious of their environment (internal and external) by applying studies and analysis in order to gather information.

This should surround the past and present which can be valuable in understanding situation and predict the future.

The useful pieces of information gathered by experience, learning analysis and evaluation are referred to as Knowledge in the literature (Davenport and Prusak, 1998). The term Knowledge often takes a variety of meanings depending on the context. It is closely related to other terms like data, information and wisdom. Knowledge is closely linked to actions (doing) and implies the know-how and understanding (Forst, 2011) so it is viewed a key driver for good decision-making within any organization.

Knowledge Management (KM) is an essential process that is required to take maximal advantages of the gathered knowledge. Organizations may invest in KM to understand (Forst, 2011):

- What they know.
- Where the knowledge is located.
- How to best transfer this knowledge to relevant individuals.
- How to use knowledge in important decisions.

A clear and rather complete definition of KM is provided by (Frost, 2012): "Knowledge management is the systematic management of an organization's knowledge assets for the purpose of creating value and meeting tactical & strategic requirements; it consists of the initiatives, processes, strategies, and systems that sustain and enhance the storage, assessment, sharing, refinement, and creation of knowledge."

Another analogous and more summarizing definition is used by (Skyrme, 2011): "Knowledge Management is the explicit and systematic management of vital knowledge - and its associated processes of creation, organization, diffusion, use and exploitation - in pursuit of business objectives."

However (Uit Beijerse, 2000) presented several definitions of KM with the perspective of structuring and organizing knowledge in addition to a definition from another perspective involving the people behind the knowledge: "Knowledge management is the achievement of the organization's goals by making the factor knowledge productive. This is done primarily by facilitating and motivating people to tap into and develop their capacities (their core competencies) and to stimulate their attitude to intrapreneurship. Besides this, knowledge management includes the entirety of systems with which the information within an organization can be managed and opened up."

To sum up, all previous definitions show the strong dependency between KM and organizational goals and strategies and how KM, if well-executed, can create value for the organization from intellectual and knowledge-based assets. KM can significantly help with the decision-making process, this process is strongly dependent to the information about the business environment and the know-how and experience of the decision-makers to identify decision-requiring occasions and choose the optimal decision based on the quantity and quality of knowledge they have.

It is important to remember that knowledge management is not about managing knowledge for knowledge's sake. The overall objective is to create value, leverage and refine the organization's knowledge assets to meet its business goals (Frost, 2017a).

For every KM project started by an organization, there was a list of objectives to be reached. Each list of set objectives is related to the organization's perception of Knowledge, its abilities, its expectations of KM, its business objectives. Based on (Davenport and Prusak, 1998) the KM objectives can be splitted into four categories via the projects:

1. Creating Knowledge repositories to store knowledge and information, in documentary forms mostly. Those repositories may contain (i) external knowledge (i.e. competition...), (ii) structured internal knowledge (i.e. research reports, marketing) or (iii) informal or tactical knowledge (i.e. discussion database and the know-how) (Rowley, 2000).
2. Providing, improving access to knowledge from experts to relevant individuals and facilitating its transfer. The focus here is on connectivity and telecommunication technologies (networks, video conferencing, sharing tools and applications ...). Passing knowledge from individuals to others could very well prevent or reduce 'Corporate Amnesia' especially since the workforce is more mobile in the last years.
3. Enhancing the Knowledge environment to be more conductive for effectively creating, transferring and using knowledge. This of course involves reviewing and altering organizational norms and strategies related to knowledge. There are many examples of KM projects under this category.

   One organization focuses on knowledge-related employee behavior with, for example, contributions to the organization's structured knowledge base attracting significant rewards and bonuses. Another organization has implemented decision audit programs in order to assess whether and how employees were applying knowledge in key decisions (Rowley, 2000).
4. Managing knowledge as an asset rather than something intangible. In so doing, it enables the organization to better protect and exploit what it knows. By 'assets' it is meant that knowledge is treated such as technologies sold under license or have potential value, customer databases. (Skyrme and Amidon, 1998) for instance proposed that knowledge could be measurable using a balanced scorecard. There still is a debate of metrics for Knowledge to support organizations evaluating their knowledge capital.

KM is about managing people, culture and organizational practices and structures. Therefore, effective KM initiatives are not exclusively technology driven. Although, IT-Based tools and systems can support KM reach its potential by facilitating interaction, exchange of ideas, storing knowledge and locating experts (Frost, 2017b). Based on the study conducted by (Improvement Service, 2009) and the work of (Frost, 2017b) IT-Based KM Tools can be classified as follow:

Collaborative Technologies (Groupware and Web 2.0)

Those technologies enable a group of people to labor together across distance and time to achieve a set of goals. Groupware is one example of those technologies. They can be categorized to (i) Communication Tools, (ii) Conferencing Tools and (iii) Collaborative Management Tools (group activities, project management tools, information management systems...). Observing those categories of Groupware, is obvious that such tools can simplify sharing explicit and tactic knowledge through publication and communication. Collaborative Management Tools can also help creating knowledge.

Therefore Web 2.0 refers to the new tendency of web pages and applications to promote two ways of communication (social networks, blogs, wikis, shared workspaces, media sharing...etc.). The application of this in businesses is called enterprise 2.0 (Gardner, 2013). KM 2.0 also emerged as a term inspired from web 2.0 and it describes the tools that “facilitate the development of social capital through knowledge sharing, which in turn increases the potential to create intellectual capital” (Cronk et al., 2012) .

Instant Messaging Technologies

Those are technologies that enable real-time collaboration between dispersed individuals and groups internally or externally. Knowledge can be transfer from the mind of experts to seekers much faster and on demand for person to person conversations. However, instant messaging proved non-practical for large groups of people interacting at the same time.

Intranets and extranets

The intranet can be considered as small-scale version of the internet existing only within an organization's quarters. It can be valuable tool in the knowledge management process. It permits the integration of multimedia communication and can act as a platform for collaborative, publishing and messaging applications. It is intended to enhance collaboration, productivity, and socialization (Frost, 2017b) .

The extranet is seen as a further extension of the intranet which enables connection with external actors like partners, suppliers and clients. Accordingly, this can enhance collaboration and knowledge transfer between multiple sides. However, it provides limited access and authorizations to external users and applies multiple security measures (authentication, authorizations, roles, profiles, firewalls, encryption...). Security of the organization's knowledge and information resources is a key concern for extranets.

Data Warehousing and Data Mining

Data Warehousing is essentially built to support decision-making within the organization and to contribute in meeting business objectives, like the objective of managing knowledge.

The quality of the available information and knowledge impacts natively the quality of the taken decisions by managers. So, the goal of storing data in a centralized system is to provide them with the right building blocks for sound information and knowledge. Data warehouses contain information ranging from measurements of performance to competitive intelligence (Tanler, 1997).

Data mining is another enabler in extracting usable information and knowledge from data warehouses by exploring hidden patterns of huge amount of data. DM is the analysis step of the "knowledge discovery in databases" process, or KDD, which is an important sub-field of Knowledge Management. (Silwattananusarn and Tuamsuk, 2012) reviewed and studied many applications of data mining in KM over the five years before 2012, they found that DM was successfully used for modeling, clustering, classification and dependency of data for extracting knowledge in several application areas like healthcare, finance, retailing, food supply, construction.

Decision Support Systems (DSS)

As presented in the first chapter, numerous types of the DSS exist and can be classified by what they are based on. In relationship to knowledge management, the focus is on data-driven DSS and Knowledge-driven DSS. The first type enhance the knowledge discovering hidden in databases by providing relevant data to decision-makers. As for the Knowledge-driven DSS, they are used to store, assets and share knowledge between persons through computer based system. The focus of such system is providing knowledge in relationship to problem-solving activities.

As mentioned earlier, there are many types of DSS that are explored and studied in the academic field. The data-driven DSS such as DWH, DM and Knowledge Management DSS (KMDSS) were the most attractive fields for academics in terms of the number of publications in the last decade. (Hosack et al., 2012) predicted that the two DSS types will merge in the future and the focus will incorporate better ways to allow organizational members to interact with available information, wherever and whenever it is available.

Expertise locator

(Young, 2010) claims that an expertise locator technology is one of the key tools of KM because "It is often true that knowing who knows what is more valuable than knowing how to do". An organization should not reinvent the wheel by not knowing who to consult about knowledge in problem-solving or decision-making. He uttered Expertise Locator (Expert Locator, Who’s Who) is an IT tool to enable effective and efficient use and/or share of existing knowledge by connecting people who need particular knowledge and people who own the knowledge.

Expertise Locator can be simple electronic yellow pages, more sophisticated systems to automatically search expertise, or even a mixture of IT and people (often called Knowledge Brokers) who support finding and connecting the person who wants the knowledge and the person who has the knowledge (Young, 2010).

2.2 Environmental Management Information Systems

Boosted energy consumption, increased energy costs, environmental sensitivity and government rules have motivated the establishment of concepts and tools to support green strategies through the management of environmental information (Rezgui and Marx Gómez, 2016a). Being green means reducing carbon dioxide emission mainly by lower power consumption but also by reducing waste or dangerous ingredients etc. can be found as goals to meet (Sauer, 2016). Many countries are working towards the target of reducing greenhouse gas emissions. For instance, in Europe, the UK and Germany have set goals to reduce carbon dioxide (CO_2) emissions by 60 % (by 2050) and 40 % (by 2020), respectively ((Kannan, 2009) Röttgen, 2010). In order to achieve such environmental goals, stakeholders need to have adequate systems supporting the management of environmental information. As result, environmental management information systems (EMIS) have become increasingly important in both the academic and business communities over the past two decades. Indeed, industry studies have highlighted this significant development (Marx Gómez, 2004), (Rautenstrauch, 2013).

EMIS are computer-based systems enabling the collect, management and reporting of environment related information within organizations. Such systems are the result of an increasing need to manage environmental information to meet the internal and external pressures such as government regulations, international standards, consumers, investors and new concurrence rules in the business world.

The objectives of these systems are protecting and preserving natural resources, preserving the overall balance and rationalization of energy consumption. They give stakeholders the ability to assess, optimize and report on the current effects of their processes and operations on the environment. These effects may be measured through a particular type of performance indicators, called environmental performance indicators (EPIs) (Jamous et al., 2013). They also allow the environmental performance examination of individual projects, products, departments, etc. Thus, may engage steps of actions – if needed – to rectify the deviations from the targets. The actions are the management decisions. Therefore, decision-makers are the first responsible for monitoring EPIs and deciding which action should be taken. Trying to assure a high-level quality of their organization’s products and/or services and respecting the environment, managers should follow standards and guidelines by using a sophisticated quality management system (QMS). This can help the coordination of activities in an organization to control and improve the efficiency and effectiveness of its performance (Petkovska and Gjorgjeska, 2013). It assures the ability to measure, continually control and improve their processes performance through indicators. DSS as a set of tools, techniques and processes that support and improve decision-making can be seen as an enabler for organizations to meet their objectives. There are significant synergies existing between EMIS and DSS. While EMIS are responsible for managing environmental information, DSS are the engine for collecting, aggregation, filtering, harmonization and presentation for decision-making proposes. Decision-making is defined as the process of choosing an option from two or many available alternatives, whereby the purpose of making decisions is achieving one or more goals (Turban et al., 2011a). In environmental cases, these goals are a combination of optimizing the business activity (for instance, profit generation) while having the most sustainable results.

In fact, DSS play a critical role in guaranteeing that decisions made by organizations are effective in achieving goals and especially environmental ones. The main advantages of such a system are simplicity, clarification and control. Processes performance should be measured through indicators and they should be continually controlled and improved.

To support green decision-making for organizations, many EMIS and environmental DSSs have been developed, with the objectives of protecting and preserving natural resources, preserving the overall balance and rationalization of energy consumption (Marx Gómez and Teuteberg, 2015). They give stakeholders the ability to assess, optimize and report on the current effects of their processes and operations on the environment. These systems have existed from since at least the early-1980s (Reynolds et al., 2014). In 1989, about 100 DSSs related to environmental management were catalogued by (Davis and Clark, 1989).

In the last decade, many successful EMIS and DSS applications have been developed, providing global and detailed information about sustainability. For instance, Lefroy et al. (Rod DB et al., 2000) developed a framework evaluating sustainable land management (FESLM) in 2000 to assess the sustainability of different land management systems practiced by farmers on sloping lands in three eastern countries. In 2004, (Swanepoel, 2004) developed a DSS for real-time control of the manufacturing process for industries to evaluate its sustainability. Still in manufacturing, (Vinodh et al., 2014) used a fuzzy logic-based model for measuring various factors in an industry to insure sustainability. Sustainable manufacturing means the production of goods in such a way that it utilizes minimum natural resources and aims for a cleaner, safer production. (Šliogerienė et al., 2011) developed a DSS for sustainability assessment of power generation technologies. The list goes on, proving that this field is indeed rich and well considered by academics.

However, they lack in the evaluation, which is a very important part of the decision-making act, since the decision-making process not only involves preparing decisions, but also making and evaluating decisions. The problem with typical DSS is that they are fundamentally designed and optimized around analysing and presenting information with the objective to support stakeholders preparing decisions, whereas information about the decision itself and its sustainability evaluation are not included. Experience and intuition are neither stored nor shared; rather, they are only in the mind of individual managers (Rezgui and Ben Maaouia, 2016). In the next section, after introducing and defining the decision-making, we will present our process understanding and the steps needed to achieve a decision-making process enabling the evaluation of decisions based on their sustainability.

2.3 Decision-Making

Many researchers have explored the decision-making process. One of the first works is the Simon's decision-making process published in 1960. He was awarded with the Nobel Prize in Economics in 1978 for his contribution to science within decision processes in economic organizations ("Nobel Prize Organization," n.d.). According to a study conducted by (Arnott and Pervan, 2005b), Simon is the most influential author of reference research in DSS. In 2009, Phillips et al. (2009) conducted an evaluation of DSS. They claimed that the evaluation of a decision support system should be made among others based on the understanding and use of Simon's phases in the decision-making process (Herbert A. Simon, 1960). In this chapter, our understanding of Simon's decision-making process will be presented.

The phases of the process are illustrated around three activities: intelligence, design and choice.

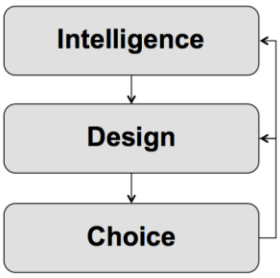

Figure 2.7 Simon Phases Process

Based on (Simon, 1977)

Whereas Simon highlights that decision-making includes four principal phases – finding occasions for making a decision, finding possible courses of action, choosing among courses of action and evaluating past choices (Simon, 1977) – there is a flow of activities from one phase to the next. At any time, there may be a return to a previous phase. The first phase – called intelligence activity – involves finding occasions for making a decision. This phase is the trigger event for a decision issue. It is accessible when the irregularity between the current situation and the target condition can be diminished and/or overcome through assorted outlines (Grünig and Kühn, 2005). Traditional DSS like BI assure this activity through alerting stakeholders via e-mails, notifications, dashboards or reports, based on pre-defined business rules. If one or many indicators exceed the threshold, the BI system automatically starts the alerting (push technology). In other words, in order to start the decision-making process, problems for an opportunity situation requiring design and choice should be identified. The nature of the decision should be recognized. The identification and classification of the problem are the main outcomes of this phase. Subsequently, during the second phase – the design activity – various courses of actions should be exanimated. It incorporates information collecting and gathering from internal and external data sources. This task helps decision-makers to understand the problem and the cause. It includes subsequently exploring different scenarios, inventing, developing and analysing various alternatives. This encompasses processes to understand the problem and generate one or many solutions. The solution may be a known decision taken in the past or a new one. In the first case, we read about simple or programmed decisions. Simon differentiates between unstructured (non-programmed) and structured (programmed) decisions (Simon, 1977): structured decisions are habitually simple decisions based on “if-then relationships” and they follow a well-known process. For instance, if the price of the row material increases by X %, then the end product price increases by X %. ABI tools are the best kind of DSS covering such decisions.

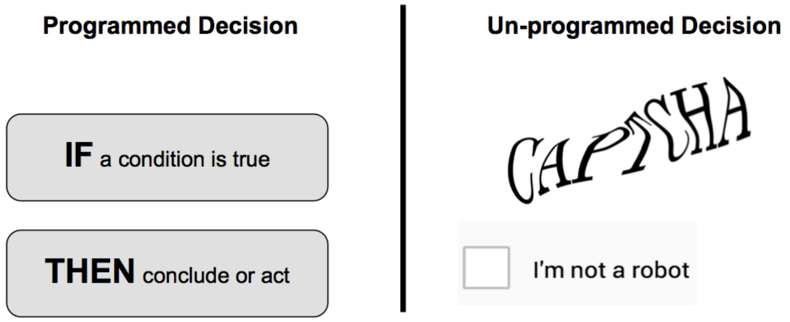

Figure 2.8 Programmed vs. Un-programmed Decision

Michalewicz et al. define adaptive BI as “the discipline of using prediction and optimization techniques to build self-learning ‘decisioning’ systems” (Michalewicz et al., 2007). Unstructured decisions are more complex and need special treatments. It is not unlikely in this case that the solution comes after rounds of brainstorming, knowledge sharing and discussions. In the choice activity phase, the best course of action should be selected. According to (Turban et al., 2011a), decision-making is a process of choosing among two or more alternative courses of action for the purpose of attaining one or more goals. Therefore, choosing an action includes natively designing goals to be achieved through the decision. The decision can only be evaluated based on measuring the achievement rate. This measurement gives more objectivity to the evaluation. It allows decision-makers to generate knowledge about the decisions and thus re-evaluate their past choices. This reevaluating process becomes a cycle. When organizations repeat this learning cycle, they gain a strong empirical understanding of how their business operates and how its decisions and actions affect the marketplace and vice versa (Eckerson, 2003).

For many organizations, a managerial decision is treated as a ‘black box’, subject to neither an explanation nor a review. The human decision-making process is invisible ((March, 1987) (Davenport et al., 2001) (Kahneman, 2003) (Lindvall, 2013). Simon highlights that decision-making includes four principal phases: finding occasions for making a decision, finding possible courses of action, choosing among courses of action and evaluating past choices (Simon, 1977).

Thus far, the traditional DSS deal only with the first phase that Simon cited, namely intelligence activity. All three phases that come afterwards are performed by humans without any kind of assistance from the system. In fact, DSS is defined as the management of flows of data, transforming it into information and then into useful knowledge. The system automates the knowledge generation and the human makes the decision outside the system. As previously mentioned, ISO 900:2015 indicates that the decision must rely on evidence balanced with experience and intuition. Evidence comes partially from information and knowledge available in the DSS, but the intuition and the experience are exclusively in the manager’s head. Our approach proposes to extend the actual DSSs with a new component – the decision evaluation – to fill the gap in decision-making. This will enable increased ability to review, challenge and change opinions and decisions, as well as increased ability to demonstrate the effectiveness/ineffectiveness of past decisions through measuring their sustainability.

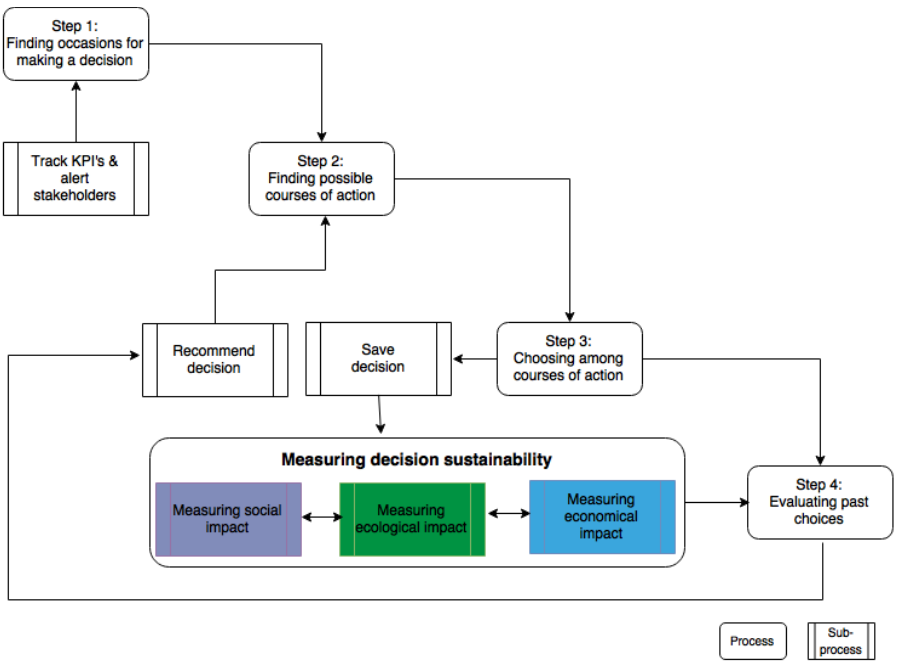

Figure 2.9 Green Decision-Making Process

Source: (Rezgui and Marx Gómez, 2016a)

All of this motivated the idea of treating the management of decisions over time and through all phases. Such management includes storing, evaluating sustainability and ranking decisions. These decisions will be stored in a central repository that serves as a core of the new green decision support system. The evaluation will rebuild the harvested knowledge in a way that is simple to use. Any company will see its decisions' sustainability evaluation in the form of a decision dashboard in which each single decision taken in the past can be seen with its reputation.

The proposed process – the green decision-making process (GDMP) – will be explained in this section. As shown in figure 2.9, our approach extends Simon's with pre-and post-activities to meet enable a decision-making based on evidence balanced with experience and intuition.

While monitoring the values of the different EPIs/KPIs, the user can find an occasion for making a decision if there is a deviation from the expected trajectory. Here, we can consider that the trigger of Simon's process (finding occasions for making a decision) occurred. In order to achieve the next phase (finding possible courses of actions), we need to rely on experience. This can only be possible if the decisions taken in the past are available and evaluated. The pre-activity – "recommend decision" – is responsible for making the previous decisions available and presenting their sustainability evaluation. Once the stakeholder has several options well described with their ecological, economic and social impact, he will be able to choose among them (step 3: choosing among courses of action). If none of the recommended actions is suitable for the issue that needs to be resolved, the user can create a new decision. In both cases, the decision will be stored. Based on the user responsibility, the decision can be saved directly without any validation or his/her manager reviewing and validating. The post-activity "save decision" includes saving all related information: the decision description, the decision-maker, the decision time and goals to achieve by applying this decision. Based on (Turban et al., 2011a), making decisions should occur to achieve one or more goals. A deadline should also be fixed for these goals to track their achievement degree.

The selected goals are arranged by domain (logistics, human resources, sales, production, etc.). For each domain, indicators are apportioned into three categories: economic, ecological and social. The goals can be set based on target values for one or many types of indicators. Regardless of the type of the selected indicators (economic, ecologic or social), they should be improved by the decision, while the others in the same domain will be tracked and evaluated (enhancement or weakening). Therefore, all pillars of the decision sustainability will be measured automatically. In order to transform this process into a sophisticated system, several researchers recommend a design of software architecture as a viable approach. "They guarantee the prosperity of software (systems) by defining sets of concepts as principles that guide analysis of specifications, designs, implementation, maintenance and evolution of software systems" (Kateule and Winter, 2016). After presenting the methodologies followed by the establishment of this work in the next chapter, the requirements definition, the specifications and the design will be presented in chapter 4.

3 Methodological Approach

The methodology followed by the establishment of this work is based on the design science in information systems research (DSISR) by (Hevner et al., 2004) and the design science research methodology (DSRM) by (Peffers et al., 2007). The next sections give more details about design science, the motivation of using this methodological approach and demonstrates how it was used to develop the decision evaluation system.

3.1 Design Science

The term “design" is defined in the Oxford dictionary, (i) as verb: “Decide upon the look and functioning of (a building, garment, or other object), by making a detailed drawing of it” and (ii) as noun “The arrangement of the features of an artefact, as produced from following a plan or drawing” (Oxford Dictionary, 2017). A deeper definition can be found by (Roozenburg and Eekels, 1995): “design is a goal-directed thinking process by which problems are analysed, objectives are defined and adjusted, proposals for solutions analysed, objectives are developed and the quality of those solutions is assessed”. Designs are then the instructions based on knowledge that turn things into value that people use. It embodies the instructions for making the things (Hevner and Chatterjee, 2015). (Vaishnavi and Kuechler, 2004) differentiate between innovative design which addresses the creation of new artifact requiring knowledge that does not exist and routine design requiring well known knowledge. They deliberate that innovative design may call for the conduct of research (design science research) to fill the knowledge gaps and result in research publication(s) or patent(s). Design is well used as central element in several disciplines including architecture, engineering, computer science, software engineering, media and art design and information systems (Hevner and Chatterjee, 2015).

A clear dissimilarity between two types of science: natural science and science of the artificial (design science) has been made by (Simon, 1996). While natural science is a body of knowledge about some class of things in the world (nature or society) that describes and explains how they behave and interact with each other. Otherwise design science is a body of knowledge about the design of artificial (man-made) objects and phenomena – artifacts - designed to meet certain desired goals (Vaishnavi and Kuechler, 2004).

In opposite to empirical science (empirical) with the aim to understand reality and concerned with general statements about nature or society, design science is concerned with the design of artificial constructs (concepts, designs) (Österle et al., 2010). Investigations using design as a research method or technique is called design science research (DSR). The output of a design science research project should be design science knowledge (Vaishnavi and Kuechler, 2004).

DSR is an accepted and wide spread research approach in the discipline of information systems (Cronholm and Göbel, 2014). It is emphasizing IT as the core of IS and is challenging the managerial and organizational issues that have been in focus within the IS discipline for many years (Orlikowski and Iacono, 2001). Table 3.1 illustrates the difference of researches in business information systems (BIS) between America and Europe.

	America	Europe
Nomenclature	Information Systems Research	Business Informatics
Focus	Explanation	Design
Result	Observation: - Properties of Information Systems - Behaviour of users	Artefacts: - Constructs, methods - Models, instantiations, prototypes
Sciences	Social Sciences: - Behaviourism - Positivism	Engineering: - Design Science Research - Constructivism
Strength	Scientific	Relevance for practice
Problem	Relevance for practice	Research methodology

Table 3.1 Business Information Systems Research

Source (Österle et al., 2010)

Based on the work of (Cronholm and Göbel, 2014) the purpose of DSR can be summarized in the following points:

- Guide design and evaluation of artifacts (Hevner et al., 2004; Sein et al., 2011).
- Support a dual mission: contribute to theory and assist in solving the practitioners problems (Baskerville and Myers, 2009; Benbasat and Zmud, 1999; Iivari, 2003; Rosemann and Vessey, 2008; Sein et al., 2011).
- Decrease the gap between responding to the need of practitioners and research rigor (Gallupe, 2007).

Furthermore design science, creates and evaluates IT artifacts intended to solve identified organizational problems (Hevner et al., 2004). Therefore, DSR can been principally seen as a problem-solving process. As a result of creating the DSR output – the IT artifact – knowledge and understanding of a design problem as well as its solution are acquired. Artifacts are implemented within an application context (e.g., a business organization) for the purpose of improving the effectiveness and efficiency of that context (Hevner and Chatterjee, 2015).

Based on the above we can deduce that DSR is the appropriate methodology for the establishment of this research work because of the following reasons:

- Focus is design: Guide design and evaluation of artifacts.

- Design output are IT artifacts.
- IT artifacts are implemented within an application context.
- Contribute to theory and assist in solving the practitioners´ problems.
- DSR is problem-solving oriented approach.

(Hevner et al., 2004) developed an information system research framework (ISRF) to encompass all design science research related concepts, define the relationship between them and present best practices in form of seven guidelines. Despite the helpfulness and expediency of this framework, it miss the ordering of the steps to follow while conducting a research project. Thus motivated (Peffers et al., 2007) to introduce the design science research methodology (DSRM). They presented a process containing six steps to follow for design science research works: identify problem, define objectives, design and development, demonstration, evaluation and communication. The research type by DSRM is defined based on the entry point. A research may be defined as problem-centred initiation (by entry in first step), objective-centred solution (by entry in second step), design and development-centred initiation (by entry in third step), and client/context initiation (by entry in fourth step). This diversity covers also the Action Design Research (ADR) process that integrates the ideas of action research with DSR (Sein et al., 2011). Both ADR and DSRM are process oriented. While ADR is organized in four distinct stages and each stage consists of activities that should be carried out, DSRM is divided into six steps with recommendations concerning what to do and objectives that should be achieved. (Cronholm and Göbel, 2014).

3.2 Information Systems Research Framework

The information systems research framework introduced by (Hevner et al., 2004, p. 79) argues that the declaration of (March and Smith, 1995) that developing and communicating knowledge concerning both the management of information technology and the use of information technology for managerial and organizational purposes involves two complementary but distinct paradigms, namely behavioural science and design science. It makes significant contributions by engaging the complementary research cycle between design science and behavioural science to address fundamental problems faced in the productive application of information technology (Hevner et al., 2004, p. 76)**.**

The following seven recommended guidelines were respected by the founding of this work:

Guideline 1: Design as an artifact

Two complementary approaches have been followed, namely behaviour and design science. While the former was used to develop and justify the theory of decisions evaluation, the latter was used to develop and build the artifact of the decision evaluation system (process, software architecture and prototype) to meet the identified business need.

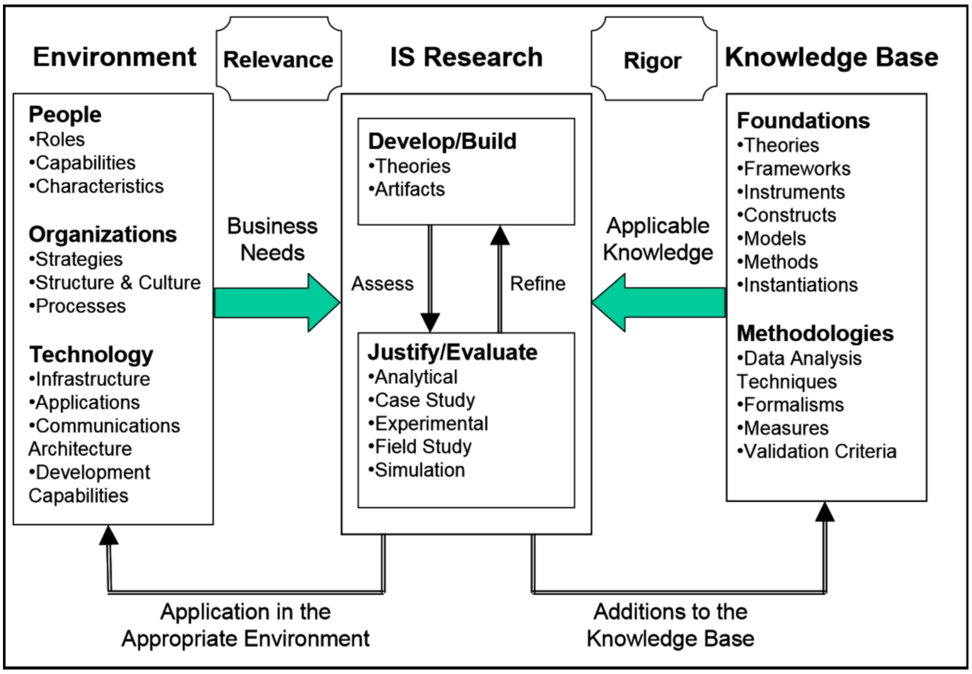

Figure 3.1 Information Systems Research Framework.

Source (Hevner et al., 2004)

Guideline 2: Problem Relevance

The environment defines the problem space (Herbert A. Simon, 1960)**.** People, (business) organizations and technology compose this environment (Silver et al., 1995) and play a major role in identifying the problem. The following figure illustrates the environment of this work.

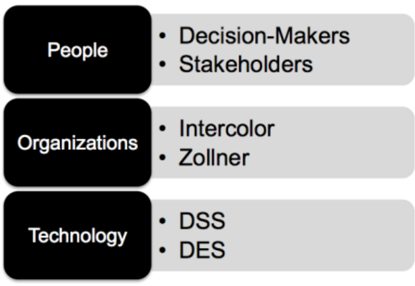

Figure 3.2 The problem space

The evaluation of decisions is addressed by framing research activities to assure research relevance (Hevner et al., 2004, p. 79)**.**

Guideline 3: Design Evaluation

The evaluation of the research results has been assured based on the justify/evaluate activity derived from the information systems research framework. The implementation of the decision evaluation system and its industrial and scientific feedbacks assures the design evaluation guideline (see Appendix A, B and C).

Guideline 4: Research Contributions

The research contributions have been assured through the integration and application of the decisions evaluation system in an appropriate industrial environment.

Guideline 5: Research Rigor

The systematic literature review in the research domain and the selection of ranked publications presented in this work played an essential role in obtaining applicable knowledge from the knowledge base. Accordingly, the research rigor was assured.

Guideline 6: Design as a search process

The iterative nature of the design science explained by (Hevner et al., 2004, p. 88) and described by (Simon, 1996) was necessary in the requirements definition, conception, design and implementation. A series of brainstorming sessions and feedback meetings after partial delivery with industrial partners and potential beneficiaries of the decision evaluation system was conducted in the form of a generate/test cycle. Thus, gave the possibility to meet guideline 6: design as a search process.

Guideline 7: Communication of research

Besides scientific publications in journals and conference proceedings, the results were presented to both technology- and management-oriented audiences. This gave the opportunity for practitioners to take advantage of the benefits offered by the artifact and it enables researchers to build a cumulative knowledge base for further extension and evaluation (Hevner et al., 2004, p. 90)**.**

3.3 Design Science Research Methodology

Furthermore, in addition to the DSIR, this research work was based on the design science in information systems research methodology proposed by (Peffers et al., 2007).

The DSRM process model comprises six activities ordered as follows: activity 1: problem identification and motivation; activity 2: define the objectives for a solution; activity 3: design and development; activity 4: demonstration; activity 5: evaluation; and activity 6: communication. The possible research entry points differ from one activity to another. Subsequently, "the process is structured in a nominally sequential order; however, there is no expectation that researches would always proceed in sequential order from activity 1 through activity 6. In reality, they may actually start at almost any step…" (Peffers et al., 2007, p. 56). In this work, the research was triggered by industry and called an objective-centred solution. While conducting several decision support projects, the need for a system that enables the evaluation of decisions was detected. Based on the recommendation of (Webster and Watson, 2002), in order to identify the problem and motivate the work, a systematic literature review was conducted. A vast number of research papers in the area of information systems related to decision evaluation has been conducted.

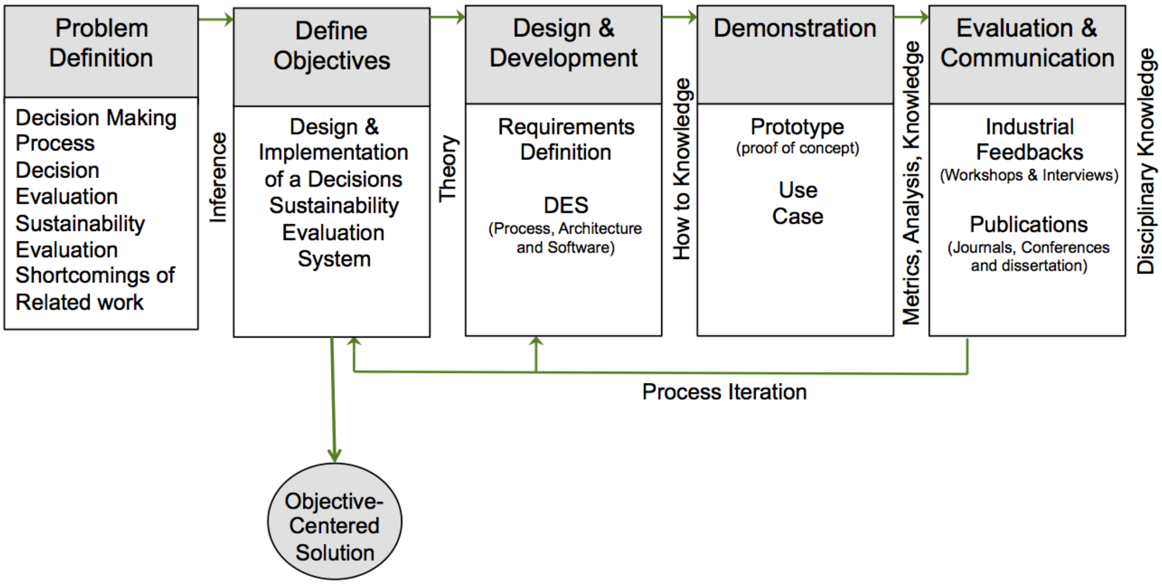

Figure 3.3 Design Science Research Methodology

Based on (Peffers et al., 2007)

The objective (in our case, the design and implementation of a decision evaluation system to enhance sustainability within organizations) was fixed. The functional and non-functional requirements definition was assured to enable the implementation of the global system process, the software architecture and the prototype. The demonstration was assured by a prototypical implementation (proof of concept) and a concrete business use case. The results obtained were evaluated by comparing them with the defined objectives. Additionally, interviews and workshops with experts in the domain were carried out (see Appendix A, B and C). The contributions of the work were disseminated in peer-reviewed scholarly publications in journals and conference proceedings (see section Publications). A process iteration based on the output of the results and feedback coming from the evaluation and communication activities allows the modification and adjustment of the objectives and the artifact.

4 Decision Evaluation System: Conception and Design

It is no longer an argument that the future of the world is not visible - if not to “harshly” say “threatened” - on different levels. The prosperity of industry and the ubiquity and ominous presence of information and communications technologies (ICT) in our lives have come at the price of endangering the eco-systems of the planet (e.g., global warming caused by emissions, utilizations of non-renewable nature resources, etc.). However, the environment and ecosystems are not the only problems threating the future, and economic instability in developed and developing countries, but the rise of civil and international conflicts too.

Working towards avoiding and preventing these problems from different dimensions (threats to the environment, economic instability and uncertainty, violence and social conflicts) in present times and the future is often called Sustainability Development (SD).

One of the most cited definitions of Sustainability Development is given by (Brundtland, 1987) where it is stated that: “... development that meets the needs of the present without compromising the ability of future generations to meet their own needs.” This definition, also known as the “Brundtland definition,” combines two ethical claims (Hilty and Hercheui, 2010):

- Intra-generational justice (meeting the needs of the present) and
- Inter-generational justice (not compromising the ability of future generations to meet their own needs).

Going towards Sustainable Development is a responsibility for both public and private sectors. Environmental Information Systems (EVIS) is used in the public sector for spreading public awareness on sustainability, political decision support and executing environmental policies. In the private sector, such systems help private organizations with legal compliance with environmental policies, environmental reporting to stakeholders and Eco-efficiency (Hilty et al., 2005). The focus in this work is on business companies and private organizations.

Sustainability management is finding its way in managers’ agendas in decision-making (carbon emissions, renewable energy consumption, social engagement, etc.). There are even some efforts to establish full independent department or division providing audit and assessments for sustainability factors and indicators in a company’s processes and activities.

Business Process Management (BPM), which refers to a body of methods, techniques, and tools to discover, analyse, redesign, execute and monitor business processes is a crucial starting point of an organization to introduce sustainable improvements and technologies (Couckuyt et al., 2017). Since its introduction, the objective of conventional BPM is typically seeking improvements in purely economic imperatives such as time, cost, quality, and flexibility or the so-called “devil’s quadrangle.”

Since sustainability also gained importance in the field of BPM, it can be considered as a fifth process performance dimension (Marx Gomez et al. 2011; Couckuyt, Van Looy, and De Backer 2017).

However, business companies, when adapting SD or "Going green," could find their conventional operations fundamentally challenged. Processes and products need to be re-invented, controlling systems have to integrate new sets of data, external and internal communication strategies require revisions, and fundamental values and knowledge systems need to adapt (Siebenhüner and Arnold, 2007).

(Siebenhüner and Arnold, 2007) emphasized on organizational learning on the sustainability subject by gathering knowledge and designing learning process. (Meisch et al., 2015; Müller-Christ, 2011; Müller-Christ et al., 2014) focused on the contribution of universities in SD and their crucial role, this supports the theory that SD needs learning processes and behavioural changes by moving the efforts of sustainability to academic and complex institutions such as universities. They developed model-reusable systems for Sustainability Indicators assessments to higher education facilities (Sustainability Check 2.0, 3.0).

Interdisciplinary relations between the Information Society and sustainable development began to be thoroughly investigated academically in the mid-2000's, this is on the international level and mostly in European countries (Hilty et al., 2005). The goal is to see how much (ICT) can contribute to SD? Can a part of the problem (ICT ubiquity and influence on globalization) become a part of the solution (Going towards sustainable uses and practices)?

The next sections discuss the relationship between ICT and SD then define the functional requirements of the Decision Evaluation System.

4.1 Global System Functional Requirements

According to (Hilty and Aebischer, 2015a) and various research (Hilty and Aebischer, 2015b), ICT can indeed use its transformational powers to affect production and consumption patterns positively. But this will not come without a price because ICT will need enabling products and infrastructures with life cycles and potential harm to ecosystems.

For instance, "dematerialization" or "Immaterialization" is a significant discipline that can be granted by ICT. It consists of replacing a material product with an immaterial service while achieving the same objective (Hilty and Aebischer, 2015b) for instance:

- Electronic media replacing print media (e.g., books, images).
- Telecommunications replacing some travel activities (e.g., video conferencing).

Those examples prove that the substitution potential of ICT is effective in decreasing or removing products’ life cycles and heavily decreasing the carbon emissions of transportation activities.

ICT also has a potential for optimization, consumption and production patterns can be enhanced for example by allowing automated governance and assessment on predefined sustainability indicators via for instance Environmental DSS, Internet Sustainability Reporting…etc. (Isenmann et al., 2005). Companies can benefit tremendously from Artificial Intelligence applied to optimize business processes to be more sustainable and cost-effective.

As for potential adverse impacts, like mentioned above: The price of using ICT is the direct effects of production, use, and disposal of machines and infrastructures (Hilty and Aebischer, 2015a). In addition to this, we have two effects that can be enabled by ICT (Hilty and Aebischer, 2015a):

- Induction effect: ICT stimulates the consumption of another resource (e.g., a printer stimulates the consumption of paper as it uses it faster than a typewriter).
- Obsolescence effect: ICT can shorten the useful life of another resource due to incompatibility (a device that is no longer supported by software updates is made obsolete).

The following figure presents an illustrative schematization of how ICT can be a part of the problem and the solution for sustainability on three levels:

- **Direct effect**: The life cycle of the technology products.
- **Enabling effects**: The effects of what could be enabled with the application of ICT.
- **Systematic effects**: Refers to long-term effects of ICT including behavioural change (lifestyles) and economic structural change on the positive side. On the contrary, we have the rebound effect which means that ICT consumes energy, thus, produce emissions in a quantity which compensates (or overcompensate) the positive dematerialization and substitution effects.

(Loos et al., 2011) consider that process change conditions the role of enabler for "green" enterprise which can be played by IT systems. Business and IT managers need to cooperate and engage in a process-focused discussion to enable a common, comprehensive understanding of processes, and the opportunities for making these processes, and ultimately understanding the organization as a process-centric entity, “green.” Their primary approach in (Loos et al., 2011) is that green IT by itself does not lead systematically to green business. The transformative power of IT can be applied if: Green IT supports process change which enables green business.

Alongside with Business Process enhancement by the cooperation of business and IT specialists and strategic thinking, well-studied and responsible decision-making in different circumstances is also necessary to achieve sustainability. Decision-making is defined as the process of choosing an option from two to many available alternatives, the purpose of making decisions is achieving one or more goals (Turban et al., 2011b).

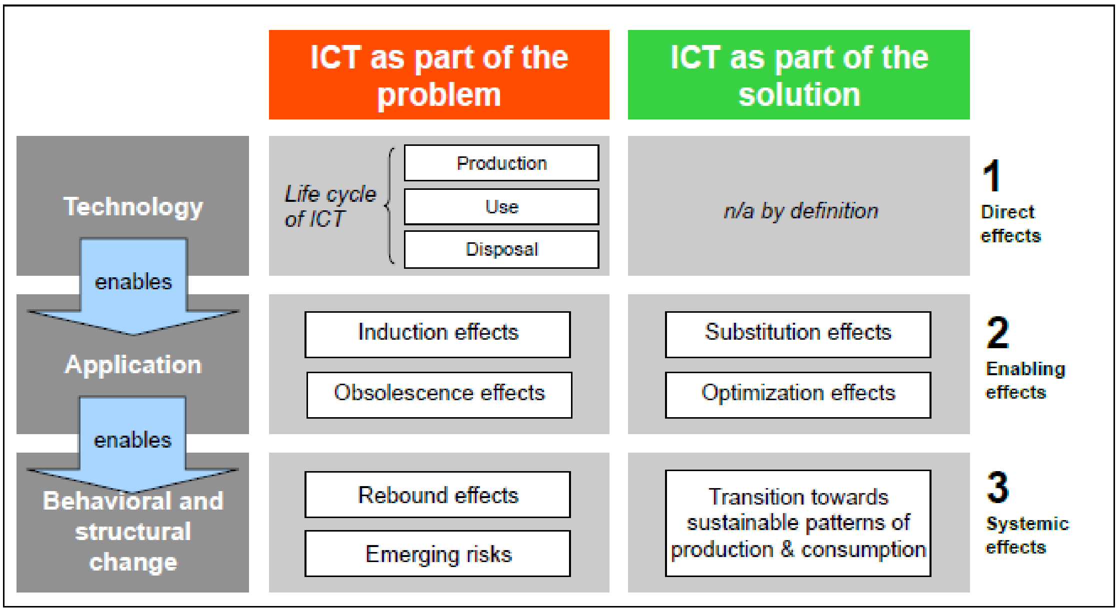

Figure 4.1 Matrix of ICT effects

Source (Hilty, 2011)

There are many Information Systems (IS) that have been developed to support managers in decision-making to choose the more sustainable decision among multiple alternatives. Those IS continue to progress by using newer technology with the same aim of SD. For instance, (Isenmann et al., 2005) and (Herzig and Godemann, 2010) along with many other works presented tools for providing internet-based, tailored and on the fly sustainability reports. Company managers can use such reports for disclosing sustainability issues. The issues often include the following contents: (1) vision and strategy, (2) profile of organization, (3) governance, (4) policy, (5) management systems, (6) stakeholder relationships, (7) and environmental, social, economic, and integrated performance indicators (Global Reporting Initiative, 2002). This disclosure and understanding of the current sustainability state and performance of the company help sustainable decision-making.

The variety of range for managerial decisions that could be taken in an organization can raise concerns of how the DES will perform. Decisions are broadly classified in three main classes (Oboni Riskope Associates, 2014; Smriti, 2017):

- Operational decisions, related to day-to-day actions and routine practices or responding to events and have short term impact.
- Tactical decisions, also have a short term impact related to planning in specific divisions of the organization, structuring workflows, acquisition or dropping resources (e.g. material, money, men).

- Strategic decisions, they concern long term plans or directions impacting the whole or major part of the organization, they may involve major departures from practices and procedures currently being followed and resources are being allocated progressively to pursue the long term goal.

Operational and tactical decisions are usually taken by lower and medium levels that are familiar to details and conditions of the organization environment. Strategic decisions are taken by higher management with a wider vision of the organization’s status and direction.

Concerning the DES, strategic decisions are too specific (ad-hoc) and wide-ranging to be considered as a reusable case. A manager cannot (even wouldn’t) use such an automated AI technique to recommend complete strategies and long-term plans that would directly affect the fate of the organization. The DES only handles tactical (or partially tactical) and operational decisions that are related to events, dynamic requirements and policies in the organization’s different domains and sub-domains: Many of those decisions can indeed be used in a shared knowledge-base as retainable and reusable cases because they can be repetitive and short termed.

The cumulative impact of operational and tactical decisions has a global effect on the organization. Thus, they affect future strategic decisions that are executed on the global scale of a company. This relation could also be vice-versa where general strategies and policies affect simpler and everyday actions, it could be described as cyclic relationship or continuum (Srinivas, 2015).

The aim of this work is the design and implementation of an ICT tool called decision evaluation system to enhance sustainability within organizations. It should support individuals in organizations to make more sustainable decisions.

It should enable the detailed tracking of the impact and the evaluation of a “hypothetically” sustainable decision. DES also should offer a case-based reasoning recommendation of potentially sustainable decisions. This support can be guaranteed through assuring the followings functional requirements (FR) listed in table 4.1. They present what the system should be able to accomplish (Lightsey, 2001):

FR1 - Allowing participation in the decision-making process for two categories of users: The two categories of users are: (i) Individuals and (ii) groups like decision committees (A group of committee members that agree on ordering and suggesting decision after discussions, analysis and voting (Black, 1948)).

Those users responsible for inserting decisions to be evaluated by the DES will require a solid background in sustainability practices and intensive knowledge of the three classes of sustainability indicators. The classes are Social Performance Indicators (SPI), the Ecological Performance Indicators (ELPI) and the Economical Performance Indicators (ENPI). The users will also need to understand the relationships of these indicators to the company's activities (Manufacturing, Sales, HR management, Finance, etc.).

It is preferable that the users are a specialized, organized and collaborative group or department in the business company that can be called "Sustainability management." This will allow having a complete, studied and well-described catalogue of sustainability performance indicators that is relative to the business activities and the operational and strategic decisions.

ID	Requirement Description
FR1	Allowing the participation in decision-making process for two categories of users
FR2	Organizing access and privileges
FR3	Enabling users to enter/configure the system parameters
FR4	Enabling insertion of decisions
FR5	Enabling decision validation
FR6	Tracking the decisions
FR7	Evaluating decisions
FR8	Enabling comments and user evaluation on decisions
FR9	Recommending decisions
FR10	Having transparency in the decision-making process
FR11	Being a sustainability-friendly (green) software

Table 4.1 Functional Requirements

FR2 - Organizing access and privileges: Granting access to the system with limited or full privileges to an editable list of users (individuals, committees). This list is managed by system administrators (IT background) that do not take part in the decision-making in the DES. By limited privileges, it is meant that a user with a role of decision-maker can create decisions only, without the right to validate them. Another user with full privilege decision -maker and validator can validate his own created decisions or other decision inserted by other individuals or committees. Expertise and skills in the sustainability field and general management experience can give privileges.

<u>FR3 - Enabling users to enter/configure the system parameters</u>: Those parameters are the key performance indicators that the organization is currently tracking for assessing its sustainability performance. Those KPIs are classified by sustainability specialists according to:

- Their sustainability nature:

  Whether they are considered social, economic and ecological performance indicators. In the literature, those fields are called sustainability topics or pillars, they may be interconnected, and it is widely accepted that their balance is the method to reach sustainability (Hilty et al., 2005).

- Their link to the company's activities or departments:

  A KPI can be related to a single domain of activities that can be (Human Resources, Sales, Logistics, etc.) and a single sub-domain that can be (HR Training, After-sales, Transport…).

- Domains and sub-domains are also configurable, and they are also considered as system parameters.

<u>FR4 - Enabling insertion of decisions</u>: To track, recommend and evaluate decisions, the DES must first enable decision-makers to enter their decisions. If none of the displayed recommended decisions is suitable to select (or no recommended decisions are available), the decision-maker creates a new decision: This includes entering the parameters:

- Decision title.
- Detailed description of the decision occasion (the problem or the opportunity that invoked the making of the decision).
- Detailed description of the proposed solution.
- Decision-executor and its contact information. This actor will be responsible for executing the decision on the ground. An excellent and fast coordination between making and executing is a key to produce better performance for an organization (Marcia W. Blenko et al., 2010).
- Decision objectives: In the DES, it is assumed that a given decision in a given sector of enterprise activities (After-sales, Finance, Healthcare, Warehousing, etc.) has a set of objectives to accomplish. Those objectives are enhancing the KPIs from the same sector of activity (After-sales, finance, etc.). This desired positive impact for a decision on a KPI is set by:

    - A deadline: A target date assigned to when the action must be completed by. Here the action represents the change (increase or decrease) of the KPI included in the decision.

- A target value: A numerical value of the KPI which should be reached by the deadline mentioned above. Reaching this value or not is a factor for the evaluation of the decision.

FR5 - Enabling decision validation: As mentioned in the second requirement, not all decisions are automatically validated if inserted by a user. Superior users "Decision-validators" will have the role of re-studying the proposed decision and then validating or rejecting the decision while mentioning the reason for rejection (For transparency and professional reasons). The following figure shows a BPMN flowchart diagram depicting the steps of decision-making and validation in the DES. The decision-maker (a user of the DES) is an agent from the sustainability management (department) inserts the decision parameters to the system with its objectives and deadlines and then validates it if given the privilege. If he does not have the privilege of decision validation, another actor "Decision-validator" becomes a part of the process with the task of reviewing the decision. After analysis and studying, he either validates the suggested decision or rejects it along with mentioning the motives for rejection.

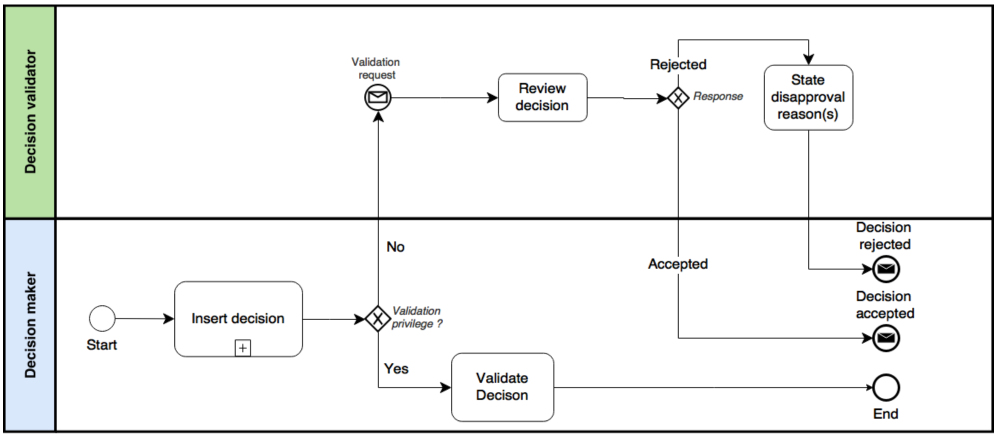

Figure 4.2 Decision-making and validation steps in DES

FR6 - Tracking decisions: This happens after a decision was created in the system and set as executed. That is another necessary operation in DES after validating a decision. An excellent and fast coordination between making and executing is key to produce better performance for an organization (Marcia W Blenko et al., 2010). When the decision is affected by a responsible executor and is executed "on real ground," it can be set as "executed" in the DES while storing its execution date. Only then it will be tracked for impact. Tracking the decision means observing its direct and indirect impact on KPIs values (enhancement, deterioration, no affection) from its execution date to the designated objective deadlines for each objective KPI. Direct impact refers to the decision's impact on KPIs that were set as its objectives. Indirect impact refers to the decision's impact on KPI that were not set as its objectives, but they belong to the same decision's sub-domain (After-sales, Finance, Healthcare, Warehousing…).

FR7 - Evaluating decisions: Based on the decisions' average impact nature (positive; negative), intensity (low, medium, or high impact) and their accomplishment of objectives, the DES should be able to attribute a final rating (numerical) and a sustainability evaluation for each decision (unsustainable, medium sustainable, or very sustainable). This will be available only after all the deadlines (target dates of the objectives) for the decision are reached. This concerns the fourth step of "evaluating past choices" from the decision-making model set by (Simon, 1977). The evaluated decisions can be displayed to users with their final calculated rating and sustainability evaluation along with its detailed impact on each affected KPI. This will be displayed in the impact dashboard shows the following information for each KPI:

- Sustainability nature of the KPI: whether the affected KPI is classified as an economic, ecologic or social performance indicator. This will help the observation of further details about the decision performance in each field of sustainability (e.g., a decision has an average final rating, a good impact on social indicators, and an adverse impact on ecology indicators, etc.).
- Name, measure unit and desired variation of the KPI.
- If the affected KPI was set as an objective. The target value and the deadline (target date) set when creating the decision is displayed. This not the case for KPIs that were not set as objectives.
- The actual 2 values of the KPI: (1) Most recent value of the KPI before the decision execution date and (2) the most recent value of the KPI before or equal to the deadline date.
- Percentage of the impact on the KPI (e.g., 30% increase, 10% decrease) and the nature of the impact (enhancement, deterioration, no impact).

FR8 - Enabling comments and user evaluation on decisions: Knowledge, in general, is considered the vital resource for organizations. Therefore, knowledge management (creation, sharing, usage) is considered highly essential for organizational success (Ipe, 2003; Stewart and Ruckdeschel, 1998). Unstructured knowledge sharing also plays a vital role in enhancing the knowledge management for an organization, especially when a significant amount of unstructured data is being exploited every day in organizations (Grimes, 2014) (e.g., reading e-mails and articles, viewing pictures and diagrams, listening to recorded audio tracks, etc.). According to a study by the Data Warehousing Institute (TDWI) in 2007, 51% of data across an organization is semi-structured or unstructured (Russom, 2007). Older and widely-cited studies of analysts from Merill Lynch, IBM, Granter, and many others state that 80-85% of the business data is unstructured.

In addition to the automatic decision evaluation by the DES, users may need to add textual comments to decisions to further understand the outcomes of the decisions from their perspective as stakeholders and close observers of the impact. The comments should be represented in textual formats next to the timestamp and their authors. Other files may be attached to a comment like reports, images, audio files, videos, etc. This is called unstructured knowledge sharing (Baars and Kemper, 2008a; Blumberg and Atre, 2003a; Rao, 2003a). A user can also evaluate the decision and give it a rating on a scale of 1 to 10. This scale is larger than the scale used in the automatic decision evaluation (1 to 5) because it will be used by a person (decision-makers). The researchers in Primary Intelligence claim that this 10-point scale provides better variability and differentiation to users and have worked best form them in their 13 years' experience (Intelligence, 2017).

FR9 - Recommending decisions: The primary purpose of recommender systems is generating suggestions about resources that a user is not aware of a priori but would probably be interested in according to (Jøsang et al., 2013a). In the proposed DES, previous and context-related decisions that are already evaluated will be recommended for the decision-maker to help him with the decision-making process. The system must be able to recommend decisions after a user describes the context of the decision (Domain, Sub-domain, occasion description, objectives description, scopes). Recommended decisions should be displayed with their previous impact and ranked by their sustainability evaluation and their similarity percentage with the current case. The decision-maker can select a recommended decision and "adapt" it to the current situation by setting new objectives or slightly editing the problem or solution description. This concerns the second step "finding possible courses of actions" from the decision-making steps set by (Simon, 1977). The same process of decision validation is executed after the selection of a recommended decision by a user with or without validation privilege (see figure 4.2).

FR10 - Having transparency in the decision-making process: Transparency, as defined by (Andrew K. Schnackenberg and Tomlinson, 2016) is: "The perceived quality of intentionally shared information from a sender." Transparency is highly crucial within an organization; it is seen as a remedy for distressed relationships between the organization and its stakeholders through its presumed ability to re-establish the trust between both sides (Andrew K. Schnackenberg and Tomlinson, 2016). In the decision-making process within the proposed DES, transparency should be granted by identifying the decision-maker, the decision-validator and decision-executor for each decision. The dates for the creation, validation, and execution for each decision should be stored for each decision. This information (or Metadata) has to be displayed whenever the decision is required to be displayed in any Graphical User Interface (GUI).

FR11 - Being a sustainability-friendly (green) software: Sustainability intersects Information and Communication Technology in two domains: Green IT (how can we make ICT itself more sustainable?) and Green by IT (how can we achieve sustainability through ICT?) (Naumann et al., 2015). In addition to evaluating and enhancing the sustainability performance of the company's activities, the DES itself as software (shared network application) need to be "green" and promotes SD for software engineering (green software engineering).

At first glance, compared to heavy industrial, health policy and transportation activities for an organization, the impact on SD of software installed on ICT machines seems negligible. It has less of an impact but is not something negligible and unimportant.

All kinds of software (web-based, desktop, mobile, etc.) induce the use of ICT because it "makes life easier." The ubiquity of information and communication technologies results in substantial amounts of energy consumption and thus, CO^2 emissions. According to (Hintemann and Clausen, 2016), who give an overview analysing different international studies addressing the topic, "IT energy consumption […] accounts for approx. 8% of global electricity consumption" (Guldner et al., 2018). The ICT hardware life cycle (usage; disposal) also depends on how software uses its resources. The wise, sustainable practice, in this case, is developing software with the efficient use of resources, thus extending the life of hardware. So, the proposed DES is required to:

- **FR11.1 - Operate with efficiency:** Minimal consumption of the machine's resources (RAM utilization, disk space, CPU utilization) and minimal data flow from/to data sources. This will reduce the energy consumption of the hosting machine and the database servers.
    - ✓ Ecological impact
    - ✓ Economic impact
- **FR11.2 - Not to induce printing and paper consumption:** Generally, ICT induces paper consumption because new printing technologies are faster to print hundreds of paper by just a few clicks (faster than older technologies like a typewriter) (Hilty and Aebischer, 2015b). Some valuable information provided by the DES may be necessary to be printed on paper occasionally. The impact can be minimized if the printable' templates use as little ink as possible and users are restricted from a significant number of prints from the DES (e.g., 60 printed papers as a maximum for a single user in a month).
    - ✓ Ecological impact
    - ✓ Economic impact
    - ✓ Social impact
- **FR11.3 - Enable organizational learning on sustainability:** This is a long-term objective we aspire to accomplish with DES. Organizational learning is considered a necessary step to SD (Siebenhüner and Arnold, 2007). Not only we used ICT to improve SD for the company's activities, but the more users (Sustainability specialists and managers) interact with the DES, the more they learn about:

- Different types and measures of sustainability indicators (knowledge sharing among sustainability experts, beginners, and other business users).
- Impact of different decisions on sustainability indicators (investigating previous cases of decisions and their impact). This is like operating Case-Based Reasoning not only on the system to recommend decisions but also on the minds of sustainability specialists.
- The different factors that other users from different backgrounds consider in order to evaluate a decision. This is done by observing comments and feedback from users on evaluated decisions.

The list is not exhaustive. It could be extended. Organizational learning is a long-term, non-guaranteed objective of DES that depends heavily on employee behaviour and company initiatives and efforts.

- ✓ Social impact

4.2 Global System Non-Functional Requirements

The non-functional requirements present the quality attributes (performance, usability, etc.) of the system and its expected behaviour (Chung and do Prado Leite, 2009). The DES should have the following qualities:

Modularity: The overall system comprises different functional components that collaborate to provide the system's global functionality. Each of the system functionalities (tuning parameters, tracking, evaluating, recommending decisions) are only permitted by executing smaller, simpler sub-functionalities that need various or common resources. A modular software design is needed to enable different functional components to interact, connect and share resources. It can even provide/acquire components to/from other modular software. This is a mandatory requirement.

Performance: The execution time for the system functionalities should be minimal. This includes minimizing and optimizing the read/write operations from databases in addition to time complexity of the algorithms used in the system. This is an optional requirement for the DES first prototypes.

Efficiency: Minimal consumption of the machine's resources (RAM, disk space, processing capacity) is recommended for the system utilization. This is an optional requirement for the DES first prototypes.

Scalability: The system and its database need to cope with the changes that may accompany the stakeholders' requirements, goals and visions. System parameters (KPIs, domains, sub-domains) may be added or deleted whenever the decision-maker requires. The number of decisions to be tracked, evaluated and recommended also varies depending on the circumstances in the organization and the identification of occasions for decision-making. This is a mandatory requirement.

Reliability: This requirement – and fault tolerance in general – is achievable by anticipating exceptional conditions and building the system to adapt with them (Divya and Premalatha, 2016). The system needs to be reliable and robust against different errors as much as possible. Errors include failed data read/write from databases, occurring exceptions of the system operations and functions and erroneous input from the users. Error notifications need to be meaningful and unambiguous. This is a mandatory requirement.

Correctness: Data correctness is a crucial requirement for all of the DES functionalities. The displayed data values must conform to the database's updated values (with correct date stamps for the DWH retrieval). The decision evaluation process also needs correct, updated KPI values to generate an accurate evaluation for decisions. The user input and the automatic decision evaluation also should be correctly stored in the databases.

Security: An authorized access to the system users needs to be assured. A typical authentication process where the user inputs his login ID and password is required before using the system. Further security procedures can be executed for important commands that emphasize more authorized access like the decision creation, selection and validation. This is a mandatory requirement.

Accuracy: Some of the user's inputs like the titles and descriptions for entered KPIs, decisions, domains and sub-domains must be spell-checked in the organization's official language. A clear, vocabulary checked definition for the problems, objectives and solution will help significantly with the case retrieval in case-based reasoning (CBR) while recommending decisions. The deadline dates for the decision objectives must always be superior with the current date of the decision creation or selection. This is a mandatory requirement for the deadline dates assignment and partially optional for the vocabulary checking of user input.

Compatibility: The system's graphical user interfaces (GUIs) must be compatible with a wide range of recent web browsers (IE, Chrome, Firefox, Safari, etc.). For the unstructured data sharing in the decision's comment section, the system must enable uploading various forms of videos (mp4, avi, wmv, etc.), audio (mp3, wma, etc.) and other document files (xlsx, doc, pptx, etc.). This is an optional requirement.

User-friendliness: All of the GUIs must be presentable, simple and understandable for novice and non-IT users. This is a mandatory requirement.

Usability: It is a mandatory requirement. A coherent design and an unambiguous expression for commands and other graphical elements is required. Users need to interact with the system functionalities with a minimal effort and time.

Table 4.1 summarizes the aforementioned optional (O), partially optional (PO) and mandatory (M) requirements and presents how the DES is expected to satisfy these requirements.

Requirement	Solution in DES
Performance (O)	Minimizing temporal complexity of the algorithms executed by the system by optimizing their codes and scripts.
	Optimizing the graphical elements in the UIs to load in minimal time and refrain from using unnecessary ones.
	Minimizing read/write operations from the Databases especially for large amounts of data or distant databases.
Efficiency (O)	Minimizing the spatial complexity of the algorithms executed by the system by optimizing their codes and scripts.
Compatibility (O)	Designing the web-interfaces with a UI component library (framework) that is compatible with moderate web browsers like Internet Explorer.
	Integrating only graphical elements that are supported by both moderate and powerful web browsers.
Accuracy (PO)	Setting the minimum value to superior to the current date for objectives' deadlines for decisions.
	Applying vocabulary checking for text fields that will contain the user's description of the occasion (problem), the solution (decision) and the objectives.
	Restricting values for scope(s) definition (e.g. "Diversity", "quality of life", "Education"... for Social scopes / "Resource management", "pollution prevention"... for Ecological scopes).

Requirement	Solution in DES
Modularity (M)	The system comprises four functional components responsible for executing the different operations, they are sharing resources (Database & DWH) and interacting with the front-end component.
Scalability (M)	Using a recent Database Management System (DBMS) for database manipulation. There are numerous features that must be within a DBMS in order to be judged as scalable. Those features include distributed design of the database, DB replication, allowing simultaneous access and minimizing shared resources (Douglas, 2012).
Reliability (M)	Managing the exceptions for failures of read/write operations from databases with Java's exception handling methods (try, catch, and block).
	Controlling the user's input and prevent form-submitting without required fields.
	Using data access objects (DAOs) to provide an abstract data retrieval instead of direct data access, the DAO works as an adapter between the data source and the system and it can adapt to different storage schemes and the changes of the data source (Deepak et al., 2001).
Correctness (M)	Generating accurate SQL queries for storing the users' input (KPIs, domains, sub-domains, decisions, and user’s parameters). These queries are automatically written and executed by the DAO.
	Obtaining a correct ETL mapping between the DWH and the database from the system administrators or other IT-experienced users.
	Generating accurate SQL queries for data retrieval. These queries are automatically written and executed by the DAO.
Security (M)	Requiring authentication parameters (login, password) for each access to the system's different features.
	Denying any unauthorized access by redirecting unauthorized users to an “Access-denied” page. This should be done by applying web (or connection) filters.

Requirement	Solution in DES
User-friendliness (M)	Designing the web-interfaces with a UI component library (framework) that provides nice, ergonomic and clear graphical elements (dashboards, charts, menus, notifications, etc.).
	Displaying simple, short and understandable texts for guidelines and notifications to users.
Usability (M)	Following the most common ergonomic guidelines and recommendation for UI design (e.g. CUergo[3], Usability.gov[4], Microsoft IU design guidelines[5]).
	Providing usability tests on the system's first prototypes by different users (non-IT-experienced) and analysing results and feedback about the user experience with the overall system.

Table 4.2 Non Functional Requirements

4.3 Global Solution Modelling

The global solution modelling aims obtaining a valid version of the DES. The solution finally chosen is not obtained in a unique iteration. Several steps were necessary; these successive steps make it possible to refine the level of details of the system to be realized. This section illustrates the global process interactions and the data structure.

4.3.1 Global Process Interactions

The flowchart diagram is a simple, understandable graphic representation of the structure and steps for an algorithm, process or system from the start to the end (Damij, 2007). Cross-functional flowcharts (or deployment flowcharts) further add a representation of the responsibility of the different actors in the process or system and the interactions between them (Scholtes, 1997).

The global process of the DES will be described in the following cross-functional flowchart diagram. This flowchart is modelled with the norms and standards of version 2.0 Business Process Model and Notation (BPMN 2.0) maintained by the Object Management Group (OMG) [6] in addition to the UML.

[3]http://ergo.human.cornell.edu/ahtutorials/interface.html
[4]https://www.usability.gov/what-and-why/user-interface-design.html
[5]https://msdn.microsoft.com/en-us/library/jj651618(v=nav.80).aspx
[6]A non-profit technology standards consortium. http://www.omg.org/

The primary goal of BPMN is to provide a notation that is understandable by all business users, namely business analysts, technical developers and users (Object Management Group, 2011).

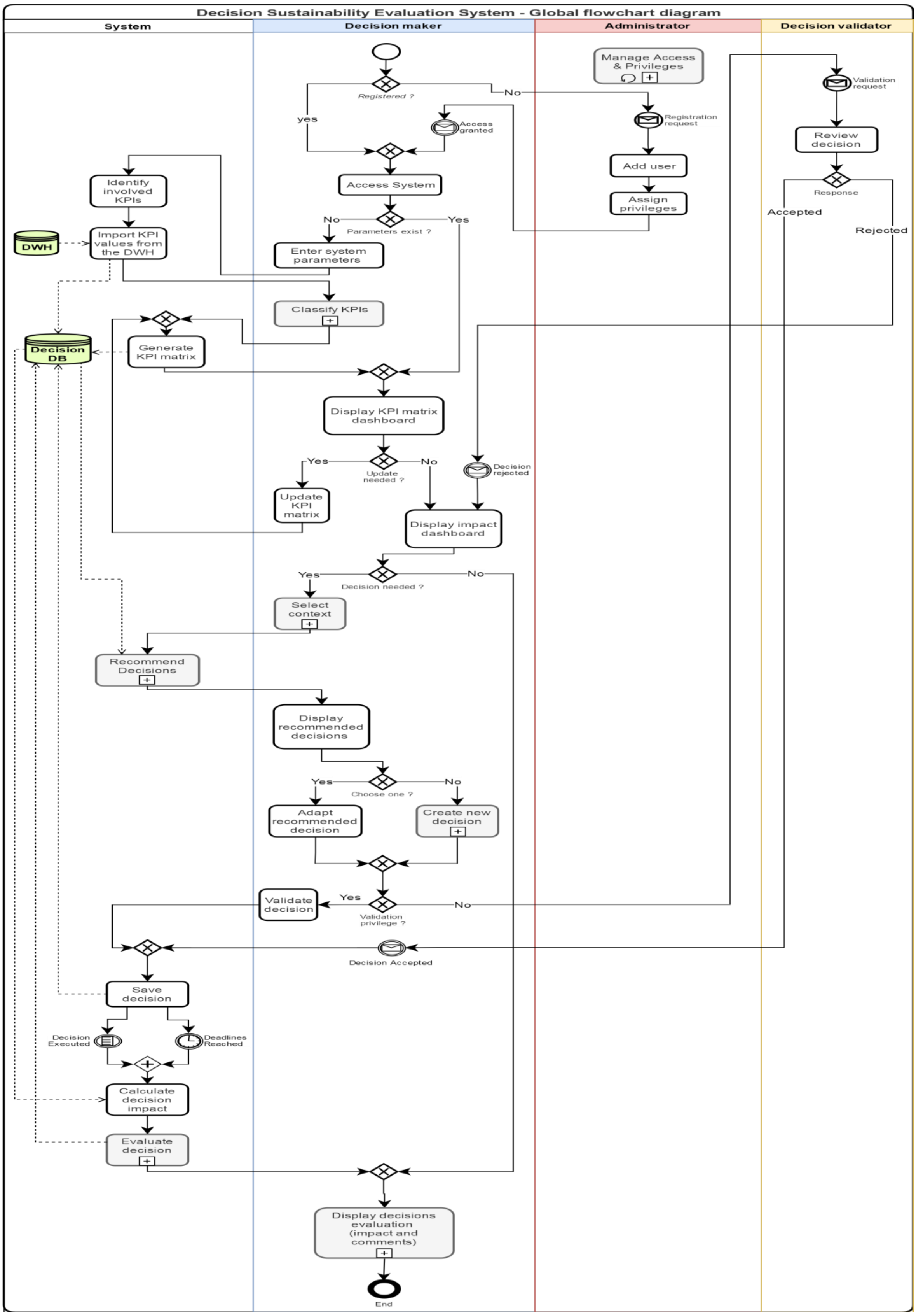

Figure 4.3 DES Flowchart diagram

The flowchart diagram for the DES illustrated in figure 4.3 represents the following scenario that includes all of the available functionalities provided by the DES:

A decision-maker can start using the DES after he has been registered and given privileges by an administrator. After a successful authentication, the decision-maker must first enter the system parameters (KPIs, domains, sub-domains), if these are not already present. The system identifies the involved KPI values and the ETL mappings constructed by technical users will help importing their values from the DWH and saving them in the decision database. The system identifies the involved KPI values (with the help of ETL mapping by data integrators) and imports their values from the DWH. These system parameters can be configured later.

The decision-maker also classifies the KPIs by domains/sub-domains and sustainability natures and the system will later store this classification in the decision database.

The KPI matrix that the system generated based on the decision-maker's classification will be displayed in dashboard, whereby the user can learn more about the KPIs while viewing this matrix and – if necessary – he can update the classification. Another updated matrix is then re-generated by the system. These aforementioned steps of configuring the system parameters, KPI classification and matrix generation is the task of the KPI & Decision Tuner component.

If no update is needed for the KPI matrix or the system parameters, the decision-maker can continue to analyse indicators in the impact dashboards or check for alerts from BI systems about the status of KPIs. If no occasion for making a decision is identified, decision-makers can display the impact and sustainability evaluation of previous decisions. These previous decisions were tracked and evaluated by the system after the deadline dates for their objectives were reached. Decision-makers can also view other users' comments and evaluations about decisions and add their own comments and evaluations.

The decision-maker identifies an occasion for making a decision by analysing indicators or receiving an alert from a BI system about KPI(s) that dropped below threshold(s). Before making a decision, the user must select the context of the decision (domains/sub-domains, problem, objectives, and scopes) to check for recommended previous decisions. Using these context parameters, the system – via the CBR mechanism – can check in the database for similar experienced cases to recommend their solutions (decisions). All of this is included in the "recommend decisions" sub-process executed within the decision recommender component. The recommended decisions are displayed to the decision-maker, who can choose to select and adapt a recommended decision or create a new decision. "Create a new decision" is also a sub-process comprising smaller processes and executed within the decision engine component.

Adapting recommended decisions or creating a new decision needs validation before saving the decision in the system for eventual evaluation. If the decision-maker has the validation privilege, he can validate his own decision. If this is not the case, another actor – called the decision-validator – will review the decision and choose to validate or reject the decision. The decision-maker will be notified about the feedback of the decision-validator.

The validated decisions are only ready for the evaluation process after two conditions are satisfied: when a decision is executed and its deadline dates have been reached, the system can evaluate it based on its direct and indirect impact on KPIs (average enhancement, average deterioration). This is the task of the decision evaluator component. The evaluated decision becomes available to show its impact dashboards on the different types of KPIs (ELPI, ENPI, and SPI). The decision-maker can choose to show other users' comments and ratings on the decision or add his own comments and a rating for the decision on a scale of 1 to 10. This is undertaken in the "display decision impact and comments" sub-process executed within the decision evaluator component.

The use case diagram models the functionality of the system from the perspective of external agents, called "actors". It shows the participation of the different actors for each of the use cases (Rumbaugh et al., 2004).

Three types of users interact with the DES, namely the "administrator", "decision-maker" and the "decision-validator". This last actor can have the same interactions as the decision-maker, in addition to the privilege to validate or reject decisions (his own decision or other users'). Decision-makers and validators can represent single users or decision committees (group users). The interactions of all three actors with the DES are represented in the use case diagram presented in figure 4.4.

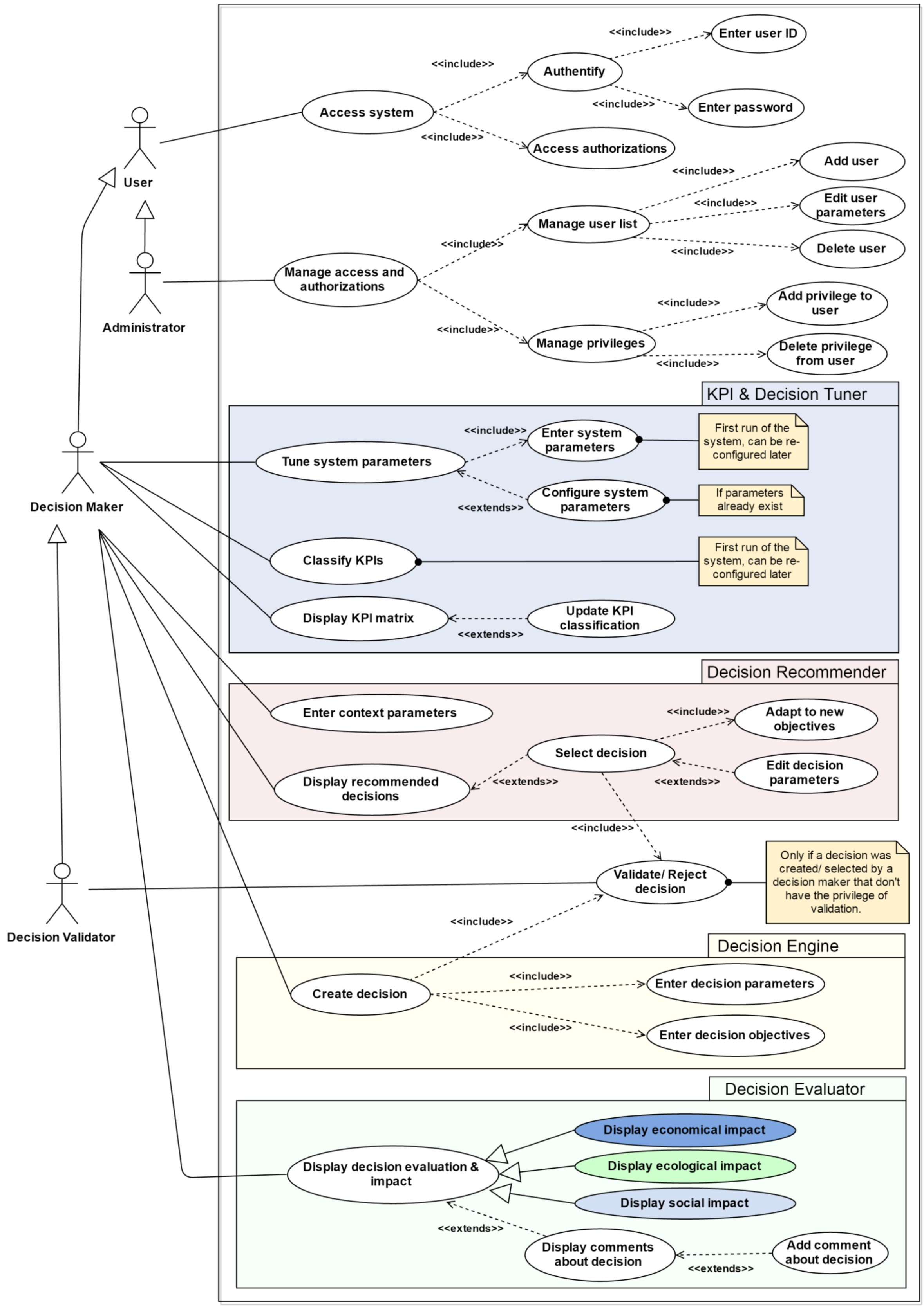

Figure 4.4 DES Use case diagram

Table 4.3 provides a brief description of the main use cases of the system:

<table>
<tr><th>Use case</th><th>Description</th></tr>
<tr><td colspan="2">Actor: All users</td></tr>
<tr><td>Access system</td><td>All types of users can only have access to the system through authentication (entering login, password). Their interaction with the system is dependent from their authorization level (or privileges).</td></tr>
<tr><td colspan="2">Actor: Decision-validator</td></tr>
<tr><td>Validate/reject decision</td><td>When a decision-maker without validation privilege creates a decision or selects a recommended decision, it will be viewed by an actor referred to as the "decision-validator" to accept or reject this decision. This actor has the same privileges as the decision-maker in addition to the validation privilege.</td></tr>
<tr><td colspan="2">Actor: Administrator</td></tr>
<tr><td>Manage access and authorizations</td><td>System administrators can manage the users list (add, edit, delete) and manage the user's privileges (grant, deny).</td></tr>
<tr><td colspan="2">Actor: Decision-maker</td></tr>
<tr><td>Classify KPI</td><td>The classification of the KPIs is also the task of the decision-maker, this includes (i) assigning the KPIs to different domains and sub-domains and (ii) classifying them by sustainability natures as ELPI, ENPI, and SPI.</td></tr>
<tr><td>Enter context parameters</td><td>Before making a decision and entering its parameters, the decision-maker is required to select and enter the context parameters of the decision which are:
– The domain and sub-domain
– The occasion description (or problem description)
– Objective(s) description
– Scope(s) description (National Research Council, 2014):
• Social scopes: diversity, quality of life, education, etc.
• Ecological scopes: resource management, pollution prevention, etc.
• Economical scopes: Cost savings, market reputation, profit, etc.
These context parameters help retrieving recommended decisions that are similar to the user context by applying the case retrieval of the CBR.</td></tr>
</table>

Use case	Description
Display KPI matrix	This matrix displays the classification of existing KPIs by domains, sub-domains and sustainability natures. The decision-maker can learn more about the KPIs while viewing this matrix and, if needed, the same user or another user can update the classification later.
Tune system parameters	Decision-makers can enter/configure the system parameters about the KPIs, the domains and sub-domains.
Display recommended decisions	If there are any recommended decisions similar to the pre-defined context, they are displayed to the user. The latter can choose a decision from the recommended decisions and adapt it to its current situation by assigning new objectives and editing its title and description.
Create decision	The user can create a new decision by entering its parameters (title, occasion description, solution description) and the decision-executor parameters (full name, e-mail). Decision creation also involves assigning objectives to the decision by selecting the KPIs to be affected, their planned values to be reached and the deadline dates for reaching those objectives.
Display decision evaluation & Impact	The decision-maker can display the evaluations (sustainable, very unsustainable, etc.) for all of the evaluated decisions. He can further display the decision's impact on KPIs. The impact dashboard displays the ecological, economic and social impact respectively for each decision. This is done by sorting and displaying the KPIs by their sustainability natures (ELPI, ENPI, and SPI). Two values of each affected KPI are displayed: the value before and after the decision and the percentage of enhancement/deterioration between the two values. In addition to the decision evaluation generated automatically by the system, the user can further understand its impact from other users' perspective by viewing their comments. He can also add his own comment.

Table 4.3 DES Main Use Cases

4.3.2 Global Data Structure

A class diagram is a graphic representation of declarative (static) model elements, such as types, classes, their contents and relationships. This diagram enables an overall view of the involved entities in the systems and the relationships between them, also known as the static view of the system (Rumbaugh et al., 2004). Figure 4.5 represents the class diagram for the DES.

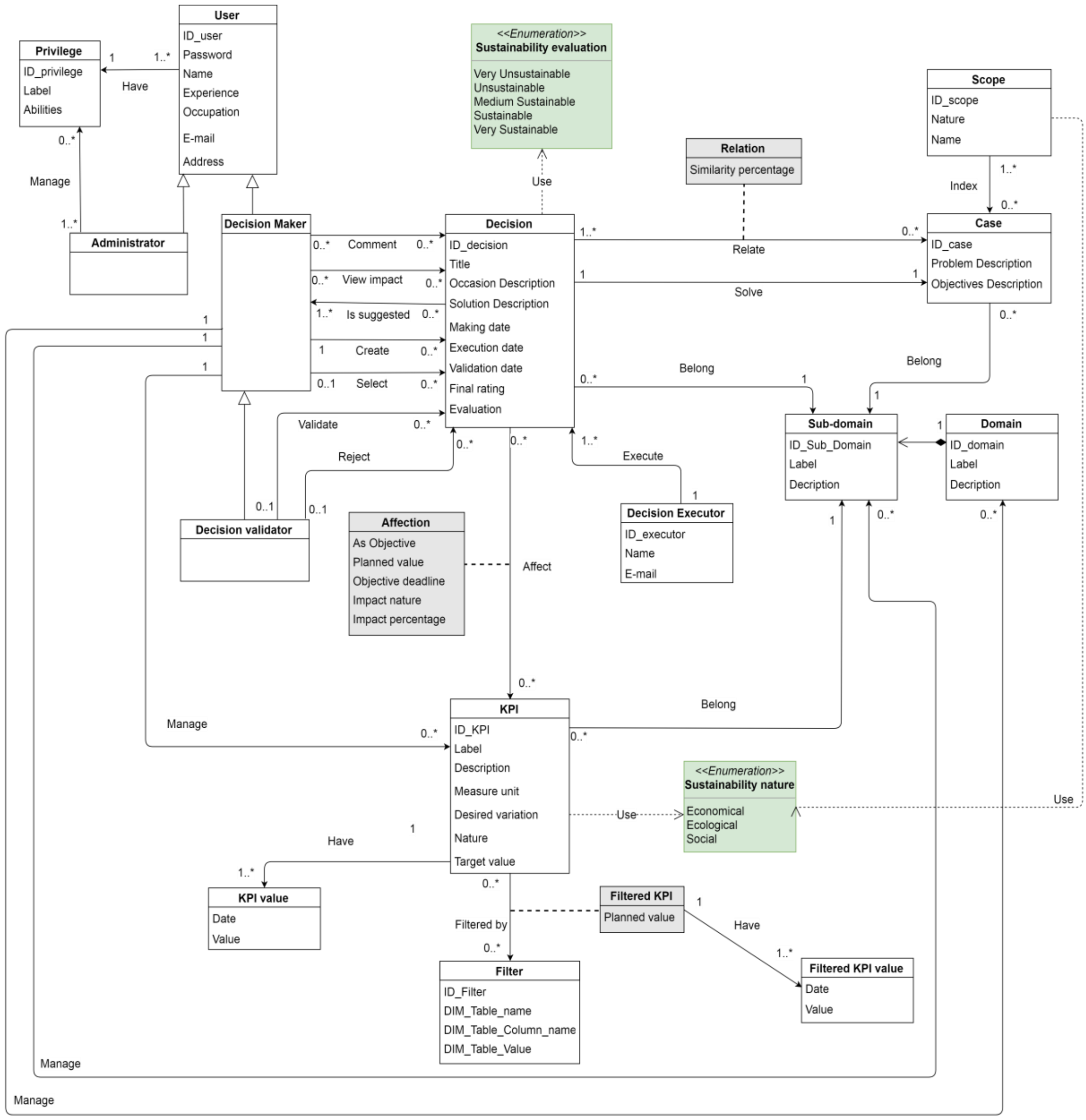

Figure 4.5 DES Class Diagram

This section provides a brief description of the main entities (classes) involved of the DES along with their attributes, data types of the attributes and their descriptions.

Class Privilege

The type of privileges attributed to the user. A user can create decisions only, create and validate decisions or can be an administrator.

Attributes	Data Type	Attribute Description
ID_Privilege	Integer	The unique ID of the privilege.
Label	String	The name of the privilege (e.g. "Creates decision only", "Creates + validates decision").
Abilities	String	The textual description of the ability granted by this privilege.

Class User

The user that interacts with the system. This user can represent a single person or a decision committee (group of persons).

Attributes	Data Type	Attribute Description
ID_User	String	The unique ID of the user.
Password	String	The password text used with the **ID_User** to access the system.
Name	String	The name and last name of the user. Or the official title of the decision committee if the user does not represent a single person.
Experience	Integer	The number of years of employment for the user.
Occupation	String	The current occupation of the user. (E.g**.** Sales Manager, CEO, IT consultant). If the user represents a decision committee, the value of this field is "(Group - No specific occupation)"
E-mail	String	The e-mail address of the user.
Address	String	The user's home address.

Class Administrator

Inherited from class user.

A type of user that can add, edit and delete other users and manage their privileges.

Attributes	Data Type	Attribute Description
--	--	Same inherited attributes of the class User.

Class Decision-maker

Inherited from class user.

A type of user that can create, select, view impact and comment decisions. A decision-maker may or may not have the privilege to validate his selected/created decision.

Attributes	Data Type	Attribute Description
--	--	Same inherited attributes of the class user.

Class Decision-validator

Inherited from class decision-maker.

A type of decision-makers that can validate their decisions and other decisions created by other decision-makers without the validation privilege.

Attributes	Data Type	Attribute Description
--	--	Same inherited attributes of the class decision-maker that inherits class **user**.

Class Decision-executor

The person (employee) responsible for the execution of a decision on the ground. The decision-executor is not necessarily a user with the system.

Attributes	Data Type	Attribute Description
ID_Executor	String	The unique ID of the decision-executor.
Name	String	The name and last name of the decision-executor.
E-mail	String	The E-mail address of the decision-executor.

Class Decision

The decision made by the decision-makers that the system will track, evaluate, display its impact and possibly recommend to users.

Attributes	Data Type	Attribute Description
ID_Decision	String	The unique ID of the decision.
Title	String	The brief title of the decision.
Occasion description	String	The textual description of the occasion (problem) for the decision.
Solution description	String	The long, textual description of the proposed solution. Alternatively called "Decision description".

Making date	Date	The date when the decision was created or selected among the recommended decisions.
Validation date	Date	The date when the decision was validated.
Execution date	Date	The date when the decision was executed.
Final rating	Double	The automatically generated rating of the decision based on its average impact on KPIs.
Evaluation	Sustainability evaluation	The automatically generated evaluation of the decision based on final rating. This value can be one of the five values presented in the enumeration class <<Enumeration>> Sustainability evaluation.

Class KPI

The KPI that is used as one of the measures for organizational performance in different domains and activities (Parmenter, 2015).

Decision-makers always aim to enhance the values of KPIs through their decisions.

Attributes	Data Type	Attribute Description
ID_Decision	String	The unique ID of the decision.
ID_KPI	String	The unique ID of the KPI.
Label	String	The brief label or title of the KPI.
Description	String	The measure unit of the KPI (e.g. in grams, in liters, percentage, units)
Measure unit	String	The desired variation of the KPI values (i.e. Increasing, Decreasing). It is important for decision-makers to learn the meaning, importance and desired variation for each KPI (Warren, 2011). (E.g. Increasing the values of the KPI "Profit per employee", Decreasing the values of the KPI "Cost per employee training"...)
Nature	Sustainability nature	The sustainability nature of the KPI, this field can take only the values in the enumeration class <<Enumeration>> Sustainability evaluation.
Target value	Double	The target value for the KPI.

Class Affection

The association class between the decision and the affected KPI containing both of their IDs, the affection type, planned values and deadlines for objectives and the impact nature and intensity.

Attributes	Data Type	Attribute Description
As objective	Boolean	Indicating that this affection of the decision on the KPI was direct or indirect. A direct affection means that the affected KPI is set as an objective for the decision, this value is then turned TRUE.
Planned value	Double	The planned value for the KPI that was set as objective when the decision was created/selected.
Objective deadline	Date	The deadline date for the objective KPI to reach the planned value. This date is also set when the decision was created/selected.
Impact nature	String	The impact of the decision on the KPI (**i.e.** positive, negative, no affection).
Impact percentage	Double	The percentage (%) of the increase or the decrease of the KPI value that was affected by the decision.

Class KPI Value

The KPI value imported with its date from the DWH. This value is identified by the KPI ID and the date since the KPI can only have one value in a certain date.

Attributes	Data Type	Attribute Description
Value	double	The value of the KPI.
Date	Date	The date stamp for the KPI value.

Class Filter

The filter (or perimeter) that can be applied for the KPI. Filtering KPIs can be possible by defining specific values within the columns of dimension tables (e.g. "Productivity of employees where DIM_Factory.Region="east" rather than tracking the productivity of employees in all of the factories of the organization). Multiple filters can be applied to a single KPI.

Attributes	Data Type	Attribute Description
ID_Filter	String	The unique Id of the filter.
DIM_table_name	String	The name of the dimension table involved in the KPI filtering.
DIM_table_column_name	String	The name of the column in the dimension table involved in the KPI filtering.
DIM_table_value	Generic (multiple types of values)	The value of the column in the dimension table involved in the KPI filtering. It should be noted that this value has a generic type (Integer, Double, Char, String) depending from the type of the selected column (e.g. Type=String if column "Region.name" was selected as a filter, Type=Double if column "Product.Price" was selected as a filter).

Class Filtered KPI

The association table between the KPI and the filter (perimeter) containing both of their IDs and the planned value for KPI restricted to the filter.

Attributes	Data Type	Attribute Description
Planned value	Double	Planned value for only the filtered KPI. This value is provided as an input from the user.

Class Filtered KPI Value

The filtered KPI value imported with its date from the DWH. This value is identified by the filtered KPI ID and the date since the filtered KPI can only have one value in a certain date.

Attributes	Data Type	Attribute Description
Value	Double	The value of the filtered KPI.
Date	Date	The data stamp of the filtered KPI value.

Class Domain

The domain of the organization activities. It comprises smaller sub-domains.

Attributes	Data Type	Attribute Description
ID_domain	String	The unique ID of the domain.
Label	String	The name or the label of the domain (e.g. Production, Sales, HR…etc.).
Description	String	The textual description of the domain.

Class Sub-domain

The sub-domain is part of a larger domain. It is used as the relationship between decisions and KPI. Both KPIs and decisions belong only to one sub-domain and decisions can only affect KPIs in the same sub-domain directly or indirectly.

Attributes	Data Type	Attribute Description
ID_sub_domain	String	The unique Id of the sub-domain.
Label	String	The name or the label of the sub-domain (e.g. Manufacturing (in production), After-sales (in Sales), Training (in HR)…etc.).
Description	String	The textual description of the sub-domain.

Class Scope

The scope (or topic) is a keyword or expression that is used to index cases for easier and more effective case retrieval in the CBR. It represents the topic of the objectives or the current problem that are defined in the context definition.

Attributes	Data Type	Attribute Description
ID_Scope	String	The unique Id of the scope.
Nature	Sustainability nature	The sustainability nature of the scope, this field can take only the values in the enumeration class <<Enumeration>>Sustainability evaluation.
Name	String	The name of the scope. It is dependent to the sustainability nature. Examples of the scope's name: • **Social scopes:** Diversity, Quality of life, Equal opportunities... • **Ecological scopes:** Resource management, Habitat preservation. • **Economical scopes:** Cost savings, Market reputation, Profit ...

Class Case

The case is a problem situation. A previous case is an experienced problem and solution that can be re-used to resolve current and future problems.

However, a new case is the current context described by the user by describing the problem and the objectives. The DES will retrieve the decisions that solved previous cases to recommend them for solving new cases.

Attributes	Data Type	Attribute Description
ID_Case	String	The unique Id of the case.
Problem_Description	String	The textual description of the problem in previous or current case.
Objectives_Description	String	The textual description of the objectives that were focused on with the previous case or recently defined in the current case.

Class Relation

The association table between the decision and the current case containing both of their IDs and the similarity percentage between them. This value is calculated within the decision recommender component and displayed for each decision in the list of recommended decisions.

Attributes	Data Type	Attribute Description
Similarity_percentage	Double	The percentage (%) of the similarity between the decision that solved a previous case and the current case. For the recommendation in the decision recommender component, the decision will be ranked based on its sustainability evaluation and its similarity to the current case.

The relationships between the entities shown in the class diagram can be listed as follows:

- The user can have only one type of privilege that will define his roles with the system. He can either be an administrator with administration rights, a decision-maker with decision creation or selection only or a decision-validator that can create new decisions, select recommended decisions and validate his or others' decisions. A single type of privileges can be granted to multiple users.
- The different types of users are depicted with the inherited classes administrator, decision-maker associated with the generalization to the parent class user. The decision-validator inherits from the decision-maker class.
- An administrator or multiple can manage the privileges of other users.
- A decision-maker can:
  - Manage (add, configure, delete) a domain or a sub-domain. More about this relation is presented in the KPI & Decision Tuner conception document.
  - Manage (add, configure, classify, delete) a KPI. More about this relation is presented in the KPI & Decision Tuner conception document.
  - Comment decisions or view their impacts on KPIs. Decisions can be suggested (recommended) to one or more decision-makers.

  - Create a new decision or select a recommended decision and adapt it to his current situation.

- A decision-validator can either validate or reject decisions. It is possible that a user with validation privileges does not take part in the validation or rejection of any of decisions.

- A decision can be suggested to multiple decision-makers. It must belong a single sub-domain and can affect one or multiple KPIs. The affection on KPIs are described with the attributes in the association class affection.

- A decision-executor must execute one or more decisions. This person is not necessarily a user with the system.

- A KPI have one or more values stored in the class KPI values. It can also be filtered by one or more filters. A filtered KPI is represented by the association class between KPI and filter named: filtered KPI.

- The filtered KPI also have one or more values stored in the class filtered KPI values.

- The sub-domain encloses decisions, KPIs and cases. It belongs to only one domain.

- The domain is composed of multiple sub-domains.

- A previous (learned) case is can be related to at least one or more decisions with a similarity percentage contained in the association class relation.

- The scope is used to index one or more cases. However, a scope can be non-related to any of the previous cases.

4.4 Front-End Modelling

The decision sustainability evaluation system allows different users to plan, make, track and evaluate decisions based on their sustainability impact, which is measured by KPIs. Some of these actions will require the user's interference and input, whereby it is necessary to have GUIs, classified here in the system under the component of "front-end".

This component comprises different web-based interfaces with which the user (represents a single person or a decision committee) interacts, it must allow them to:

- Login/Logout and access their authorizations depending from their privileges (decision-maker, decision-validator).

- Enter/Update the software parameters that the system will manage (KPIs, domains, sub-domains).

- Generate/Update/Display the classification of KPIs by domains and sub-domains.
- Display/Choose recommended decisions when there is an occasion for deciding.
- Create/Validate a new decision with defining its objectives if none of the recommended decisions is chosen.
- Display the sustainability evaluation of the previous decisions and their impact on different KPIs.

In addition this component must allow the administrators to:

- Login/Logout.
- Manage the systems' users (add, edit, show list, delete).
- Manage user's privileges (add, edit, delete).

Further details about those functionalities are presented in the requirements sub-section.

4.4.1 Requirements Definition

The "front-end" component functional requirements can be classified as follows:

Authentication & Authorization: The difference between the two terms is that authentication means requiring the subject (user) to demonstrate some form of evidence (fingerprint, password, etc.) to prove its identity. The authorization comes after the successful authentication to verify that the subject (user) has access to specific features or resources (Lai et al., 1999).

To have confidential and secured access to the DES, the registered user must enter his identification code (login) and his password using text fields then click the "Login" button. A hypertext link "Forgot your password?" offering the possibility for the user to recover his password is typical for all authentication-requiring applications and services. The DES also features this password recovery. Only authorized interfaces should appear to the users with different privileges. User management interfaces should only appear to administrators. Decision-making interfaces, decision evaluation and comments interfaces and system parameters configurations appear only to decision-makers. The decision validation interfaces can only appear to decision-makers with validations privileges (decision-Validator).

Administrator View: The requirement of the admin interface is to display a form for adding new users and their privileges in the system in addition to displaying the registered users. From the registered user list, the administrator can select a user then choose a command button enabling to show details, edit parameters or delete.

The administrator can grant system administration rights for other users. However, he does not take part in the decision-making process, whereby he does not have access to the decision-related GUIs in the front-end component.

Decision-Maker View: The interfaces for decision-makers (stakeholders) are dependent on their privileges granted by the administrators.

In the first access of the system by a non-admin user, he will be required to initialize the system parameters. This means that he need to use two types of forms within the interface to enter the KPIs parameters (ID, name, description and measure unit, etc.) and the domains/sub-domains parameters (ID, name, description). These entered parameters can require updates and configuration from the same user or others.

Another initialization for the system is also required, whereby the user must classify the KPIs by domains, sub-domains and their sustainability natures via an interface. This interface displays a list of the different KPI names in front of drop-down lists for the available domains, sub-domains and sustainability nature (ecological, economic, social), whereby the user selects the right classification for each KPI. The classification can also be updated later when displaying the KPI matrix dashboard generated by the system. If there is an occasion for making a decision, the system must display an interface enabling the input of the parameters of the decision context. The users select the domains, sub-domains and the scopes (keywords for the context) via drop-down lists and use the available text fields to describe the objective(s) and the occasion for the decision.

Before heading straight to planning and making a new decision, the interface must display a list of recommended, previously-made decisions related to the defined context. These decisions are found using the case-based reasoner component, which uses the CBR paradigm to search and retrieve previous similar cases and their solutions (decisions) (Aamodt and Plaza, 1994). The recommended decisions' titles, previous impact on KPIs and the sustainability evaluations must be displayed in a sorted list, whereby they are sorted by their sustainability evaluations and their similarity to the defined context. The user may choose to re-make a previous decision from this displayed list or create a new decision.

In the decision creation interface, the user needs a coherent form containing text fields to enter the decision parameters (ID, title, occasion description and solution description) and the parameters of the decision-executor (name, last name, e-mail). He will also need to select the KPIs to be affected by the decision along with the objective values and deadline dates. An optional filtering for the objective KPIs is available by selecting the concerned dimension tables, columns and values that will restrict the calculation of the KPI values (e.g. Recyclable raw material per product - for the factory "ABC" and the product model "X"). After the decision creation is confirmed, it will need to be validated by the same user whether the user has the privilege of validating decisions. If this is not the case, it will need to be validated by another user with this privilege.

A list of decisions awaiting their validation must be displayed to the users with the validation privilege. The decision's title, maker, executor, domain/sub-domain, description and objectives must all be shown to the decision-validator before the validation or the rejection via command buttons (e.g. "Validate Decision", "Reject Decision").

The system should display an interface containing the previous and evaluated decisions. For each decision, its sustainability evaluation is shown (e.g. very sustainable, medium sustainable, etc.). The system should also display the recorded impact of the decision on the different types of KPIs, namely ELPIs (ecological performance indicators), ENPIs (economic performance indicators) and SPIs.

The system should enable users to comment decisions and attach various types of files to their comments. This is unstructured knowledge sharing that is proved useful for organizations (See global conception document).

4.4.2 Graphical User Interfaces

Prototyping GUI helps to present the graphical representation, contents and the different functionalities for the system, although no linking patterns between the interfaces are presented. For this reason, the interface flow diagram is elaborated.

4.4.2.1 Interface Flow

The interface flow diagram (Alternatively called the window navigation diagram) presents high-level relationships and interactions between the system's different interfaces (Ambler, 1998a, 1998b, 2001, 2004; Page-Jones, 1995).

Figure 4.6 presents the flowchart diagram for the front-end interfaces of the DES. The boxes represent the different interfaces (forms, data lists, dashboards, dialog boxes) and the arrows present the possible flow between interfaces and the actions necessary to jump from an interface to another (Ambler, 2004).

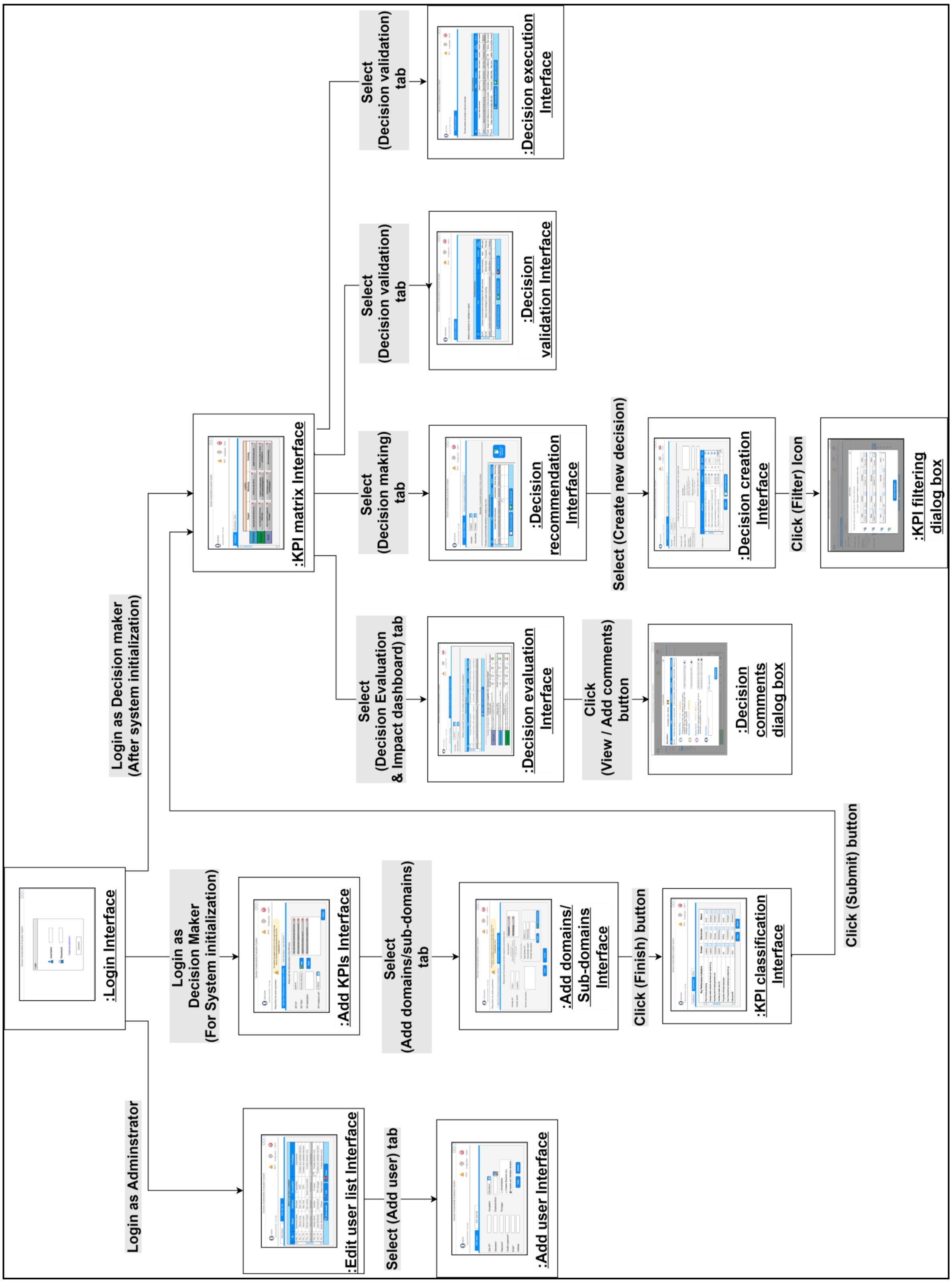

Figure 4.6 Interface flow diagram for the DES user interfaces

4.4.2.2 Interfaces Description

After a successful authentication, the system will identify the user and its privilege with the system (administrator, decision-maker only, decision-maker + validator). Only authorized interfaces and functionalities will be available for each type of users.

The following interface is shown only for administrators to register new users by entering their attributes and selecting their privileges.

As for managing the registered users of the system, the "Edit user list" interface offers the possibility to administrators to show the details of users, edit their parameters and delete them from interacting with the system.

The first non-admin user to login to the system will be in charge of entering the system parameters. He must enter the KPI details (ID, title, description, measure unit, desired variation) with a dedicated interface. System parameters also include entering the details of domains and sub-domains (ID, name, description). The DES will check the values of those KPIs to verify how the decisions have affected them (enhancement or deterioration) and evaluate these decisions based on their average impact on KPIs. KPIs and decisions are classified under sub-domains, which also belong to domains.

The user will new need to assign the entered KPIs by their respective domains/sub-domains and classify them by their sustainability nature (ecological, economic or social) using the following interface.

The output of the KPI classification is called the KPI matrix, which is displayed to the users in the following interface. This interface enables users to learn about the KPI classification, update the classification or delete KPIs.

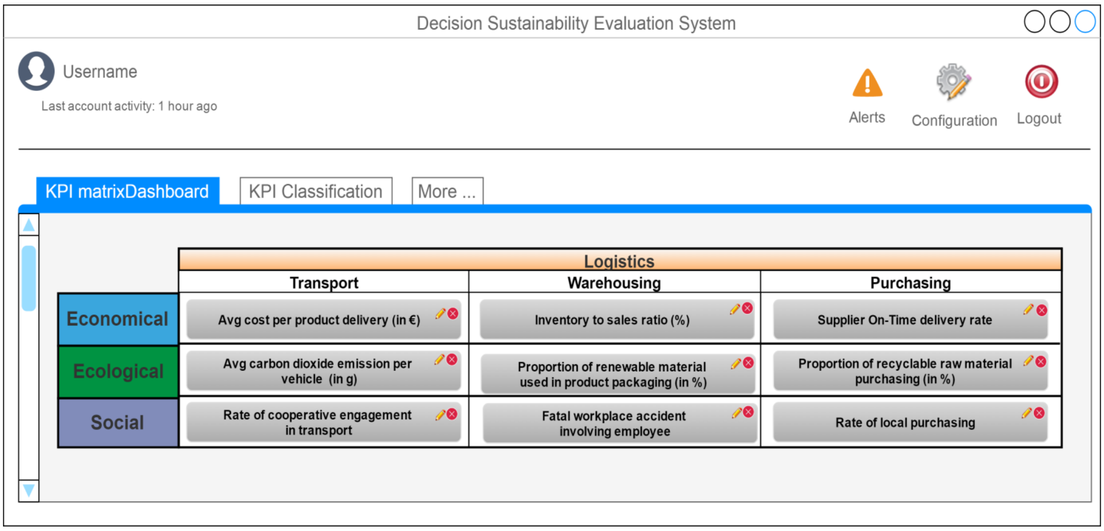

Figure 4.7 KPI matrix Interface prototype

If an occasion for making a decision is identified by a user, a created decision interface is available, which enables the user to select the main context parameters (domain/sub-domain) of the decision. These fields are mandatory because both a recommended decision and a newly-created decision must belong to a single sub-domain. He may also enter additional context parameters of the decision, which are:

- Occasion description (or problem description).
- Objective(s) description.
- Scope(s) description, examples:
  - **Social scopes:** Diversity, Quality of life, Education...
  - **Ecological scopes:** Resource management, Pollution prevention...
  - **Economic scopes:** Cost savings, Market reputation, Profit...

These additional parameters will further specify the context of the user and helps enhancing the recommendation of decisions. Recommended decisions become sorted not only by their sustainability nature but with their similarity to the context features entered by the user. Retrieving previous decisions similar to the context is task of the case-based reasoner component. Ranking them is the task of the decision recommender.

The interface displays the recommended decisions previously made belonging to the selected sub-domain. The user can show the previous impact for each selected decision and choose whether to select and adapt it or make a new decision.

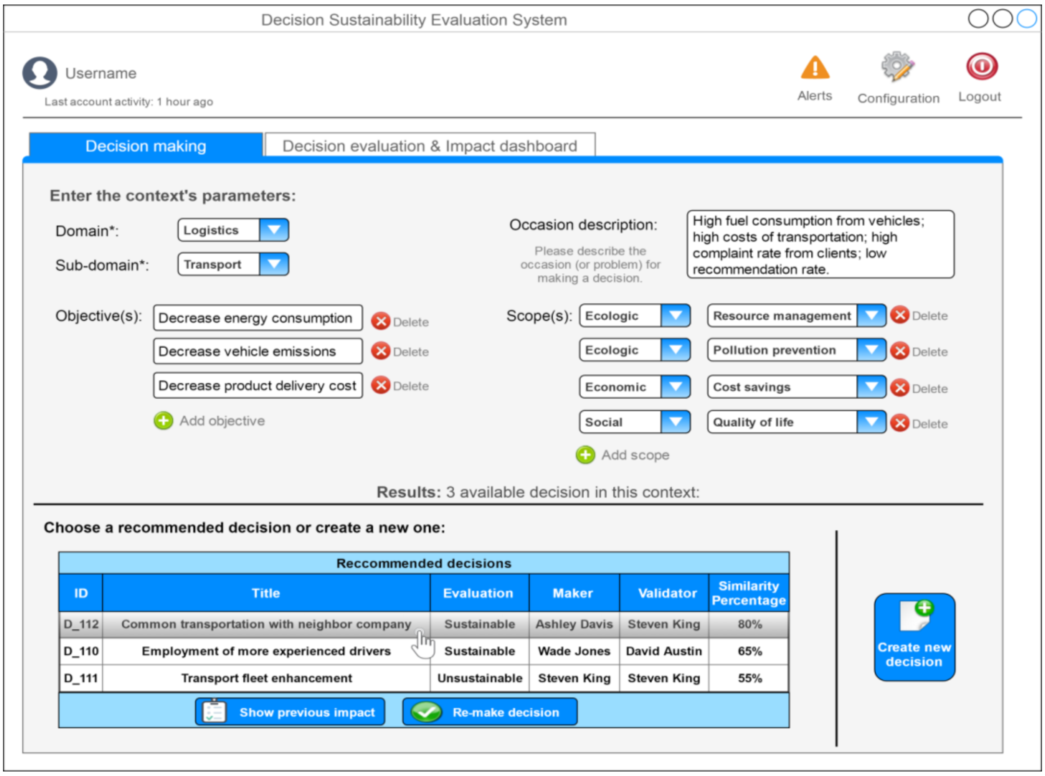

Figure 4.8 Decision-making (Recommended) Interface prototype

If none of the previous decision seems suitable to the user's vision, clicking the button "Create new decision" will redirect to the following interface (figure 4.9). This interface will enable him to assign the decision parameters (ID, title, occasion description, solution description) and the decision-executor parameters (name, last name, e-mail).

For the objectives of the decision, the decision-maker interacts with a checklist of all KPIs related to the decision's sub-domain ("Transport" here). From this list, the user selects the related indicators and assigns for each indicator the goal desired to achieve through this decision, as well as the deadline of achievement. The decision will be evaluated by the system based on these objectives.

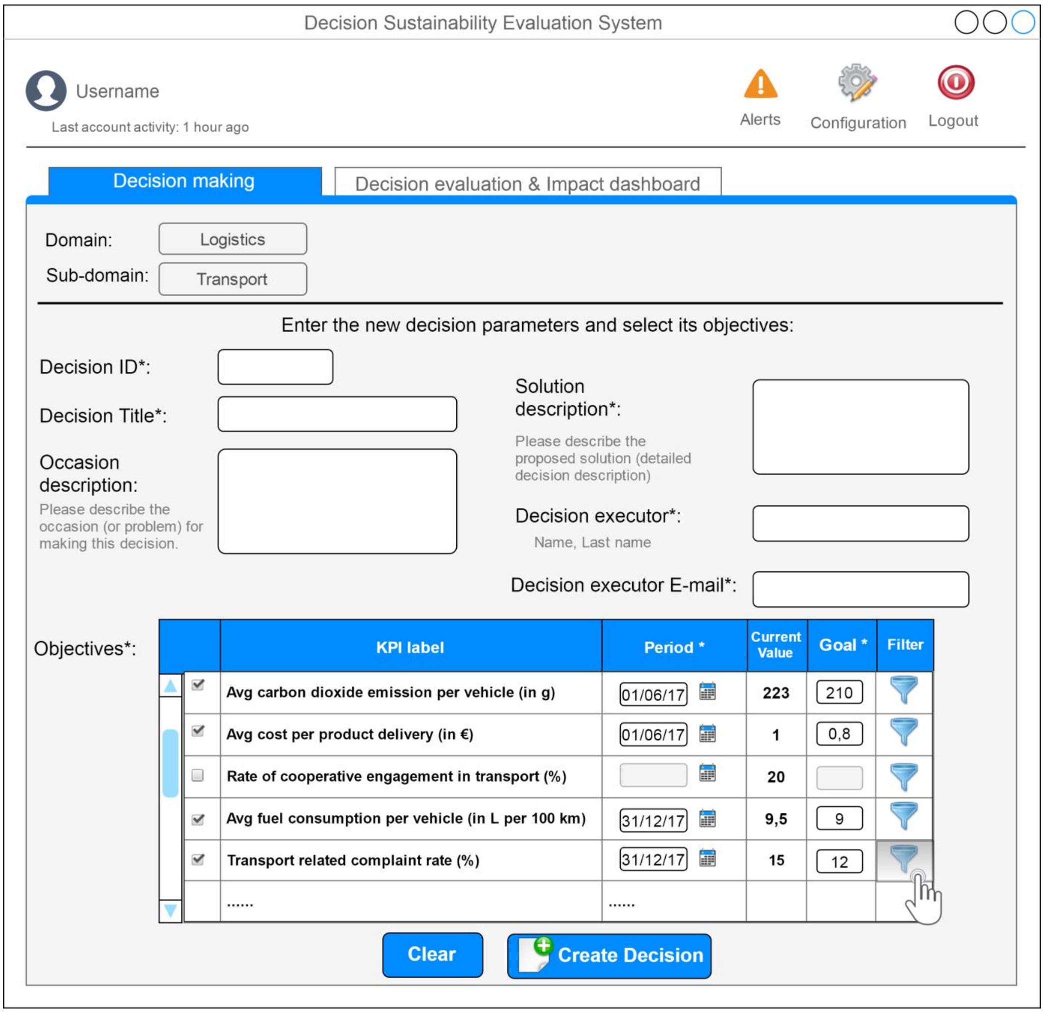

Figure 4.9 Decision-making (creation) Interface prototype

From the checklist displayed in the UI presented in figure 4.9, the user can add filters for a selected KPI to highlight that the decision concerns enhancing the KPI value for only a certain perimeter and not on a global scale. The user clicks on the filtering icon for the KPI " " in the decision creation UI and a dialog box appears. This dialog enables defining the dimension tables, columns and their values that will serve as the perimeter of the KPI.

Filtering the KPI "transport-related complaint rate (%)" is a representative example. Rather than tracking this value on the global scale and calculating the average of all recorded transport-related complaints, here the user specifies that the calculation of this value will concern only the regions "Montréal" and "Quebec" and the transportation of the product model "Car paint X20".

With the privilege of validating the decisions, the user can validate his own decision from this interface. If this is not the case, the decision(s) will be awaiting the validation of a superior user (decision-validator). The following interface shows all pending decisions waiting for the feedback of the decision-validator to view their details (objectives, periods, description) and whether to accept or reject them.

Another interface permits showing all of the validated decisions, although not all of these decisions have been executed in reality or they may have already been executed but the system is not aware. Users will have to interact with this interface to change the decision status to "executed" and assign its execution date. The system will need these two pieces of information to start evaluating the decision only after it has been executed.

After entering one or more decisions via the previous interfaces, the system will evaluate them based on their impact on the planned KPIs (direct impact) and other KPIs from the same sub-domain (indirect impact). The evaluation (from unsustainable to very sustainable) will be shown for each decision in the interface represented in figure 4.10.

Selecting a decision will show more details about its sustainability impact based on KPI ameliorations or deteriorations. Three dashboards are shown for each selected decision to represent its impact on SPIs, ENPIs and ELPIs.

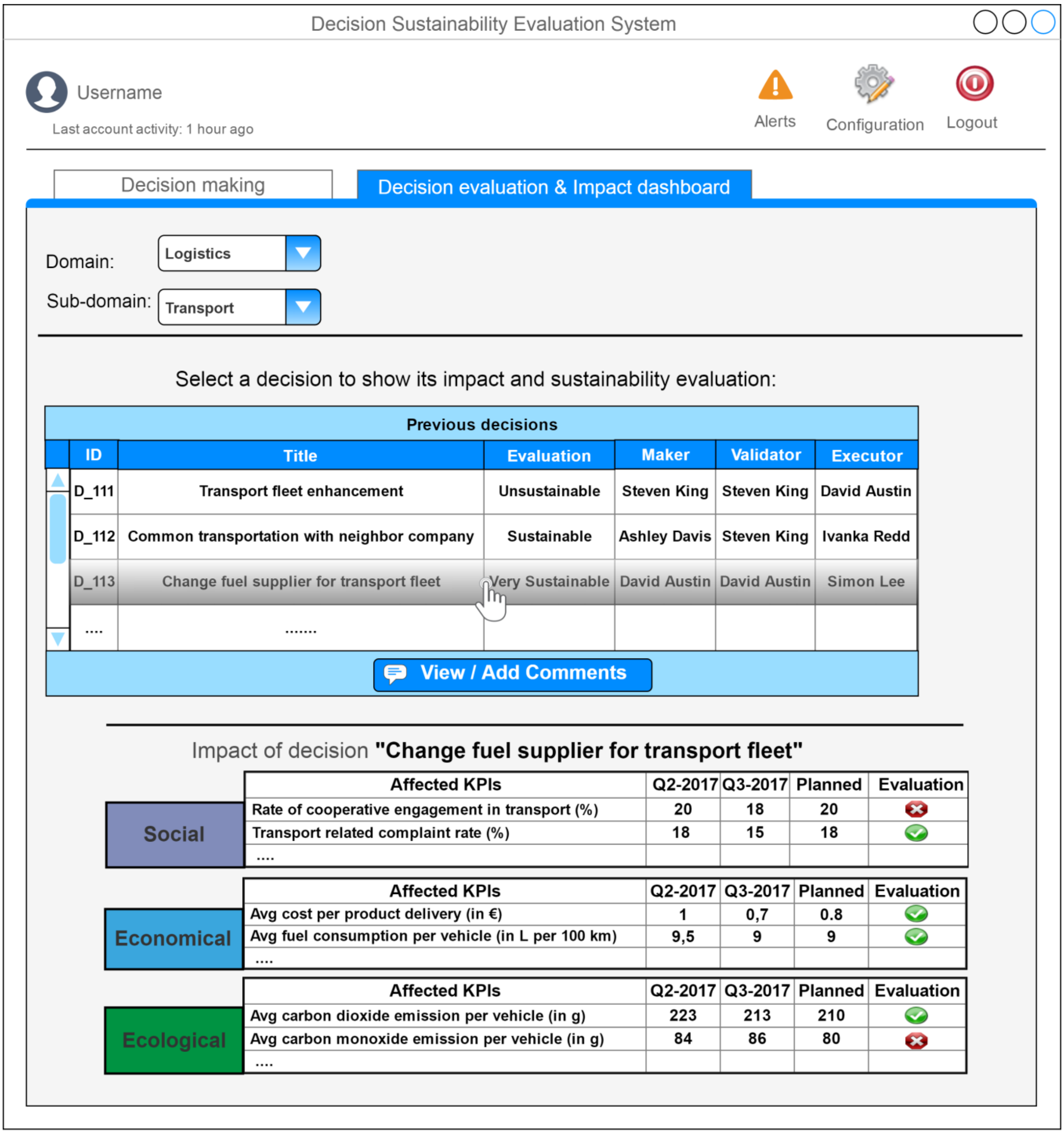

Figure 4.10 Decisions' impact and evaluation Interface prototype

The users may also need to view/add feedback about evaluated decisions. Viewing comments and various types of attached files will further make the user understand the decision's outcomes and evaluation from the other users' perspectives. By selecting a decision pressing the button "View / Add Comments" in the decision evaluation UI, a comments dialog box appears.

This dialog box contains the average user rating and the comments section for the selected decision. It shows the comments of different users, the dates for each of the comments and the users' rating for the decision (on a scale of 1 to 10). Various types of files can be attached to comments if the user wants to enrich his comments or present proof(s) and justifications. The current user can add a simple textual comment, upload a file or execute both of these actions.

4.5 KPI & Decision Tuner Modelling

Current environmental DSS and EMIS give stakeholders the ability to assess, optimize and report on the current effects of their processes and operations on the environment (Marx Gómez and Teuteberg, 2015). However, those systems do not consider nor calculate the impact of the decision on the environment or its sustainability. In order to make more sustainable decisions, we need to provide individuals in the organization with a DES. This system is expected to enable the organization's stakeholders to track, evaluate, recommend and comment decisions.

The sustainability evaluation and the recommendation of the decision's will be based on its impact on the different types of KPIs, namely social performance indicators (SPIs), ecological performance indicators (ELPIs) and economic performance indicators (ENPI). Of course, these indicators are in the same sub-domain of the decision (e.g. warehousing, after-sales, HR training, manufacturing, etc.).

In order to enable an accurate and detailed evaluation for the decision based on clearly-defined KPIs, we need to (Rezgui and Marx Gómez, 2016b):

- Generate the KPIs and the domains/sub-domains that the system will be adapted with. This is provided as inputs by the users.
- Classify the KPIs by sustainability nature (social, ecological, economical) and the domains/ sub-domains of the organization.
- Monitor the KPI values for enhancements or deteriorations.

To satisfy these needs, a component named the "KPI & Decision Tuner" exists within the DES. The functional requirements of this component are presented in the following section.

4.5.1 Requirements Definition

The "KPI & Decision Tuner" component's functional requirements can be classified as follows:

Obtain the initial system parameters: This means enabling the user to enter the list of KPIs and their parameters and the list of domains/sub-domains and their parameters. This step is executed when initializing the system for the first time and those initial parameters can be configured later.

Obtain & Display KPI classification: The system should enable the user – via another interface – to classify the KPIs by:

- The existing domains and sub-domains.
- The sustainability nature of the KPI, whether social, ecological or economic.

- After the input by the user, a matrix called the "KPI matrix" is generated by the system as the main output for this module. This matrix is presented in a dashboard for the user in the IU and it shows the KPIs in relation with their respective domains, sub-domains and sustainability nature. The user may choose to configure the KPI classification from this displayed matrix. The decision database will store the classification of KPIs.

Monitor KPI values: The system is also responsible for monitoring the changes (enhancement or deteriorations) for the KPI values after making decisions. These values are imported from the DWH with date stamps for each value.

4.5.2 Use Case

The use case diagram for the *KPI & Decision Tuner* is presented in figure 4.11. It describes the user's interaction with the DES, specifically with the *KPI & Decision Tuner* component. One type of actors interacts with this component, which are decision-makers, regardless of their privileges (with or without the decision validation privilege). The decision-maker will have the right to enter/configure the system parameters and enter/update the KPI classification.

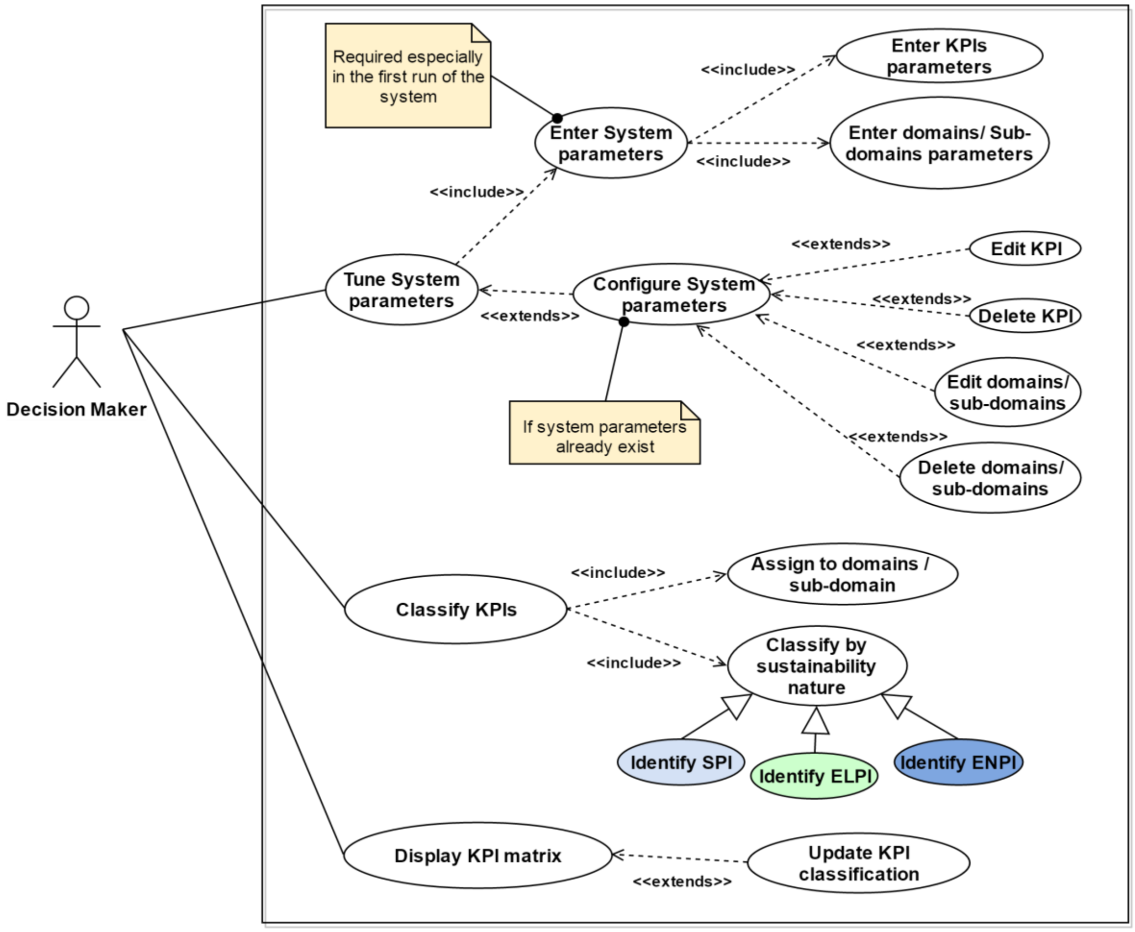

Figure 4.11 Use case diagram - KPI & Decision Tuner component

Table 4.3 provides a brief description of the main use cases between the actor (decision-maker) and the KPI & Decision Tuner component and explains the different steps, conditions and extensions of these use cases.

<table>
<tr><th>Use Case</th><th>Description</th></tr>
<tr><td colspan="2">Actor: Decision-maker</td></tr>
<tr><td>Tune system parameters</td><td>Tuning system parameters means entering or configuring system parameters (KPIs, Domains, and Sub-domains).

Decision-maker Decision-makers will be required to initialize the system parameters in the first run of the DES by adding the parameters of KPIs (ID, label, description, measure unit, desired variation) and the parameters of domains and sub-domains (ID, title, description).

Only when the system already has parameters, the user can (optionally) edit or delete KPIs or domains.</td></tr>
<tr><td>Classify KPI</td><td>The classification of the KPIs in also the task of the decision-maker, this includes:
• Assigning the KPIs to different domains and sub-domains.
• Classifying them by sustainability natures as ELPI, ENPI, and SPI.</td></tr>
<tr><td>Display KPI matrix</td><td>This matrix displays the classification of existing KPIs. The decision-maker can learn more about the KPIs' while viewing this matrix before creating, selecting or viewing impacts of decisions.

If an update is needed after viewing this matrix, the decision-maker can update the classification.</td></tr>
</table>

Table 4.4 KPI & Decision Tuner Main Use Cases

4.5.3 Component Process Interactions

The process of the KPI & Decision Tuner will be described in the following cross-functional flowchart diagram in figure 4.12.

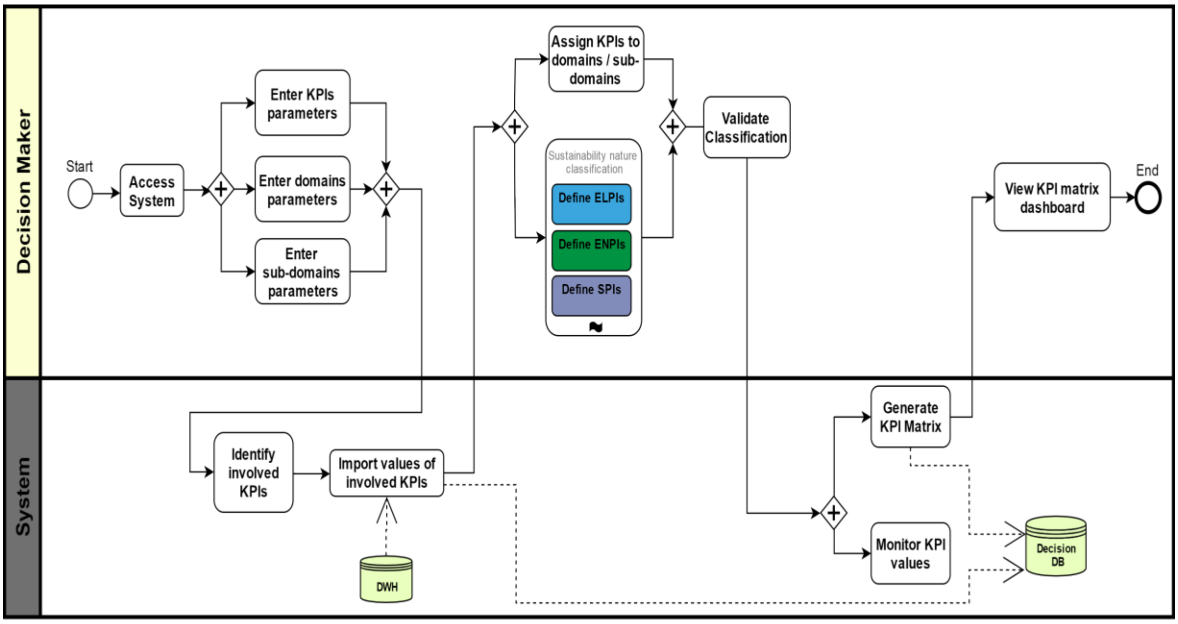

Figure 4.12 Flowchart diagram - KPI & Decision Tuner component

The flowchart diagram for the KPI & Decision Tuner component illustrated by figure 4.12 represents the following scenario:

A decision-maker can start using the DES after he has been registered and given privileges by an administrator. After a successful authentication, the decision-maker must first enter the system parameters (KPIs, domains, sub-domains) in any order. This step is required if those parameters do not already exist in the system. The system identifies the involved KPI values and imports their values from the DWH and saves them in the decision DB with the help of ETL mapping done by technical users. Those system parameters can be configured later.

The decision-maker also classifies the KPIs by domains/sub-domains and sustainability nature (defining ELPIs, ENPIs or SPIs). In some cases, none of the existing non-classified KPIs are related to a sustainability field (e.g. social), which is why the activities of defining the ELPIs, ENPIs or SPIs are grouped in the BPMN diagram in an ad-hoc sub-process. The activities grouped in ad-hoc processes can be executed in any order or skipped.

The system generates the ‘KPI matrix’ based on the inputted classification, which is also stored in the decision database. The KPI matrix is displayed in multiple dashboards showing the KPIs in relationship with their respective sustainability classes, domains/sub-domains. It helps the decision-maker to learn more about the KPIs, verify the correctness of the classification and possibly update the KPI classification.

4.5.4 Component Data Structure

The entities displayed in the class diagram, their attributes and their relationships are all explained in the global class diagram description.

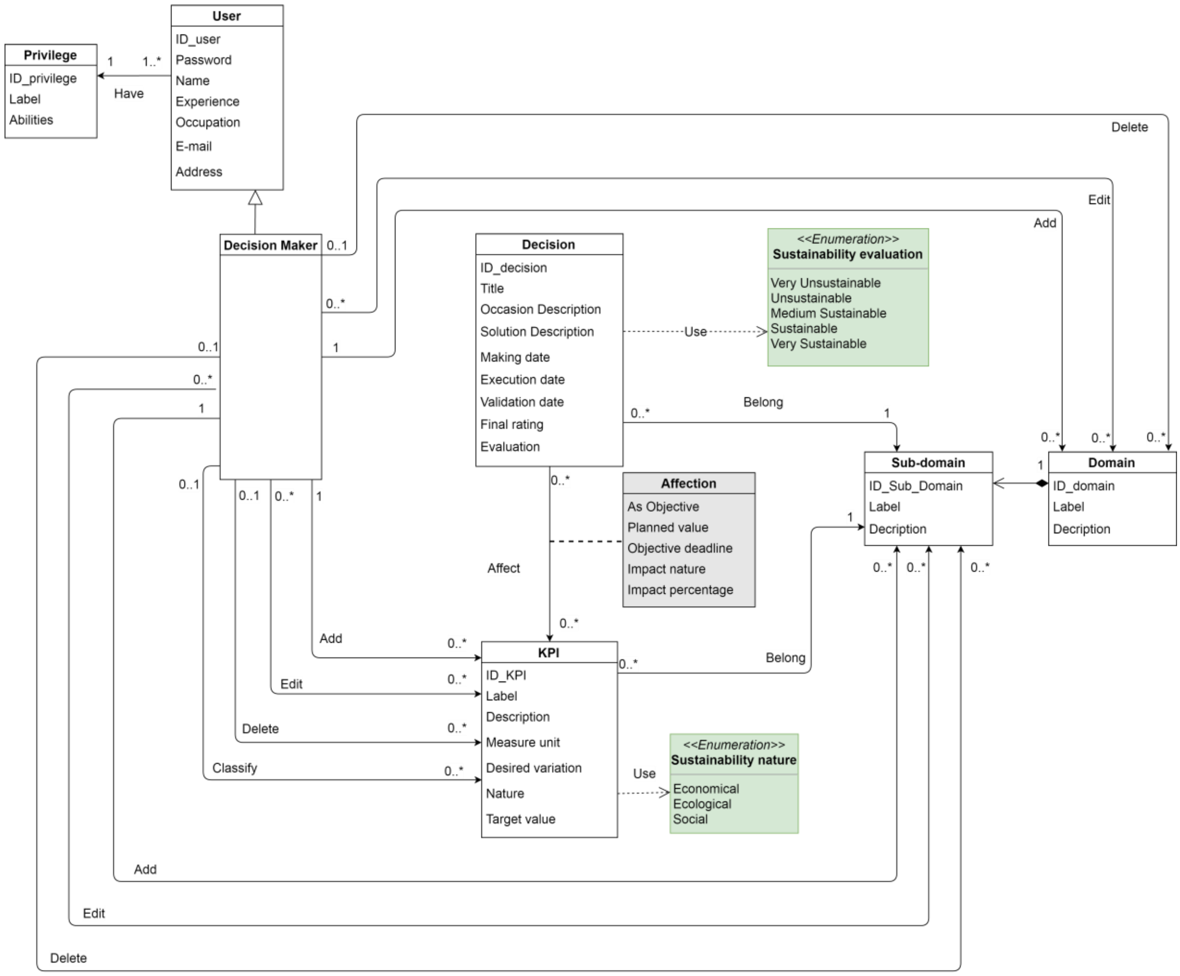

Figure 4.13 KPI & Decision Tuner Class Diagram

For the KPI & Decision Tuner, the specific relationships between the entities in the class diagram displayed in figure 4.13 can be summarized as follows:

- The user can have only one type of privileges that will define his role with the system. He can either be an administrator with administration rights or a decision-maker with decision creation and possibly with decision validation privileges. A single type of privileges can be granted to multiple users.
- The decision-maker is the sole user with this component. The class for the decision-maker associated with the generalization to the parent class user to inherit its attributes and associations.

- A decision-maker can:
  - Add or delete one or more domains or sub-domains. Multiple decision-makers can apply changes to domains or sub-domains optionally.
  - Add or delete one or more KPIs. Multiple decision-makers can apply changes to KPIs' parameters.
  - Classify the KPI by domains, sub-domains and sustainability natures. A KPI can remain unclassified (temporarily) after its parameters have been entered by a decision-maker, it can be classified later by the same decision-maker or by another one.

4.6 Decision Evaluator Modelling

One of the core advantages of the decision sustainability evaluation system is the automation of evaluating the decisions based on their impact on different KPIs.

The decision evaluator component is responsible for evaluating all the executed decisions entered via the system, specifically with the help of decision engine component. The evaluation process needs:

- Monitoring the decision's direct and indirect impact on the KPIs in the same sub-domain (explanations of the direct and indirect impact on KPIs will be provided in the next section).
- Providing an automatic evaluation for the decision based on its recorded impact on KPIs. A decision rating can vary from "(1) very unsustainable" to "(5) very sustainable".
- Displaying a decision evaluation dashboard in the GUI showing the evaluation of the decision and its impact on the different classes of KPIs, namely SPIs, ENPIs and ELPIs.
- Commenting decisions. In addition to the automatic decision evaluation by the DES, users may need to add comments to evaluate decisions to further understand the outcomes of the decisions from the stakeholders' perspective. The comments may be in the form of texts, reports, images, audio files, videos, etc. This is called unstructured knowledge sharing (Baars and Kemper, 2008b; Blumberg and Atre, 2003b; Rao, 2003b).

The functional requirements of the decision evaluator component are presented in the following section.

4.6.1 Requirements Definition

The "decision evaluator" component's functional requirements can be classified as follows:

Monitor the decision impact on KPIs: For every creation of a new decision (or the selection of previous one), multiple KPIs were selected to be affected by this decision as its "objective KPIs". A planned value and a deadline date are set for each KPI.

After the decision has been executed and its objectives' deadlines reached, the system should be able to evaluate the decision by monitoring all the values of its objective KPIs whether they have enhanced, weakened or were not affected. These previous and current values should be imported from the updated DWH. This is called monitoring the direct impact of the decision.

The system should also monitor the indirect impact of the decision. This means tracking the values of other KPIs that are assigned to same sub-domain with the decision (transport, production planning, HR staffing, etc.) but were not selected as objectives.

Rate the decision based on its KPI impact: This must also be provided by the system and executed automatically. The data gathered about the previous values of KPIs (before the decision execution) and their current values (after the decision's deadlines were reached) will be used to calculate the average enhancement or deterioration of all KPIs and the distances between their current and planned values. The decision rating will be based on this average percentage of the enhancement or deterioration of all KPIs with direct and indirect relationships and the distance between the current and planned values (Rezgui and Marx Gómez, 2016a). Several mathematical formulas should be applied to evaluate of the decision, whereby these formulas are presented below. The score (or the rating) of the decision should be on a scale of (1) to (5) as follows:

(1) Very Unsustainable	The decision significantly affected the KPIs values negatively.
(2) Unsustainable	The decision affected the KPIs values negatively.
(3) Medium Sustainable	No remarkable affections on KPIs were recorded. Slight enhancements or deteriorations.
(4) Sustainable	The decision affected the KPIs values positively.
(5) Very Sustainable	The decision significantly affected the KPIs values positively.

This evaluation will later be attributed to the decision and will be displayed when the system recommends decisions to the users in the decision recommendation UI or the decision impact and evaluation UI.

Generate the evaluation of the decision: The system must display decision's evaluation [(1) very unsustainable to (5) very sustainable] and its detailed impact on the different types of KPIs (social, ecological, economic). Three dashboards are shown for each decision representing the last, current and planned values of the SPIs, ENPIs and ELPIs with their dates.

Obtain comments on decisions: The decision evaluation from the users' perspective can be undertaken by enabling unstructured data sharing. The DES must enable the users to comment on decisions using texts and/or by attaching various types of files to previous decisions (documents, reports, audio, video, etc.). Each decision displayed in the evaluation interface will have a comment section containing user comments and enabling adding new comments (and/or attaching files).

The decision impact percentage on a single KPI **($\mathbf{Impact}_{KPI}$)** is based on the percentage of positive or negative variation between the value of the KPI before the decision and after the decision should be calculated as follows:

$$\text{Impact}_{KPI}(\%) = \frac{\text{last value } after\ decision - last\ value\ before\ decision}{last\ value\ before\ decision} * 100 \quad \textit{if KPI desired variation is maximizing}$$

$$= \frac{\text{last value } before\ decision - last\ value\ after\ decision}{last\ value\ before\ decision} * 100 \quad \textit{if KPI desired variation is minimizing}$$

With these two equations, the value of **($\mathbf{Impact}_{KPI}$)** is only positive when there is an enhancement for the KPI whether its desired variation is maximization or minimization. If this value is negative, this means that there was a deterioration.

The average direct impact on KPIs(**$\mathbf{Avg\_direct\_Impact}_{KPIs}$**)is calculated by:

$$\text{Avg}_{\text{direct}_{\text{impact}}}{}_{KPIs} = \frac{\sum_{i=0}^{i\leq\ nbr_{objectives}} \text{Impact}_{KPI} i}{nbr_{objectives}} \quad \text{if KPI was set as objective for decision}$$

Moreover, the average indirect impact on KPIs (**$\mathbf{Avg\_indirect\_Impact}_{KPIs}$**) is calculated by:

$$\text{Avg}_{\text{indirect}_{\text{impact}}}{}_{KPIs} = \frac{\sum_{i=0}^{i\leq nbr_{affected}} \text{Impact}_{KPI} i}{nbr_{affected}} \quad \textit{if KPI was not set as objective for decision}$$

Where **nbr_affected** is the number of affected KPIs that were not selected as objective for the decision, which means that the impact on the KPI is less or more than zero **($\mathbf{Impact}_{KPI} \neq 0$)**.

With the average impact percentages calculated for all the KPIs affected by the decision directly or indirectly, the decision's sustainability evaluation ($\mathbf{Rating}(\text{decision})$) can be calculated with the following function:

$$\boldsymbol{Rating}(decision) = \left(ImpF_d * \text{Avg}_{\text{direct}_{\text{impact}}}{}_{KPIs}\right) + \left(ImpF_{ind} * \text{Avg}_{\text{indirect}_{\text{impact}}}{}_{KPIs}\right)$$

With:

- ($\boldsymbol{ImpF_d}$) is the importance factor of the direct impact in the decision evaluation.
$$\boldsymbol{ImpF_d\ in}\ [\mathbf{0..1}]$$
- ($\boldsymbol{ImpF_{ind}}$) is the importance factor of the indirect impact in the decision evaluation.
$$\boldsymbol{ImpF_{ind}\ in}\ [\mathbf{0..1}]$$
- $\boldsymbol{ImpF_d + ImpF_{ind} = 1}$

The sustainability evaluation is based on the value final rating of the decision$\boldsymbol{Rating}(decision)$**,** which represents both direct and indirect impact percentages joined with their respective importance factors (weights).The evaluation is designed as follows:

(1) Very Unsustainable	$\boldsymbol{Rating}(decision) \leq -30$
(2) Unsustainable	$\boldsymbol{Rating}(decision)\ \boldsymbol{in}\ ]-30\,..-5[$
(3) Medium Sustainable	$\boldsymbol{Rating}(decision)\ \boldsymbol{in}\ ]-5\,..10[$
(4) Sustainable	$\boldsymbol{Rating}(decision)\ \boldsymbol{in}\ ]10\,..30[$
(5) Very Sustainable	$\boldsymbol{Rating}(decision) \geq 30$

4.6.2 Use Case

The monitoring of the KPI values and the rating of the decision is executed by the system and thus it is not shown here in this diagram. The decision evaluation GUI will not display any of the previous decisions unless it was evaluated by the system.

The use case diagram for the decision evaluator is presented in figure 4.14. It describes the user's interaction with the DES, specifically with the decision evaluator component. One type of actors interacts with this component, which are decision-makers, regardless from their privileges (with or without the decision validation privilege).

Table 4.4 provides a brief description of the main use cases between the actor (decision-maker) and the decision evaluator component and explains the different steps, conditions and extensions of those use cases.

Use Case	Description
Actor: Decision-maker	
Select domain/sub-domain	The decision-maker must first select the domain and sub-domain for the decisions that he wishes to see their evaluations.
Display evaluated decisions	This decision-maker displays all of the evaluated decisions classified under the selected domain. Optionally the user can select a decision to: - Display its detailed impact on KPIs. The impact dashboard displays the ecological, economic and social impact respectively for each decision. This is done by sorting and displaying the KPIs by their sustainability natures (ELPI, ENPI, and SPI). Two values of each affected KPI is displayed. The value before and after the decision and the percentage of enhancement/ deterioration between the two values. - In addition to the decision evaluation generated automatically by the system, the user can further understand its impact and evaluation from other users' perspective by viewing their comments on the selected decision. He can add his own comment as well.

Table 4.5 Decision Evaluator Tuner Main Use Cases

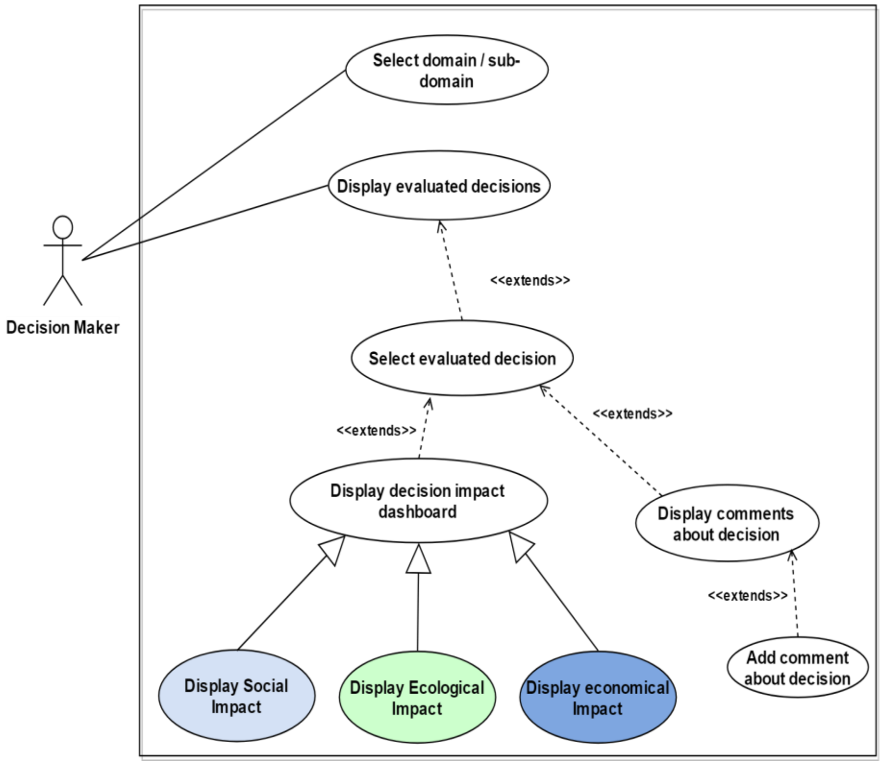

Figure 4.14 Use case diagram - Decision Evaluator component

4.6.3 Component Process Interactions

The flowchart diagram for the decision evaluator component illustrated by figure 4.15 represents the following scenario:

A validated decision can only be evaluated after two conditions are satisfied: (1) it has already been executed and (2) all deadlines of its objectives have been reached (latest deadline date reached). With these being satisfied, the system starts evaluating the decision based on its direct and indirect impact on KPIs, whereby updated KPI values should be retrieved from the DWH and saved in the decision database to accurately calculate the average impact on KPIs. The system separates the calculation of the direct and indirect average impact to apply the importance factors for each type of impact needed to obtain the final rating of the decision (See section "Decision evaluation formulae"). The system automatically assigns a sustainability evaluation to the decision based on its final rating.

The output of the decision evaluation can be accessed by the user (decision-maker) after selecting the domain and sub-domain. The UI displays all of the evaluated decisions classified under the selected sub-domain. The user selects a single evaluated decision to view its detailed impact on KPIs (enhancements, deteriorations) displayed in a dashboard. This dashboard classifies and sorts the KPIs by their sustainability natures (ELPI, ENPI, and SPI) to present a better view of the decision's ecological, economic and social impact separately.

In addition to the automated decision evaluation, the user can further understand its impact and evaluation from other users' perspective by viewing their comments and their numeric evaluation on the selected decision. He can also add his own comment and numeric evaluation to the decision.

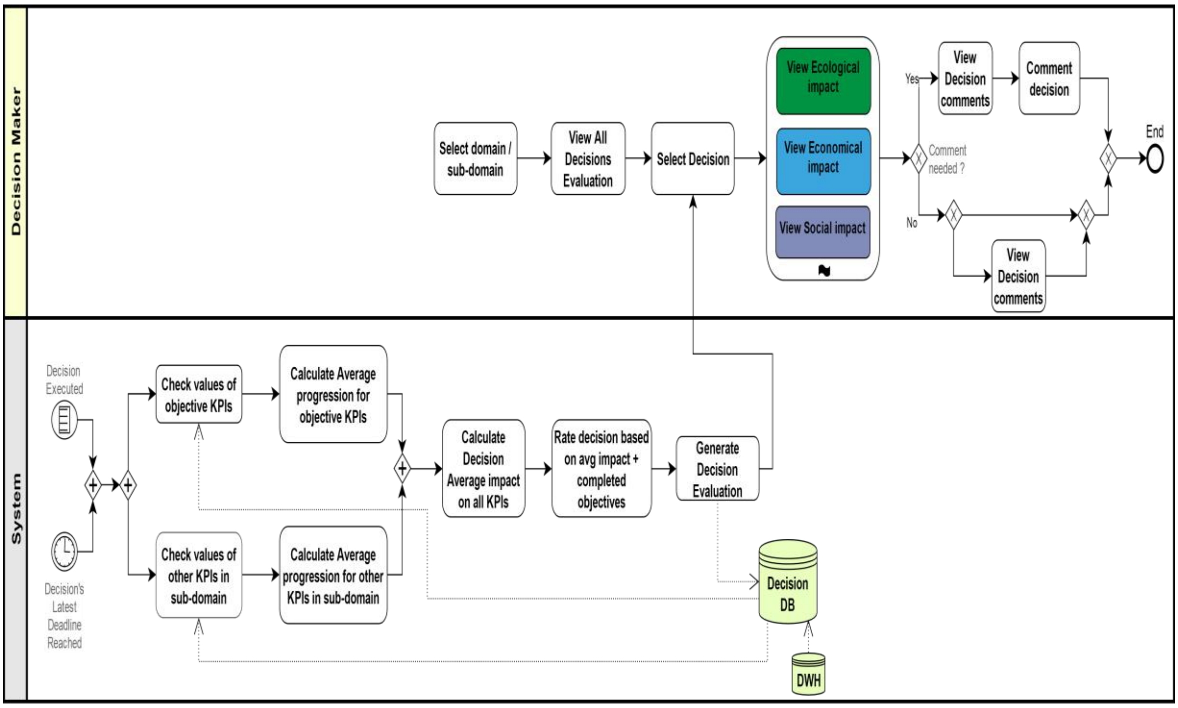

Figure 4.15 Flowchart diagram - decision evaluator component

4.6.4 Component Data Structure

The entities displayed in the global class diagram, their attributes and their relationships are all explained in the global class diagram description, aside from the association class "comment", which is only explained here. It includes the comments that a user can attribute to a decision to state his own evaluation in addition to the automatic system evaluation.

For the decision evaluator component, only one specific relationship between the entities in the global class diagram exists:

The decision-maker (with disregard to his privileges) can attribute one or more comments to one or more decisions. This association is represented by the association class "comment" thoroughly described in table 4.6.

Attributes	Data Type	Attribute Description
Comment date	Date	The date which the comment was written by the author (user). This date is used to identify the comment and enable a single user to write multiple comments on the same decision in different dates.
Comment text	String	The text of the comment typed in by the user to the decision.
Rating	Integer	The decision evaluation given by the user on a scale of 1 to 10.
Attached file	BLOB (Binary Large Object)	The file attached to the comment that will help the user further explain his point of view or provide a proof.

Table 4.6 Class description for the association class "Comment"

4.7 Decision Engine Modelling

The DES – unlike classical EMIS – enables the storage of a decision with its different parameters to grant the tracking of the decision's impact in the future.

To enter and store any given decision in the DES, we need to:

- Set the decision parameters (ID, title, problem description, solution description). This input will be provided by the user.
- Set the objectives of the decision, which is represented by the list of KPIs to be affected, their desired values and the deadline dates for each planned achievement. (Bouyssou et al., 2015) affirmed this question: “On what are we deciding? A formal decision problem needs to fix a set of objects on which to apply a decision procedure”. This is also an input by the user.
- Have transparency in the decision-making process. Transparency – as defined by (Andrew K Schnackenberg and Tomlinson, 2016) – is: "The perceived quality of intentionally shared information from a sender.". In the DES, the transparency of the decision-making process is

granted by defining the decision-makers (and the decision-validator, if needed) and the decision-executor and the dates of creation, validation and execution for all decisions.

- This simple task executed by the system will also give a general view about the decision-maker's perceptivity, intuition and reasoning skills after evaluating his decisions. A good decision-maker will generally make sustainable decisions.

To satisfy these needs, a component named the "decision engine" exists within the DES. The functional requirements for this component are presented in the following section.

4.7.1 Requirements Definition

The "decision engine" component's functional requirements can be classified as follows:

Obtain decision parameters: A simple, understandable UI should be displayed by the user so that he can use the text fields to enter the decision's ID, title and description. It should also be noted that the decision is attributed to only one sub-domain that needs to be displayed for the user to verify the context of his decision.

Obtain the decision goals: Before confirming the decision, the same UI used in the above functional requirement must enable the decision-maker to select at least one KPI that the decision aims to affect. Otherwise, the decision confirmation will not be available as the purpose of any given decision is achieving one or more goals (Turban et al., 2011a).

A checklist of all KPIs in the same sub-domain of the decision is displayed and the decision selects the desired KPIs. For every selected KPI, it is mandatory to enter the objective value and the deadline of achievement.

Define involved actors for the decision: In order to provide transparency for the decision-making process, several actors must be identified for each saved decision. The decision-maker (a single person or a decision committee) assigns a decision-executor for the created decision and enters his parameters (decision-executor full name and e-mail). The system stores the identities of both actors for each decision, namely the decision-maker and the decision-validator. The dates of the decision creation and execution should also be saved.

Validating the decision could be undertaken by the same decision-maker if he is granted this privilege. If this is not the case, the decision is validated by another user with the decision validation privilege referred to as the decision-validator. It is essential that the system identifies and saves the identity of the user who validated the decision in both cases (decision validated by its creator or by another user). The validation date should also be saved.

4.7.2 Use Case Component Interactions

For this component, two types of users interact with the decision engine component, namely the "decision-maker" and the "decision-validator" (this could be single users or decision committees). This last actor can have the same interactions as the decision-maker in addition the privilege to validate his own created decisions. As for other decisions taken by other decision-makers, he can choose whether to validate or reject them. Both of the actors' interactions are represented by the use cases presented in figure 4.16.

Use Case	Description
Actor: Decision-maker	
Create decision	Creating decision using the provided form elements in the GUI includes: • Entering the decision parameters (ID, title, occasion description, solution description). • Entering the parameters of the decision-executor (name and last name, E-mail address). • Assigning objectives to the decision by selecting the KPIs to be affected, their planned values and the deadline dates for achieving those planned values. (Optional) The decision-maker may add a perimeter (Filter(s)) to the objective KPIs. This is to point out that the decision concerns only enhancing the KPI values for a certain perimeter and not on the global scale of the organization (e.g. Enhancing "Service satisfaction rate" for the customers in Region "X" and "Y" instead enhancing the "Service satisfaction rate" for customers from all regions).
Confirm decision	The inputted decision parameters, executor and objectives need to be confirmed. The confirmation invokes: Validating the decision by the same decision-maker if he has the rights to do so.
Actor: Decision-validator	
View created decision	If a decision was created by a decision-maker without the decision validation privilege, a decision-validator need to review this decision's parameters, this includes: Validating or rejecting the decision. (Optional) A decision-validator can view the planned objectives (objective KPIs, planned values, deadlines) for the decision before validating or rejecting a decision.

Table 4.7 Decision Engine Main Use Cases

Table 4.7 provides a brief description of the main use cases between the actors (decision-maker, decision-validator) and the decision engine component and explains the different steps, conditions and extensions of these use cases.

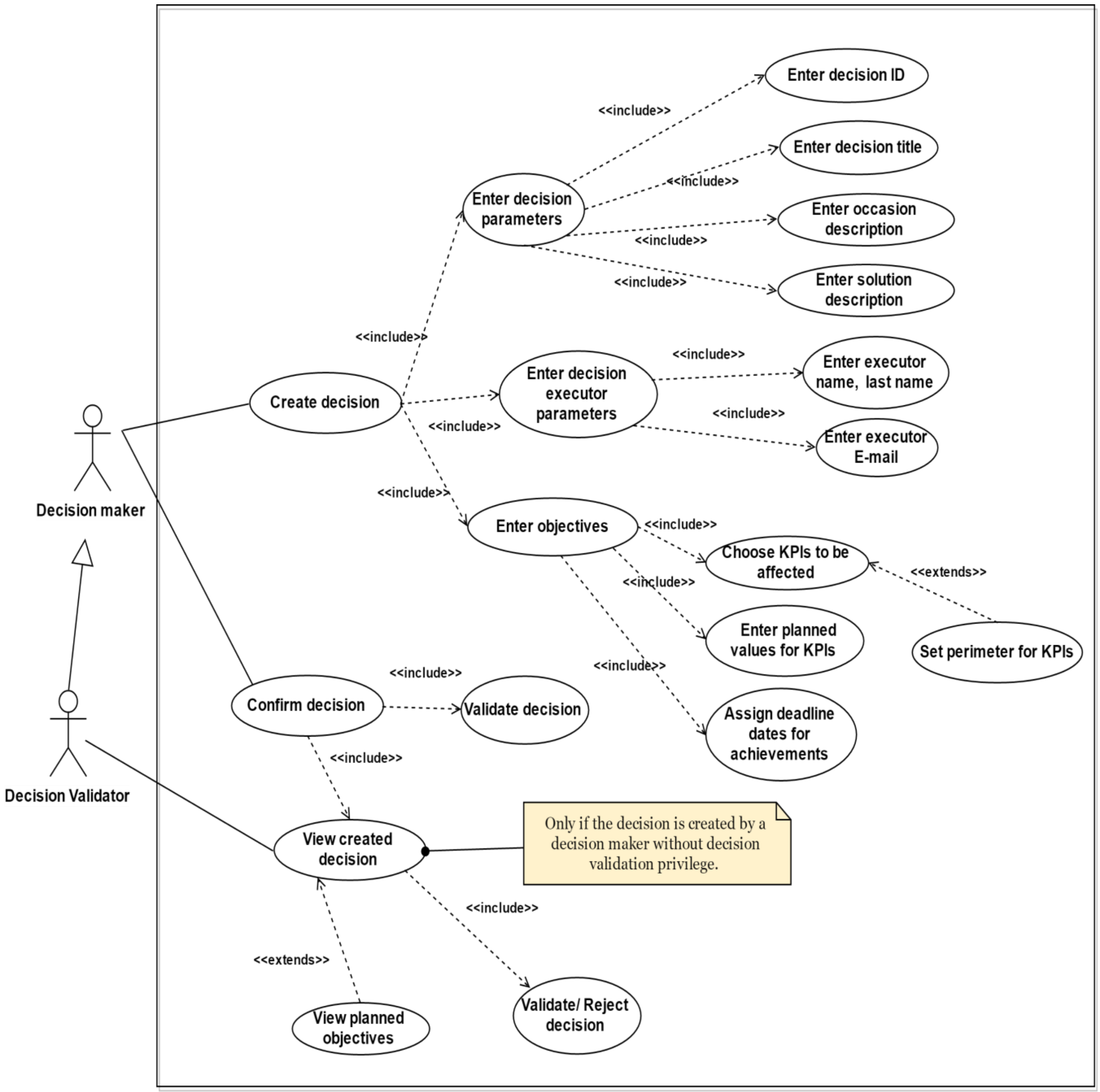

Figure 4.16 Use case diagram for the Decision Engine component

4.7.3 Component Process Interactions

The process of the decision engine will be described in the following cross-functional flowchart diagram in figure 4.17.

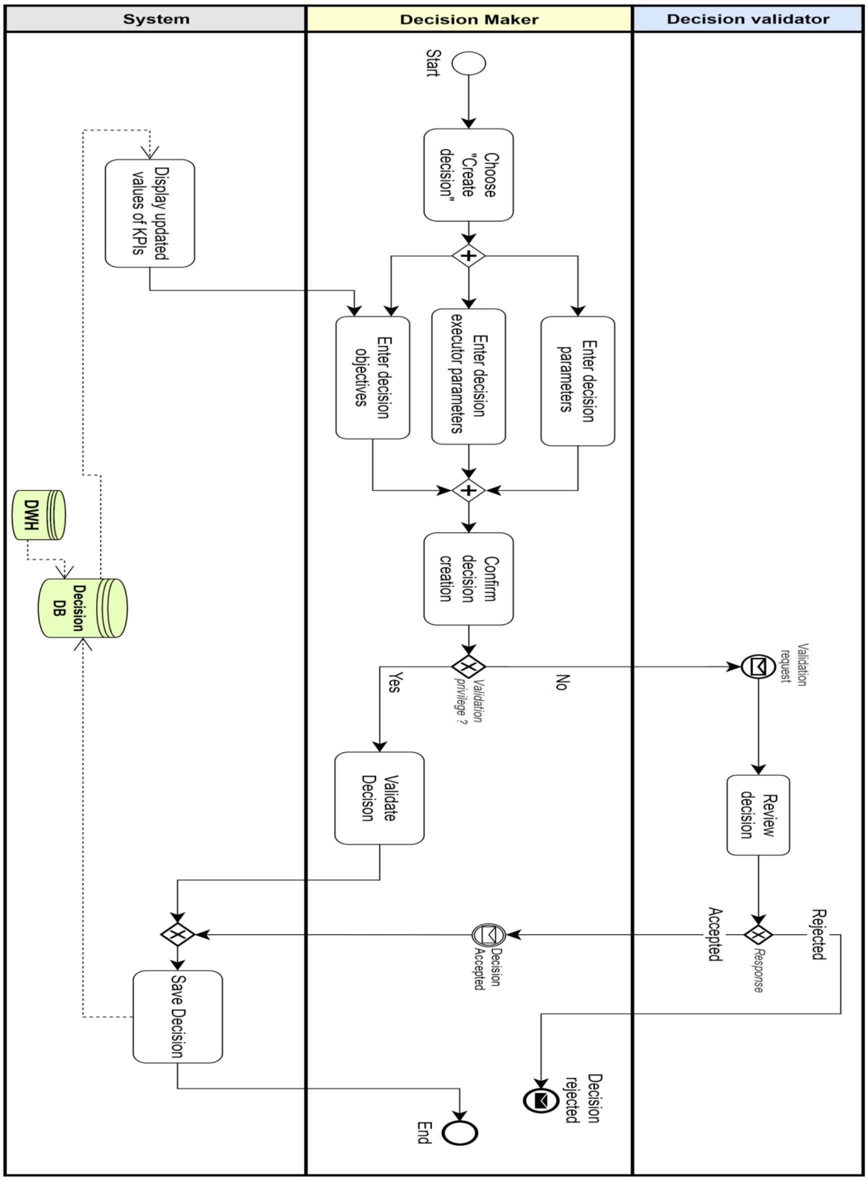

Figure 4.17 Flowchart diagram - Decision Engine component

The flowchart diagram for the decision engine component illustrated by figure 4.17 represents the following scenario:

After identifying an occasion for making a decision (with routine KPI analysis, BI alerts, etc.) and after none of the recommended decisions seemed suitable for the decision-maker, the creation of a new decision is available.

The decision-maker clicks the "create decision" button, whereby a decision creation UI is displayed. Via this interface, the decision-maker must perform three tasks in any order before confirming the decision creation:

- Enter the decision parameters (ID, title, occasion description, solution description).
- Enter the decision-executor parameters (name, last name, e-mail).

- Set the decision objectives (choosing objective KPIs, planned values, objective deadlines). This task requires the system to retrieve the updated values of the KPIs (last recorded value). These values are found in the decision database after being imported from the DWH with the ETL process. Displaying updated KPI values helps the decision-maker to set realistic, analysis-based planned values for the decision objectives.

After finishing the previous tasks, the decision-maker confirms the decision creation. The decision still requires validation before it is stored in the database. If the decision-maker has the decision validation privilege, he can validate his own decision. If this is not the case, it will require validation from a superior user with this privilege referred to as the "decision-validator". This user reviews this decision, he can either validate the decision or reject it.

4.7.4 Component Data Structure

The entities displayed in the class diagram, their attributes and their relationships are all explained in the global class diagram description (see Global Data Structure).

4.8 Decision Recommender Modelling

The main purpose of all recommender systems is generating suggestions about resources that a user a priori is not aware of but would probably be interested in, according to (Jøsang et al., 2013b). The proposed DES also has this feature of recommending decisions (resources) for the user.

This may help the user with the decision-making process, specifically in second step of finding possible courses of actions from the decision-making steps set by (Simon, 1977).

In order to provide a useful and effective decision recommendation system, we need to:

- Set the context for the current decision-making situation.
- E.g. the current situation that needs decision-making concerns which domain of the organization (HR, sales, etc.) or what sub-domain (healthcare, after-sales, etc.)? What are the occasions (or problems) of the current situation that invoked the necessity for making a decision (Simon, 1977)? What are the objectives that the decision-maker wants to achieve? A keyword-based search is also a good addition to retrieve better results while searching for recommended decision.
- Identify and list all of the previous decisions that are related to the selected context (domain, sub-domain, problem, objectives). The similarity between the defined context and the recommended decisions may vary from one decision to another.

- Rank the recommended decisions from best to worst based on their sustainability evaluation and their similarity to the current problem and the decision-maker's objectives.
- Enable the decision-maker to choose from those recommended decisions and adapt them to the current situation by setting new KPIs to enhance as objectives and deadline dates for those objectives. Adapting the decision is equivalent to making a new decision, whereby it will also require validation from the same decision-maker or another user with the validation privilege.

To satisfy these needs, a component named the "decision recommender" is integrated with the proposed DES. The functional, non-functional requirements and the conception of this component as presented in our work “Recommendation of sustainable decisions within a decision evaluation system using case-based reasoning” (Rezgui et al., 2018) are presented in the following sub-sections.

4.8.1 Requirements Definition

The "decision recommender" component's functional requirements can be classified as follows:

Obtain the context parameters: From the available list of domains and sub-domains, the user (who could represent a single person or a decision committee) chooses those related to his decision. He should also input one or more occasion(s) or problem(s) that invoked the decision-making necessity and one or more objective to be attained by the decision. The decision-maker may (optionally) choose indexes from a list to further enhance the recommended decision retrieval. The DES is about making and evaluating sustainable decisions, whereby the three sustainability pillars (society, economy, and environment) can be set as categories for many important topics that we will use them as indexes (scopes) to corner the topics of the sustainable decisions. Figure 4.18 displays some of the scopes that a user can search for in the context definition to be recommended with the appropriate indexed decisions. Examples of the scopes in the figure are: "quality of life", "equal opportunities" and "education" for the social sustainability field or "smart growth" for the economic sustainability field (National Research Council, 2014) (cross-field scopes like "business ethics" and "public involvement" are not included).

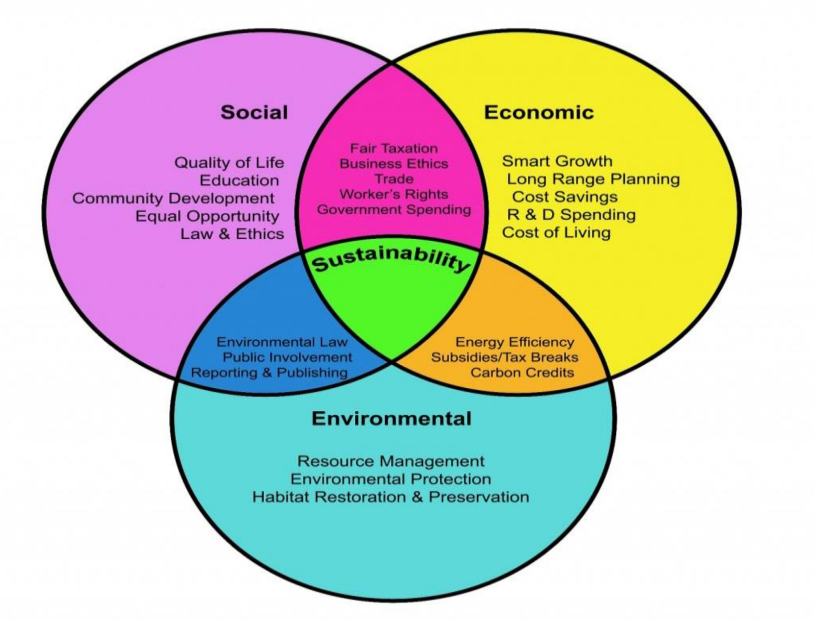

Figure 4.18 Decisions Sustainability Scopes

Retrieve the available related decisions: The system should identify the previous decisions related to the selected context (domain, sub-domain, problem, objectives, and scopes) to be recommended to the user. These decisions are imported from the decision database with their recorded affections on KPIs.

Rank the recommended decisions: To ease the task of choosing the best decision among the available ones, the system should sort the recommended decisions by two criteria: the (1) sustainability evaluation provided by the "decision evaluator" component and (2) the similarity to the current context set by the user.

Adapt recommended decisions: If selected, the recommended decision is not simply "re-made"; rather, it should be adapted to the current situation, vision and goals of the decision-maker. Therefore, the system must enable assigning new objectives and deadlines for the recommended decision. Adapting recommended decisions also include the necessity of validating the decision by the same decision-maker if possible, or another user with the decision validation privilege.

4.8.2 Case-Based Reasoning

The decision recommender component allows recommending decisions to users after defining the context by identifying the occasions (or the problems) and objectives and entering other parameters (domain, sub-domain, scopes). This approach of recommendation is known as CBR in the literature (Aamodt and Plaza, 1994). CBR is a field of AI and it can utilize the specific knowledge of previously-experienced, concrete problem situations (cases) for resolving the current case by applying or adapting the previous solutions (Aamodt and Plaza, 1994; ElKafrawy and Mohamed, 2014), just like an experienced physician identifying the sickness and its remedy for his current patient based on the similarity of the symptoms between the current patient and previous patients.

A case usually denotes a problem situation and its solution (Aamodt and Plaza, 1994; Sarkheyli and Söffker, 2015). By this logic, a previous case in the proposed DES represents the set of problems and objectives and the previous decision that was used to solve it. A current case denotes only the current set of problems and objectives and it is yet to be assigned with a solution (decision) to solve it. This case can be solved by a new or a recommended decision.

Figure 4.19 shows the four-step (retrieve, re-use, revise, retain) process model set by (Aamodt and Plaza, 1994; ElKafrawy and Mohamed, 2014) and found in typical CBR systems, whereby this process model is still the most commonly used for modern CBR libraries and software like myCBR[7], FreeCBR[8], jCOLIBRI[9] and eXiTCBR[10] (ElKafrawy and Mohamed, 2014). The description of the four steps is presented below with their relationships with the proposed DES.

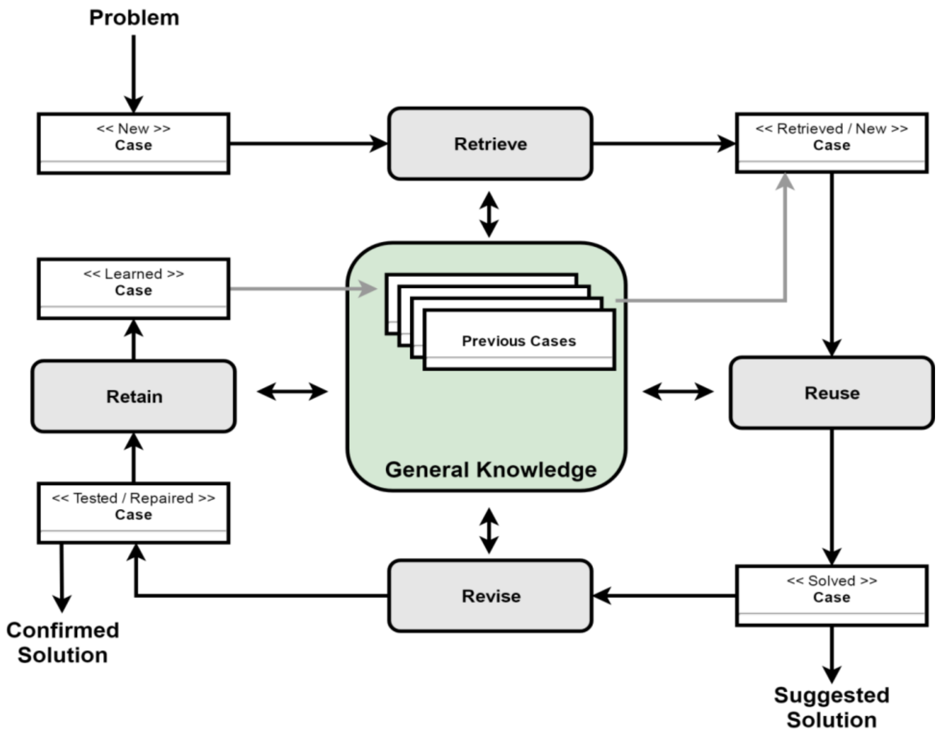

Figure 4.19 Case Based Reasoning Model

Based on (Aamodt and Plaza, 1994)

[7] www.mycbr-project.net/
[8] www.freecbr.sourceforge.net/
[9] www.gaia.fdi.ucm.es/research/colibri/jcolibri/
[10] www.exitcbr.udg.edu/

(Roth-Berghofer, 2003) highlighted a set of "shortcomings" of this four-step process model. He stated that this model lacks the distinction between the steps that apply changes to the knowledge database (maintenance) and the steps that do not apply such changes. He also stated that Aamodt and Plaza did not foresee the maintenance of the knowledge base and their model does not support storing maintenance data and introspection on system changes. To answer to these shortcomings, he proposes a six-step process model with two novel steps ("review", "restore") and a classification of the steps based on whether they apply changes to the cases knowledge database. Nevertheless, most of the CBR systems use the four-step model and do not support maintainable cases like in the six-step model.

The CBR applied in the proposed DES has the four R's processes, just like most CBR algorithms.

These four processes applied to the cases are executed across multiple components of the DES but their output is shown by the case retrieval shown in this component (decision recommender). The CBR processes in the DES and their relationship with the DES components are explained as follows:

Retrieve case: After defining a new case (defining a context). Similar (or rather similar) previous cases are retrieved from the knowledge database, these previous cases are constructed from the contexts (problems, objectives, scopes) and their evaluated solutions (decisions). The cases are retrieved in this component, the decision recommender, they are displayed in the UI along with their similarity percentage to the entered context parameters (problems description, objectives description, selected scopes).

Re-use case: This step concerns re-using the information and knowledge in a previous (learned) case to solve the current problem. Recommended decision can be re-used (adapted) with the current case context by assigning new KPI objectives (or keeping the previous ones) and extending the deadlines. It should be noted that a new context and a new solution (thus a new case) are established while creating a new decision with the decision engine component. Both newly-established and re-used cases will be stored and evaluated for eventual retrieval.

Revise case: A "solved case" is obtained after adapting a previous case or using a new solution to the problem. This solved case need to be evaluated after being applied in the real environment. In the DES, this process is achieved by evaluating the newly-created and the adapted decisions with the decision evaluator component. The recommended decisions will be ranked based on this evaluation in addition to the similarity with the current case.

Retain case: The researchers define this step as the process of incorporating what is useful to retain from the new problem-solving episode into the existing knowledge, which involves eliminating the poorly-evaluated cases. However, in the DES, regardless of whether the decision used to solve a case had a poor or good evaluation, the case will be retained. We assume that a decision – even poorly evaluated – can perform differently with alternate periods of time, conditions, new situations and new set objectives. Therefore, poorly-evaluated decisions also need to be retained, whereby it can be adapted with the current case and possibly perform better with a new context or period of time.

Cases may be indexed by a pre-fixed or open vocabulary, and within a flat or hierarchical index structure (Aamodt and Plaza, 1994; Sarkheyli and Söffker, 2015). The cases in the DES are indexed with a flat hierarchical index structure. This structure is represented by the scopes (children) that are mapped with their respective sustainability topics (parents). Those scopes that the proposed DES uses to index the cases were shown in the previous section.

Figure 4.20 presents the CBR process model in the proposed DES.

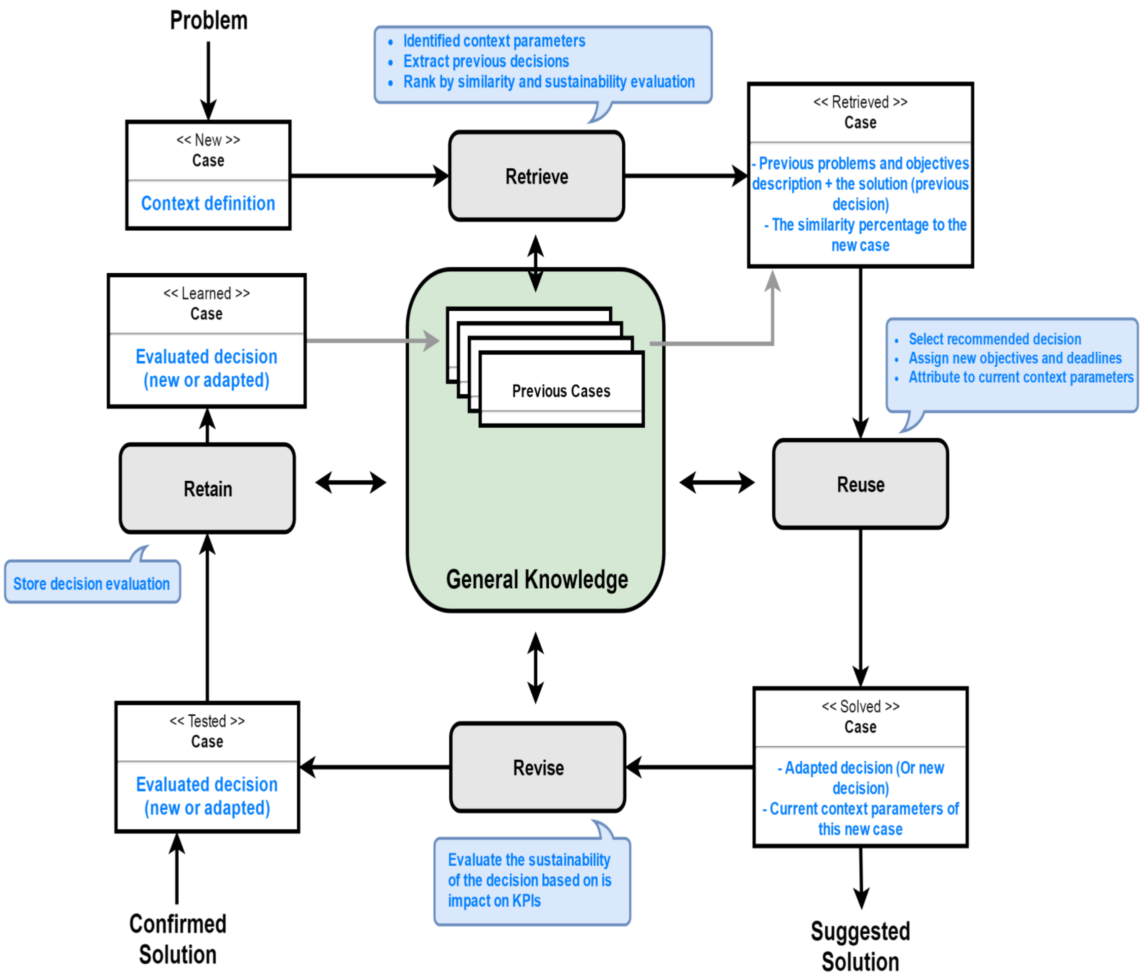

Figure 4.20 CBR Process model in the proposed DES

Source (Rezgui et al., 2018)

4.8.3 Use Case

Two types of actors interact with the decision recommender component, namely the "decision-maker" and the "decision-validator" (which could be single users or decision committees). This latter actor can have the same interactions as the decision-maker in addition to the privilege of validating the created or adapted decisions (his decision or other users') or rejecting them. Both actors' interactions are represented by the use cases presented in figure 4.21.

Just like the process of the decision creation within the decision engine component, the system will save the parameters of the decision-maker (and decision-validator, if such exists) and the decision-making date for transparency requirements in the decision-making process.

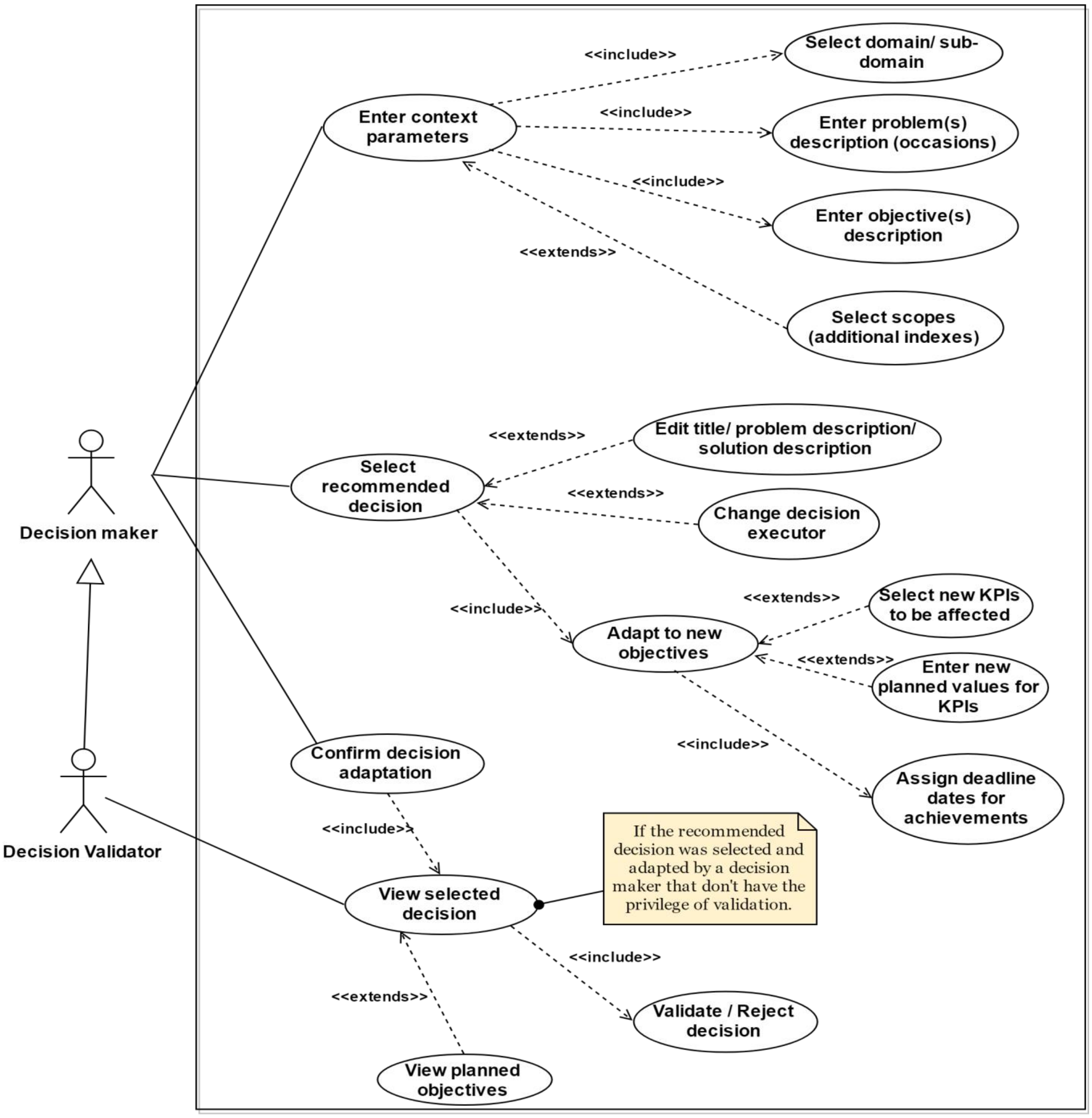

Figure 4.21 Use case diagram - Decision Recommender component

The following table provides a brief description of the main use cases between the actors and the decision recommender component and explains the different steps, conditions and extensions of these use cases.

<table>
<tr><th>Use Case</th><th>Description</th></tr>
<tr><td colspan="2">Actor: Decision-maker</td></tr>
<tr><td>Enter context parameters</td><td>The decision-maker inputs the context parameters which means entering mandatory parameters (Domain, Sub-domain, Problems description, Objectives description) and optional parameters (scopes).</td></tr>
<tr><td>Select recommended decision</td><td>From the list of the recommended decisions, a decision-maker may choose one decision to adapt it to his current case, this includes:
Adapting it to the new objectives by selecting new KPIs as objectives, setting their planned values and assign new deadlines. (Or just setting new deadlines and keeping the same previous KPIs objectives).
(Optional) The parameters of the adapted decision (title, objectives description, problem description) and the decision-executor can be edited by the decision-maker to be more convenient to his current case.</td></tr>
<tr><td colspan="2">Actor: Decision-maker & Decision-validator</td></tr>
<tr><td>Confirm decision adaption</td><td>The decision-maker confirms his adaption of the recommended decision by successfully submitting its parameters in the UI. But this will involve validating the decision before starting to record its impact on KPIs:
Validating the decision can only be by a user with decision validation privileges so any decision-maker with this privilege can validate his own decision. Otherwise, another superior user (decision-validator) may view the decision parameters and its planned objectives than chooses between validating and rejecting the decision.
A rejected decision will not be deleted and can be viewed later along with the identity of the user who rejected it and the rejection reason (if provided by the user).</td></tr>
</table>

Table 4.8 Decision Recommender Main Use Cases

4.8.4 Component Process Interactions

The cross-functional flowchart diagram for the decision recommender component is illustrated by figure 4.22.

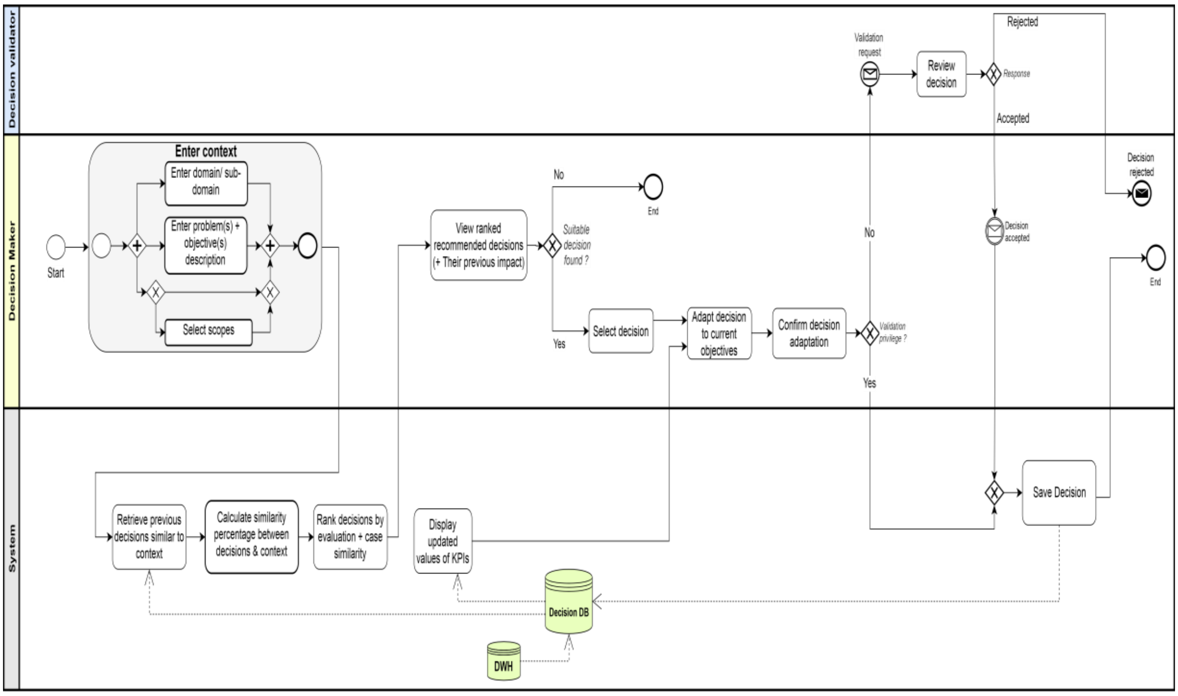

Figure 4.22 Flowchart diagram - Decision Recommender component

It represents the following scenario:

A decision-maker identifies an occasion or problem for making a decision (Simon, 1977). In the DES-provided UI for the decision-making, a decision-maker must first enter some parameters about the current context (current case) of the decision-making. These parameters include the domain/sub-domain of the decision, the problem(s) and objective(s) description, the scopes used as keywords (optional). These parameters will help the DES to retrieve similar cases with some or most of the entered parameters, each previous retrieved case was solved with a single decision that will be recommended to the user here. The system calculates the similarity percentages between the retrieved cases (decisions) and the current context parameters, whereby these values will be shown for each recommended decision. The system then uses the similarity percentage and the sustainability evaluation for each decision to rank (sort) them in a descending order.

The decision-maker views the ranked recommended decisions in the UI and can view the previous impact on KPIs for each one. Should none of the decisions are suitable to take, he can proceed to creating a new decision and the process of the decision recommendation ends. Otherwise, he can select a recommended decision and adapt it to the current context by:

- Re-assigning KPI objectives (optional).
- Extending deadlines (mandatory).
- Editing its parameters (title, objectives description, problem description, and decision-executor) (optional).

After entering and configuring the new parameters of the decision, the decision-maker confirms the adaption. The decision still requires validation. If the decision-maker has the decision validation privilege, he can validate his own. If this is not the case, it will require validation from a user with this privilege referred to as the "decision-validator". This user reviews this decision, he can either validate the decision or reject it.

The entities displayed in the class diagram, their attributes and their relationships are all explained in the global class diagram description.

4.9 Data Management Modelling

Data management modelling represents the analysis and design of the information contained in the DES. This section identifies the logical entities and the dependencies between them.

4.9.1 Data Storage

In the proposed DES, the data management will be assured through managing two main databases, namely the data warehouse and the decision database**.**

The DWH will be used to retrieve the KPI values over time to track their enhancements and deteriorations. The attributes (ID, name, description, measure unit, etc.) of those KPIs are given as an input by the user and stored in the decision database before their mapping to the physical columns in the DWH using the process of ETL. The information stored in decision database is categorized into:

- System information:
    - Users (ID, name, e-mail, address, occupation).
    - Their authorizations (privileges, responsibilities).
- Decision information:
    - KPIs (ID, name, description, measure unit, desired variation, sustainability nature).
    - Decisions (ID, title, description, creation date, validation date, execution date, sustainability evaluation, final rating), their objectives (planned values for KPIs, deadlines) and their actual impact (enhancement or deterioration, percentage of enhancement / deterioration) for each KPI that was set as objective or not.
    - Domains and sub-domains (ID, title, description) for all the KPIs and decisions.

Based on the general system requirements and the specific requirements for all of its components, the requirements for the database system are defined and presented in the next section.

4.9.2 Requirements Definition

The database system functional requirements can be defined as follows:

Verify user input: For the authentication, the input (login, password) is compared with the stored information in the database of users to allow only authorized access to the DES. In case of conflict between the input and stored date, the access is denied. The usernames and passwords are stored and retrieved from the decision database.

Manage user data: The DES administrators must be able to manage the user list of the system by adding new users, editing their parameters, granting and depriving privileges. This required user-specific data of the DES is stored and retrieved from the decision database.

Store data: All users' inputs (KPIs, domains, sub-domains, decisions) must be stored persistently in the decision database. Some of existing stored data can be overwritten by new inputs when the user chooses to re-configure them, which includes the parameters of KPIs, domains and sub-domains and excludes the decision parameters. The data storage concerns only the decision database.

Retrieve requested data: The data retrieval concerns the DWH and the decision database. Executing an operation within the DES sometimes requires operation-specific data retrieval from both databases. The KPI values and changes over time are retrieved from the DWH to help setting decision objectives and to be displayed in decision evaluation and impact dashboards.

The KPI parameters, their domains/sub-domains and the previous decisions are retrieved from the decision database, which is required for the decision creation, evaluation and recommendation.

Delete data: Existing data in the decision database can be deleted by users who have authorized permissions. System administrators can delete users and deprive them from using the DES or delete some privileges for other users. Decision-makers can delete KPIs, domains and sub-domains from the configuration panel in the DES.

ETL Mapping: The dimensions and fact tables in the DWH (source) should be mapped to the KPIs table in the decision database (target) from time to time for a defined period. Regular, correct and consistent ETL mapping is essential to obtain accurate decision evaluation, since the latter is based on the average impact on KPIs, the impact calculation should be based on updated and correct KPI values.

5 Decision Evaluation System: Software Architecture

In this chapter, the common language, standards, specification and design for the validation of the proposed solution will be explained through presenting the software architecture. Based upon (Bass et al., 2003), we can define the software architecture as "the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them". After presenting the overall architecture in section 5.1, the five components: front-end, KPI & decision tuner, decision evaluator, decision engine and decision recommender will be explained. The UML component diagram was used for the software architecture description.

It is very hard for software engineering researchers to agree about the nature of the relationship between requirements engineering and software architecting (Liu and Mei, 2003). Following the development model, software architects should adapt the way they design the architecture because the impact goes both ways: the architecture can be an obstacle to designers to meet some requirements, and the requirements can influence the architecture that designers select or develop.

The functional and non-functional requirements, presented in the previews section should establish the foundation of the overall architecture for the DES. For that, a clear mapping between the requirements and the different components of the architecture is needed. It gives an improved traceability and consistency in concepts between requirements and designs. Besides the technical implementation issues that may arise due to a required software change, the semantic differences between process-oriented business needs on the one hand, and mostly object-oriented software architecture views on the other hand, remains a major software engineering challenge (Jelschen et al., 2016). Therefore, a sustainable development of a platform and deployment of a software solution requires a successful transformation of information between the disciplines. This interplay between the problem (requirements) and the solution (architecture) is at the heart of any engineering design activity (Chen et al. 2013).

The requirements can be mapped as follow:

- The first functional requirement (FR1) "Allowing the participation in decision-making process": is meant to have the decision-makers store their decisions and get evaluation and feedback on them. This requirement is mapped to the front end and the decision database components because the process of making a decision includes reviewing past decisions and an evaluating them according to the sustainability criteria and also according to the impact on business profitability.

- The second functional requirement (FR2) “Organizing access and privileges”: is also mapped to the front end component via the authentication form. It allows users to access the application and get the necessary privileges and authorizations.
- The third functional requirement (FR3) “Enabling users to enter/configure the system parameters” is mapped to more than one component. It uses the front end to enable the users to enter their configurations and it stores them into the decision database.
- The fourth functional requirement (FR4) “Enabling insertion of decisions”: is mapped to the decision database via the front end.
- The fifth functional requirement (FR5) “Enabling decision validation”: is mapped to the user feedback unit in the decision evaluator component.
- The sixth functional requirement (FR6) “Tracking the decisions”: is mapped to more than one component. It is mapped to the decision database since all historically made decision are stored there. But also to the decision evaluator, and more precisely the decision impact monitor.
- The seventh functional requirement (FR7) “Evaluating decisions”: is mapped to the decision evaluator component. The evaluation is done manually throughout the user feedback unit or automatically throughout automatic evaluation generator.
- The eighth functional requirement (FR8) “Enabling comments and user evaluation on decisions”: is mapped to the user feedback unit. This unit acts like an enabler for group decision-making and the users allowed to evaluate decisions should have a high level of experience and expertise.
- The ninth functional requirement (FR9) “Recommending decisions”: is mapped to the recommendation unit under the decision recommender component. This unit gives a score for every decision and recommends the highest ranked decision to be used for similar circumstances.
- The tenth functional requirement (FR10) “Having transparency in the decision-making process”: is mapped to the front end because the user can access all ratings and recommendations.
- The last functional requirement (FR11) “Being a sustainability-friendly (green) software”: is considered by the design and development of the DES visa consumption minimization of the machine's resources (RAM utilization, disk space, CPU utilization) and minimal data flow from/to data sources. This will reduce the energy consumption of the hosting machine and the database servers.

5.1 Overall Architecture

There are five collaborating functional components that construct the DES:

Front-end: This component comprises the different web-based interfaces with which the user (administrator, decision-maker or decision-validator) interacts with the system. These interfaces allow the users to:

- Login/logout.
- Manage the users' access and privileges (for administrators).
- Enter/update the parameters that the system will manage (KPIs, domains, sub-domains).
- Generate/update/display the classification of KPIs by domains and sub-domains.
- Enter context parameters (domains, sub-domains, occasion description, objective description, scopes) before making or choosing a decision.
- Display/choose recommended decisions when users identify occasions for making decisions and find decisions related to their current context (current case) and well-evaluated.
- Create/adapt/validate a decision (newly-created or recommended) with assigning its KPIs objectives and entering/modifying its parameters. A decision-executor must be assigned for each decision.
- Display the sustainability evaluation of the previous decisions and their impact on different KPIs.
- View/add comments and attached files to the evaluated decisions. Users may also attribute an evaluation (on a scale of 1 to 10) to the decision.
- KPI & Decision Tuner: This component allows the input and storage of the system parameters that other system functionalities will require. It is responsible for:
    - Obtaining and storing the parameters of KPIs, the domains and sub-domains provided as an input by the user.
    - Obtaining the user's classification of the KPIs by sustainability nature (social, ecological, economical) and the domains/sub-domains of the organization. This classification is stored in the decision database and displayed in a matrix called the "KPI matrix".
    - Monitoring the KPIs values for enhancements or deteriorations.

Decision evaluator: This component tracks and evaluates previous decisions stored in the Decision database. This can be undertaken in the following steps:

- Monitoring the decision's direct and indirect impact on the KPIs in the same sub-domain for enhancements and deteriorations. A decision's impact is calculated between its execution date and its last objective deadline date.
- Calculating the decision's final rating. This rating is based on the average percentage of the enhancement or deterioration of all KPIs with direct and indirect relationships and the distance between their current and planned values.
- Providing an automatic evaluation for the decision based on the decision rating. A decision evaluation can vary from "(1) very unsustainable" to "(5) very sustainable".
- Displaying a decision evaluation dashboard in the GUI showing the evaluation of the decision and its impact on the different classes of KPIs, namely SPIs, ELPIs and ENPIs.
- Managing the user feedback on evaluated decisions (comments, users' evaluation).

Decision engine: This component enables the input and storage of new decisions in the decision database. It is responsible for:

- Obtaining and storing the decision parameters provided as an input by the users. The decision parameters include the title, occasion (problem) description and solution description. The decision-executor parameters (full name, e-mail) are also included.
- Obtaining and storing the decision objectives. It is represented by the list of KPIs to be affected, their planned values and the deadline dates for each planned objective.
- Defining the decision-maker, validator and executor for each decision along with the creation date, validation date and execution date. This information is stored for each decision for transparency requirements in the decision-making process in an organization.

Decision recommender: This component allows recommending decisions to users after identifying their situation (case), which comprises the occasion or the problem and the objectives. This recommendation approach is referred to as CbR in the literature.

CbR can utilize the specific knowledge of previously-experienced, concrete problem situations (cases) for resolving the current case by applying or adapting the previous solutions. By this logic, a previous case in the proposed DES represents the set of problems and objectives and the previous decision that was used to solve it. A current case denotes only the current set of problems and objectives and it is yet to be assigned with a solution (decision) to solve it. This case can be solved by a new or recommended decision.

The CbR paradigm for problem-solving is integrated in the different components of the DES, although its main functionalities are within the decision recommender component. This component is responsible for:

- Obtaining the context parameters that will be used to retrieve recommended decisions. The user defines the context of the decision by selecting the concerned domain and sub-domain, inputting the desired objective(s), the occasion (or problem) description and the scopes. The scopes serve as keywords to further index the cases in the knowledge base and helps for a better case retrieval.
- Discovering the decisions related to the defined context. From the previous similar cases, the decision recommender identifies the previous solutions (decision) for those cases and extracts them from the decision database.
- Calculating the similarity percentage between the recommended decisions' and the current context parameters.
- Ranking the decisions by two criteria: their (1) sustainability evaluation provided by the "decision evaluator" component and (2) their similarity percentage to the defined context.
- Displaying the recommended decisions in a sorted list and enabling the user to choose a decision from this list to view its previous impact and/or adapt it to its current situation. The decision is adapted by setting it new objectives or extending the deadlines of its current objectives, its title, description and executor can also be edited.

The following figure presents the components diagram for the DES and their inter-collaboration and shared resources.

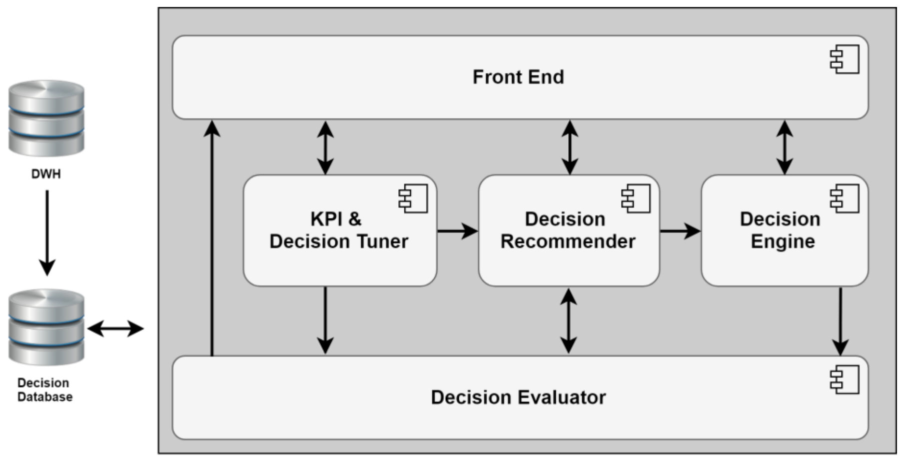

Figure 5.1 Decision Evaluation System Architecture as component diagram

Based on (Rezgui and Marx Gómez, 2017)

5.2 KPI & Decision Tuner Component

The following figure presents the architecture of the KPI & Decision Tuner component. The DWH will be used as the source to obtain the list of KPIs, their previous and current values with date stamps (to understand the enhancement or deterioration) over time, whereby these values will be stored in the decision database. The decision database will be used to store the domains, sub-domains and later to store the KPI classification by nature (economical, ecological, and social) and by domains/sub-domains. A DAO will be needed to abstract the data source's client interface from its data access mechanism.

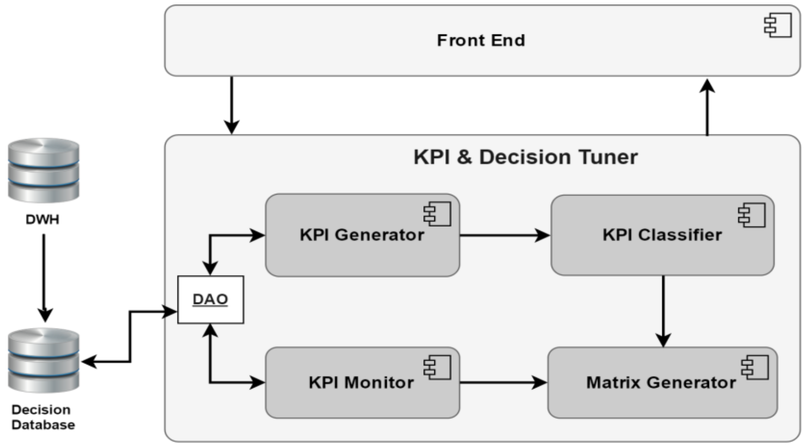

Figure 5.2 KPI & Decision Tuner components diagram

Source (Rezgui and Marx Gómez, 2016b)

The four collaborating, functional sub-components of the KPI & Decision Tuner are:

- **KPI generator**: This sub-component obtains the parameters of KPIs, the domains and sub-domains from the user input and stores them in the decision database. The same sub-component enables importing the KPI values from the DWH (with the help of ETL mapping between the KPI values tables in the DDB and the DWH).
- **KPI classifier**: This sub-component enables users to the KPIs by domains/sub-domains and sustainability natures and stores this classification in the decision database.
- **KPI monitor**: This sub-component monitors the KPI values for enhancements or deteriorations after those values are imported into the decision database.
- **Matrix Generator**: This creates a matrix called the KPI matrix, which is the main output of the system. It shows the KPIs in relation with their respective sustainability fields, domains/sub-domains.

5.3 Decision Evaluator Component

There are three functional sub-components of the decision evaluator component **(See figure 5.3):**

- **Decision impact monitor**: Has the role of tracking the changes (enhancement, deterioration) occurring to the KPIs values between the two dates of the decision execution and the deadlines assigned to the different KPIs. These values are imported from the DWH with the ETL process and saved into the decision database.
- **Rating unit**: This sub-component should learn about the KPI enhancements and deteriorations before providing a rating for the decision's sustainability based on the KPI values before and after the decision. This sub-component rates the decision based on their positive and negative impact on KPIs as well as meeting the pre-set objectives. This evaluation will be then stored in the decision database.
- **Automatic evaluation generator**: This sub-component is responsible for generating the decision impact & automatic sustainability evaluation dashboard in a GUI. The evaluation will be based on the impact of the decision on KPIs (direct and indirect). The interface will show – for a selected decision – the impact on the social, ecological and economic performance indicators.
- **Users feedback unit**: This sub-component is responsible for providing a UI for each evaluated decision that displays the previous comments of other users. It also enables the current decision-maker to write his own comment on a selected decision.

  The user can also evaluate the decision and give it a rating on a scale of 1 to 10. This scale is larger than the scale used in the automatic decision evaluation (1 to 5) because it will be used by a person (decision-makers). Researchers in Primary Intelligence[11] claim that this 10-point scale provides better variability and differentiation to users and have worked best form them in their experience (Primary Intelligence in Excellent research methodologies, 2017).

[11] B2B marketing and sales solutions provider: https://www.primary-intel.com

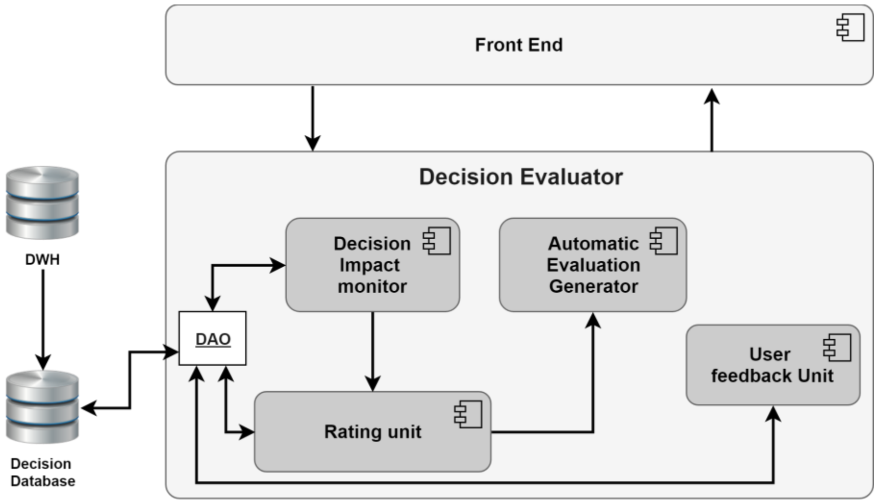

Figure 5.3 Decision Evaluator components diagram

5.4 Decision Engine Component

There are three functional sub-components of the decision engine component:

- **Decision tuner**: This sub-component is responsible for enabling the user to enter the decision parameters (ID, title, occasion description, solution description) and storing these parameters in the decision database.
- **Goals tuner**: This sub-component is responsible for showing the updated values of KPIs to be displayed in a list, whereby these updated values are retrieved from the DWH and stored with date stamps in the decision database. This sub-component will enable the user to select the objective KPIs from the list of KPIs, their planned values and their deadlines. These parameters are then stored in the decision database.
- **Decision actors tuner**: This sub-component will identify the decision-maker, the decision-validator and the decision-executor for each decision and save this information in the decision database. The dates of creation, validation and execution will also be stored. This will help providing more transparency in the decision-making process and evaluating the decision-makers and executors based on the sustainability evaluation of the decisions they take part in.

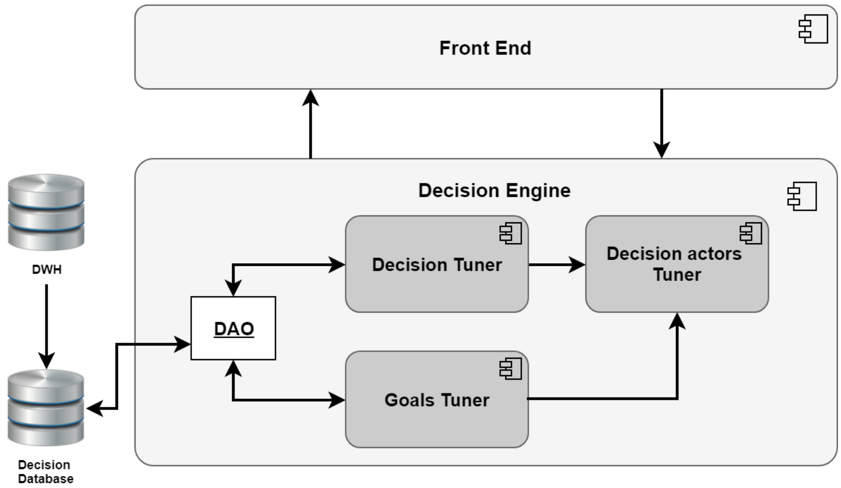

Figure 5.4 Decision Engine components diagram

5.5 Decision Recommender Component

There are three functional sub-components of the decision recommender component:

- **Context generator**: This sub-component – as the name suggests – will generate the context from the parameters entered by the user (domain, sub-domain, problem description, objective description, scopes), whereby these parameters define what is called the "current case" in the CBR. These parameters defining the current context will be used to retrieve previous decisions that have similar previous context parameters.

- **Similarity calculator**: This component calculates the similarity percentage between a retrieved case (decision) and the current case (current context parameters). The similarity percentage will be shown in the UI for each of the displayed recommended decisions and it is based on four similarity measurements:

  - Textual similarity of the objectives descriptions.
  - Textual similarity of the problems descriptions.
  - Exact similarity of the selected scopes.
  - Exact similarity of the sub-domain of the decision.

  The textual similarity between the strings is calculated based on the Levenshtein distance (LD) algorithm (Gilleland, 2009; Sulzberger, 2017). This algorithm is described in the implementation document.

- **Ranking unit**: This sub-component will sort the retrieved decisions based on two criteria: (1) the sustainability evaluation and (2) its similarity to the current context (current case). This sort is optional and the user may change the sorting method to only one criterion in the UI (e.g. sorting decisions only by their sustainability evaluations).

- **Recommendation unit**: This sub-component is responsible for displaying the ranked decisions to the user in the GUI and displaying the previous impacts on KPIs for each recommended decision. This UI will also enable the user to select a recommended decision and adapt it to the current situation. The process of adapting recommended decisions is similar to the process of the decision engine component.

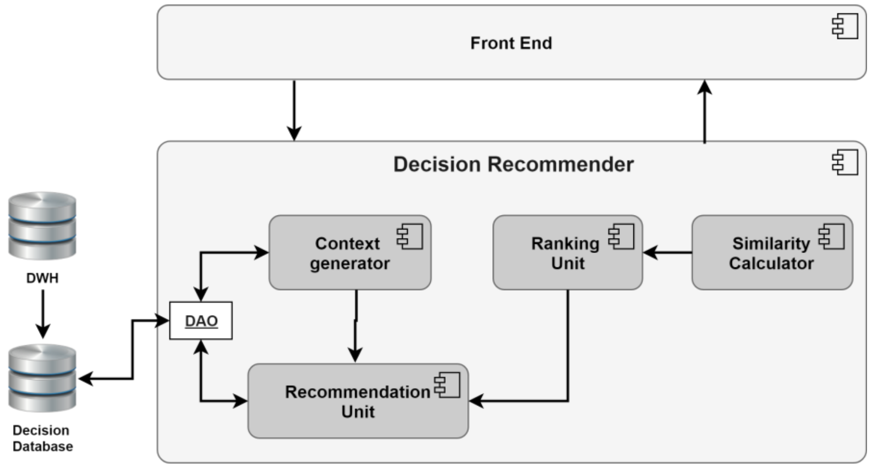

Figure 5.5 Decision Recommender components diagram

Source (Rezgui et al., 2018)

6 Implementation and Evaluation: Proof of Concept

The development of the DES (the software) was conducted to meet guideline: design as artifact (Hevner et al., 2004). Additionally to demonstrate the application as proof of concept in the DSRM by (Peffers et al., 2007).

One of the important factors to define the quality of a software is its documentation. (Forward, 2002) defines such a deliverable as “an artifact whose purpose is to communicate information about the software system to which it belongs”. Successful software documentation can increase the level of confidence of the end deliverable as well as enhancing and ensuring a product’s success through its usability, marketability and ease of support (Kipyegen et al., 2013).

This chapter concerns enhancing the level of confidence between the end deliverable, the transparency and the ease of review and configuration for other software developers. It presents the technologies used to create the DES software, the classes, their operations and attributes used for its construction. The chapter ends with a walkthrough of this developed web application by presenting some of the screenshots presenting its main functionalities in the form of a business case.

6.1 Prototypical Implementation Landscape

The software architecture of the DES that was developed with Java Platform Enterprise Edition (J2EE) is not significantly different from typical J2EE applications that comprise three basic layers, represented by the involved machines (client machine, J2EE server machine, database server machine) or the MVC typical architecture (model, view, controller).

However, there is a difference in the layered architecture by introducing an additional component in the persistence layer that contains the DAOs and the persistent classes, which are the equivalent of the model in the MVC architecture. The functionalities of this layer are within the J2EE server machine and it communicates with the database machine using the implemented JDBC driver (parameters of the communication are the machine address, username, and password).

Figure 6.1 presents a high-level architecture of the DES.

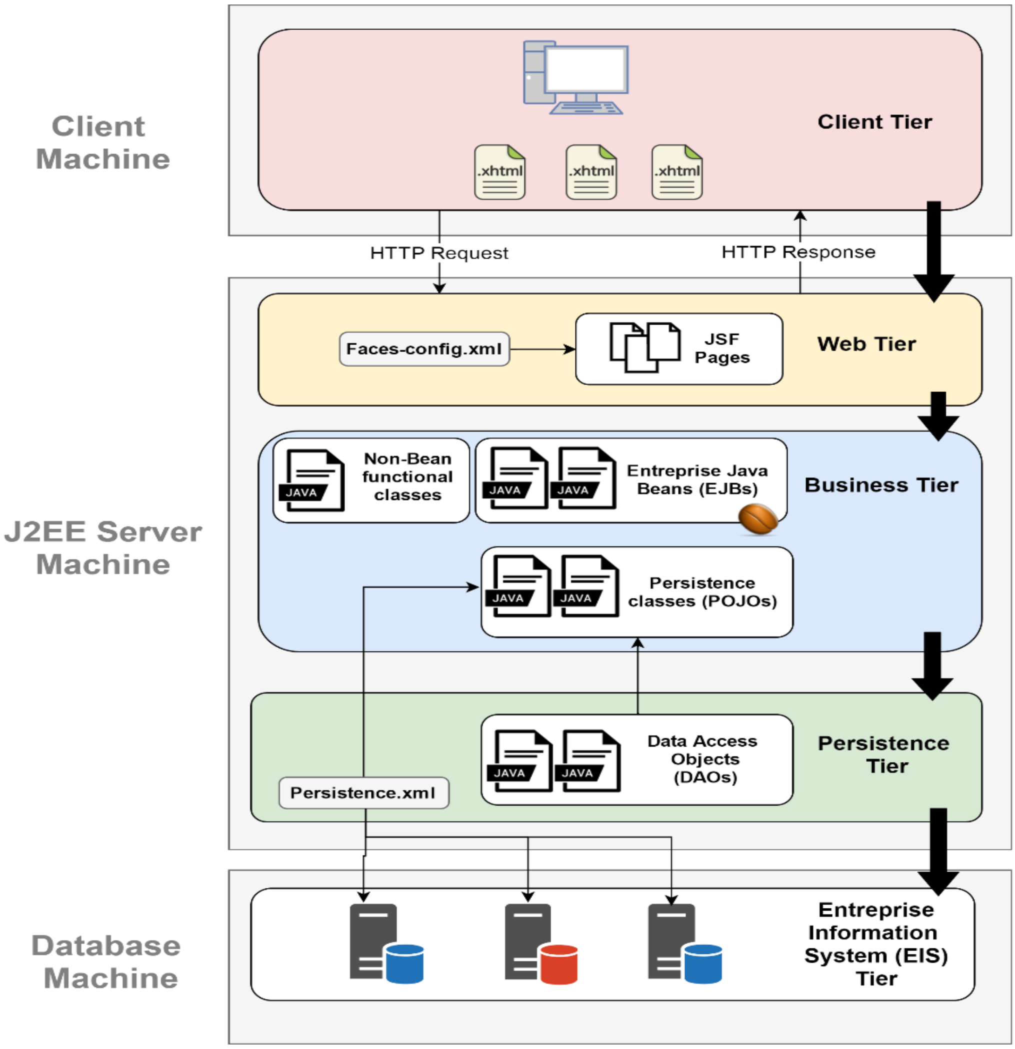

Figure 6.1 DES Software Architecture

In figure 6.1, a multi-tier architecture is shown running on three machines:

- **The client machine**: Contains the client tier. It comprises application logic accessed directly by an end user through a UI. The logic in the client tier could include browser-based clients in this case. The logic could include also Java components running on a desktop computer, or Java 2 Platform (Oracle, 2014; Sun Microsystems, 2004). The mark-up language to construct the shown UI interfaces are plain HTML and/or XHTML. The user cannot have access to see the real JSF components that generated the page.
- **The J2EE server machine (Application server)**: Contains three layers (tiers):

  Web Tier: Where the web components are presented in the J2EE servers, those components are not plain HTML or XHTML, they are rather JSF-based and Primefaces-based components that generate plain HTML or XHTML codes. Multiple types of scripts (Javascript…) can also be shown here to provide functionalities that will apply in the web applications. The "faces-config.xml" is a configuration file that can be used to organize the navigation between web pages and define some of the managed beans without the need of implementing them. (E.g. using the “Java.util.date” as a managed bean to get current dates in the JSF page).

Business Tier: The business service tier consists of logic that performs the main functions of the application: processing data, implementing business rules, coordinating multiple users and managing resources. In JSF-based applications, the functional classes are written in Java and most of these classes are Enterprise Java Beans (EJBs), alternatively called “Managed beans”. The attributes and the operations of those beans be accessed from the JSF pages using the EL expressions e.g. #{bean1.operation1(paramX)} or #{bean1.attrib3}. Some of the Java classes in the business tier may not be EJBs and therefore not accessed by JSF pages, they are implemented to apply several functionalities required by EJBs. For instance, the similarity measurement between two texts that is used for the CBR in the decision recommendation process. The POJOs or the persistent classes are managed by the EJBs, these classes are composed only of attributes, getters and setters and one or more constructors. A persistent class contains an ID to allow easy identification of the objects in the EJBs or in the database.

Persistence Tier: This tier contains the DAOs to provide an abstract data retrieval instead of direct data access, the DAO is actually a Java class that works as an adapter between the data source and the J2EE applications and it can adapt to different storage schemes and the changes of the data source. The problem with accessing data directly is that the source of the data can change. Based on the Core J2EE Patterns (Deepak et al., 2001), the DAO completely hides the data source implementation details from its clients. Because the interface exposed by the DAO to clients does not change when the underlying data source implementation changes, this pattern allows the DAO to adapt to different storage schemes without affecting its clients or business components. Several operations can be implemented in the DAO depending from the requirements of the J2EE application. For example, in the DES or any other secured application, it was essential that the user passwords should be encrypted in the database, so the encryption and decryption functions should be present in the DAO for password storage/retrieval unlike other regular attributes like the username or e-mail address. The default operations that were automatically generated in the DAOs concern: Finding all instances, finding by properties, Saving, Updating and Deleting. Independent of the data source technology used in the source system, the components are able to exchange data correctly.

The “persistence.xml” is an essential file for the persistence, defining the persistence classes that are equivalent to the tables in the database and the connection parameters to the database (JDBC driver, username, password, database URL address). Should any of those parameters change, a simple configuration is required in this xml file and the entire application will run error-free.

- **The database machine**: The data tier or the EIS tier comprises data used by business logic. The enterprise information system tier handles EIS software and includes enterprise infrastructure systems, such as enterprise resource planning (ERP), mainframe transaction processing, database systems and other legacy information systems (Oracle, 2014). In this case, the source files are databases managed by a MySQL4 DBMS. The "persistence.xml" should be updated if any of those source files' connection parameters are changed. However, if the table is "altered", the changes will be applied on the equivalent persistent class (e.g. table name, column name, column type, column became unique, etc.).

A detailed implementation landscape is presented in annex 1.

6.2 Components Implementation

The next sections will contain a description of the Java packages used in the DES and all of their Java classes. For the Java classes description, a brief description containing the class name and role is presented. A detailed description containing the attributes and operations (attributes in the gray cells and operations in the green cells) as well as comments is available in annex 2, 3 and 4.

6.2.1 Managing Beans and Controller Package

In the DES implementation, this package contains the managed beans that are accessible by the JSF pages and are responsible for executing the main functionalities of the system (tuning parameters, decision-making, decision evaluation and decision recommendation). Figure 6.2 provides a list of classes in this package.

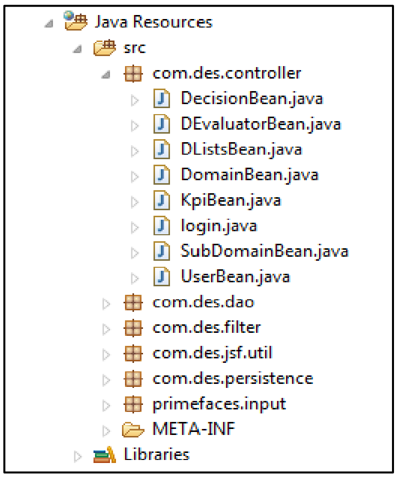

Figure 6.2 Controller package

In this section a high-level description of the classes (EJBs) contained in this package is offered. A detailed description of all attributes and operations for each class is available in the Annex 1: Managing Beans and Controller Package.

Managed bean DecisionBean

Class name: DesisionBean.java

Class role: This managed bean is a session-scoped bean that is responsible for the decision-making process in the DES, namely entering the context parameters, searching for recommended decisions, adapting a recommended decision to current context or creating a new decision.

Managed bean DEvaluatorBean

Class name: DEvaluatorBean.java

Class role: This managed bean is a view-scoped bean that is responsible for the automatic decision evaluation process in the DES as well as showing the previously-evaluated decisions and their impacts on KPIs, adding/viewing comments about decision.

Managed bean DListsBean

Class name: DListsBean.java

Class role: This managed bean is a session-scoped bean that is responsible for showing the lists of non-validated decisions so that the users with the right privileges can either validate or reject decisions. It also shows the decisions with the status "non-executed" so that users can change this status to "executed" if this is the case in reality.

Managed bean DomainBean

Class name: DomainBean.java

Class role: This managed bean is a session-scoped bean that is responsible for managing the domains in the DES (creation, editing, deleting).

Managed bean KpiBean

Class name: KpiBean.java

Class role: This managed bean is a session-scoped bean that is responsible for managing the KPIs in the DES (creation, editing, selecting, classification by sub-domains and sustainability natures).

Managed bean login

Class name: Login.java

Class role: This managed bean is a session-scoped bean that is responsible for logging in (after verifying username and password) and logging out. This bean will be used by all of the JSF pages to get the parameters of the logged-in user and to trigger the event of the logout.

Managed bean SubDomainBean

Class name: SubDomainBean.java

Class role: This managed bean is a session-scoped bean that is responsible for managing the sub-domains in the DES (creation, editing, deleting).

Managed bean UserBean

Class name: UserBean.java

Class role: This Managed bean is a session-scoped bean that is responsible for managing the users in the DES (Creation, Edition, and Deleting).

6.2.2 Filtering Package

This package only contains one Java class that is not a managed bean. However, it is essential for the DES or any secured, role-based software that has only limited access for each type of users. Figure 6.3 shows this class in its containing package.

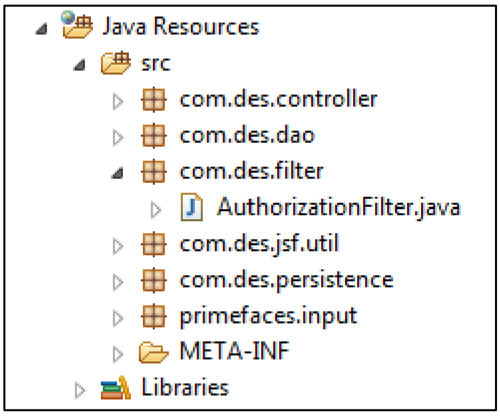

Figure 6.3 Filtering package

This class is initialized by the code below:

```
@WebFilter(filterName = "AuthFilter", urlPatterns = {
        "*.xhtml" })
```

This annotation `@WebFilter` is used to declare a servlet filter that will intercept the request from any web page and will act accordingly (sending responses) to the specified code in the method **doFilter()**.This annotation is processed by the container at deployment time, and the corresponding filter applied to the specified URL patterns (***".xhtml"*** here).

The method **doFilter()** blocks any request for unauthorized access. This unauthorized access in DES is detected if the logged-in user tries to access a web page that is not in the folder of the pages authorized for his role:

- **[priv = 1]** "Administrator".
- **[priv = 2]** "Decision-maker."
- **[priv = 3]** "Decision-validator."

For instance, if the user requests a page that is not in the open pages in the folder "/public/" but in the folder "/_Administrator/" and his role is not "Admin" **[priv != 1]**. The web filter redirects the navigation to a page informing that the access was denied.

Figure 6.4 shows the authorized web pages in their respective folders, the "**/_Decision_validator/**" folder is below the "**/_Decision_maker/**" folder because any decision-validator can access any page in this folder. However, decision-makers without validation privileges cannot access the decision validation pages and functionalities.

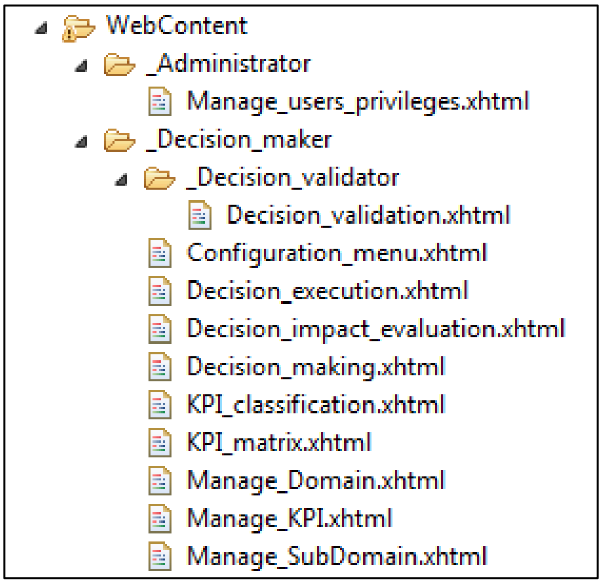

Figure 6.4 Authorized web pages for different user roles

6.2.3 Utilities Package

This package contains Java classes that are used by the EJBs. It contains also the "**SessionUtils.java**" class that is used by the authorization filter presented above. Figure 6.5 shows those non-EJB classes.

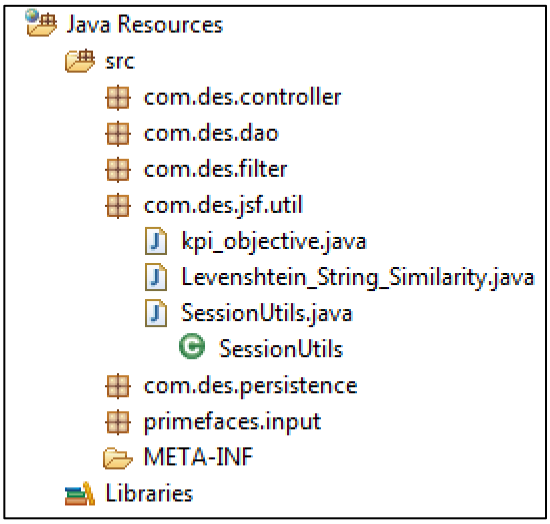

Figure 6.5 Utilities package

In this section a high-level description of the classes contained in this package and their bond to the EJBs and other requesting classes is presented. A detailed description of all attributes and operations for each class is available in the Annex 2: Utilities Package.

Class kpi_objective

Class name: Kpi_objective.java

Class role: This class is required by the Java Bean **"DecisionBean.java",** which is responsible for creating and adapting decisions.

It is a temporary class to represent the available KPI objectives for showing and selection in the UI. The KPIs are retrieved by the selected sub-domain and the decision objective will require the deadline and the planned value for each parameter. Therefore, a kpi_objective is defined by the KPI, planned_value, deadline date.

Class Levenshtein_String_Similarity

Class name: Levenshtein_String_Similarity.java

Class role: This class is used to calculate the similarity between two texts using the Levenshtein distance algorithm. Credits: (Gogolev, 2013).

Levenshtein Distance (LD) similarity is an algorithm developed by Vladimir Levenshtein in 1965. This algorithm measures of the similarity between two strings: e.g. the source string (S1) and the target string (S2). The LD is the number of deletions, insertions or substitutions required to transform S1 into S2 or vice versa. The similarity is case-insensitive (Gilleland, 2009; Sulzberger, 2017).

The DES will use this similarity calculation in the CBR mechanism for the decision recommendation in the bean "decisionBean.java". The algorithm will be used to calculate the similarity between the searched parameters in the context definition (objectives descriptions, problems descriptions).

As previously mentioned, the algorithm of Levenshtein distance (LD) – also called the "edit distance" – can calculate the similarity percentage between two strings.

Example: Given two strings: the source string (S1) and the target string (S2). Therefore, the higher the LD value, more different the strings S1 and S2 are. This value equals 0 if the two strings are similar.

Accordingly, after obtaining the value LD of two strings, we can reverse it to obtain how similar they are. The next formulas illustrate the calculation of the similarity percentage SP of two strings Str1 & Str2:

$$\mathrm{SP}(\mathrm{Str}_1, \mathrm{Str}_2) = \frac{\mathrm{Longest} \; - \; \mathrm{LD}(\mathrm{Str}_1, \mathrm{Str}_2)}{\mathrm{Longest}} * 100$$

where: The value "Longest" represents the length of longest string between the two strings $\mathrm{Str}_1, \mathrm{Str}_2$.

$$\mathrm{Longest} = \mathrm{MAX}(\mathrm{Str}_1.\mathrm{length}\,,\; \mathrm{Str}_2.\mathrm{length})$$

Attributes and operations	Comments
No attributes, only local attributes used within the static methods of this class.	
double **editDistance (<u>String</u> s1, <u>String</u> s2)**	A **static** method that calculates the Levenshtein distance or the edit distance between two strings **s1**and **s2**. The lesser the Levenshtein distance the more similar the two strings are. This algorithm have the same outcome of the function available with the library *common-lang3.jar*: **StringUtils -> getLevenshteinDistance(s1, s2)**.
double **similarity(<u>String</u> s1, <u>String</u> s2)**	A **static** method that calculates the similarity between two given strings **s1**and **s2**and returns a value between 0 and 1. '0' means that the two strings are not similar at all and '1' means that the two strings are identical (Case-insensitive).

Table 6.1 Levenshtein Method

Class SessionsUtils

Class name: <u>SessionsUtils.java</u>

Class role: This class is used to obtain parameters about the current HTTP session by the authorization filter to verify the “username” attributed to the session (on the authentication) and invalidate the current session by the bean “**login.java**” when disconnecting.

6.2.4 Data Access Objects Package

This package contains the DAOs used to retrieve data from the database and map them into the persistent Java classes. Each of those DAOs implements an interface containing all (or most) of its methods, whereby these methods are inherited from the Java interfaces. It would make more sense if a single Java interface was developed for all the existing DAOs, although the tool used to generate the DAO (MyEclipse Java Persistence API (JPA)) generated an interface for each DAO for no obvious reason. Figure 6.6 represents all the DAO classes used in the DES.

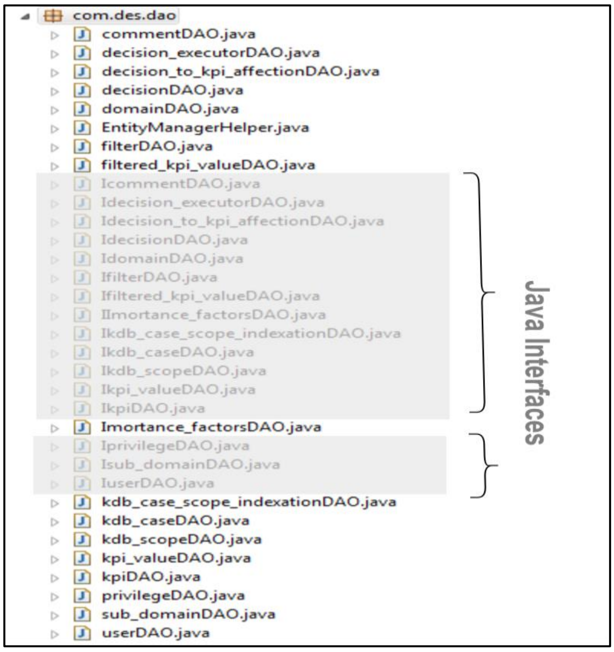

Figure 6.6 DAO Package

The default code in the DAOs has common objectives besides the personalized methods, they have no attributes but all of them have these methods:

- **Void Save(***[entity_class]***Entity)**: Perform an initial save of a previously unsaved entity.
- **Void Delete(***[entity_class]* **Entity)**:Delete a persistent entity.
- **Void Update(***[entity_class]***Entity)**:Persist a previously saved entity and return it or a copy of it to the sender.
- **Entity findById(***[type_ID]***id):** Finding the single entity with the given id.
- **List<Entity>findByProperty(String propertyName, Object value):** Finding all entities with a specific property value where the property is named “propertyName” and the “value” could be any type of object (String, Integer, Double, Date…)
- **List<Entity> findAll():** Finding all existing entities in the database.

Where *[entity_class]* could be any of the persistent classes (POJOs) like: "decision.java", "kpi.java", "sub_domain.java", "user.java"…

The classes are inside the package **com.des.persistence**, where the propertyName could be any of the attributes of the persistent classes.

Table 6.2 presents the additional methods that we developed inside the DAOs and the edited default methods to answer to the DES requirements.

Method header	Description
List<decision> **find_all_executed()**	Finding all executed decisions, that is, the execution date is not null. The default methods do not support finding null or not null values. (DecisionDAO.java).
List<decision> **find_all_evaluated()**	Finding all evaluated decisions, that is, the sustainability evaluation field is not null. (DecisionDAO.java).
List<decision> **find_all_evaluated(sub_domain sd)**	Finding all evaluated decisions for a specific sub-domain **sd**, namely the sustainability evaluation field is not null. (DecisionDAO.java).
kpi_value **find_last_before(Date dt, kpi k)**	Finding the last recorded value of the KPI **k** before or equal to the date **dt**. For example, if **dt** is the current date, this function returns the last recorded kpi_value of the KPI **k**. (Kpi_valueDAO.java).
List<kpi> **find_by_domain(domain d)**	Finding all the KPIs classified in the sub-domains that are under the domain **d**. A domain is not a property in the kpi persistent class therefore the DAO cannot directly find these instances with the method **FindByProperty**. (KpiDAO.java).
Boolean **validate(String userlog, String password)**	Returns true if the password string **password** matches the user login **userlog**. (UserDAO.java).
(Edited method) *Void* **FindByPassword(Object password)**	Since the password is encrypted in the database. The string comparison between the requested value **password** and the passwords stored in the database will be based on the encrypted strings of both values. If the encrypted string of the requested value **password** matches the password encrypted in the database means that their decrypted values are also the same. (UserDAO.java).
(Edited method) *void* **Save(Useru)**	The password entered from the UI is encrypted before saving the user **u** in the database. (UserDAO.java).

Table 6.2 Data Access Objects Methods

6.2.5 Persistence Package

This package contains the persistent classes (or Plain Old Java Objects (POJOs)) that are manageable by the DAOs and the EJBs. These classes represent the data stored in databases as Java objects. The persistence only contains private attributes (equivalent to the columns in the database), their public getters & setters and a number of public constructors. **MyEclipse Java Persistence API** (**JPA**) was used to generate those classes.

The first thing to note here is that the number of classes is slightly superior to the number of tables in the decision database, given that there are two types of classes:

- Classes representing tables: Each class of this category represents a table in the database like: "**comment.java**" representing table "des.ddb_comment", "**decision_executor.java**", "**decision_to_kpi_affection.java**". These classes are annotated and represented as follows:

```
@Entity@Table(name="ddb_comment" ,catalog="des")
public class comment {
                        //code
                  }
```

- Classes representing identificators: Each class of this category represents a composed primary key that occur in association classes for *Many to Many* relationships and any other tables that have multiple primary keys.
  E.g. The comment is an association between a user and a decision where the user can comment multiple decisions and the decision can have comments from multiple users. Therefore, the primary of the comment is composed of three elements keys:
  - Decision ID (Foreign key).
  - User ID (Foreign key).
  - Comment ID (to enable a user to comment a single decision more than once).
- To represent this with the DAO in Java application, we have the classes "*[entity]***Id.java**" like "**commentId.java**", this class have only three attributes that represent the composed primary key of the class "**comment.java**". It will be annotated and represented as follows:

```
@Embeddable
public class commentId  {
private Integer idComment;
private String idUser;
private Integer idDecision;
//Rest of the code
}
```

Figure 6.7 represents all the persistent classes used in the DES.

All of these classes must be listed in the persistence file "**persistence.xml**"; otherwise, they will not be viewed as persistent and the DAOs cannot manage them with the database.

The following tables represent the persistent classes (POJOs), their attributes and methods for each class. Their bound in the database (Table name, column name) is also presented here.

Figure 6.7 Persistence Package

In this section a high-level description of the persistent classes (POJOs), contained in this package is offered. A detailed description of all attributes, methods and their dound in the database (table name, column name) for each class is available in the Annex 3: Persistence Package.

Persistent class comment

Class name: comment.java

Class role: This class represents the table "ddb_comment" in the decision database.

Persistent class commentId

Class name: commentId.java

Class role: This class represents the composed ID of the class "**comment.java**".

Persistent class decision_executor

Class name: Decision_executor.java

Class role: This class represents the table "ddb_decision_executor" in the decision database.

Persistent class Decision_to_kpi_affection

Class name: Decision_to_kpi_affection.java

Class role: This class represents the affection or the impact of the decisions on KPIs. The table bound to this class in the decision database is "ddb_decision_to_kpi_affection".

Persistent class Decision_to_kpi_affectionId

Class name: Decision_to_kpi_affectionId.java

Class role: This class represents the composed ID of the class "**decision_to_kpi_affection.java**".

Persistent class Decision

Class name: Decision.java

Class role: This class represents the table "ddb_decision" in the decision database.

Persistent class domain

Class name: domain.java

Class role: This class represents the table "ddb_domain" in the decision database.

Persistent class Importance_factors

Class name: Importance_fators.java

Class role: This class represents the table "ddb_Importance_factors" in the decision database.

This is a small table with only one row that contains the importance factors of direct and indirect impact on KPIs in the decision evaluation process.

Persistent class Importance_factorsId

Class name: Importance_fatorsId.java

Class role: This class represents the composed ID of the class "**Importance_factors.java**". The JPA tool generates this type of class also to identify the referred class if the referred class does not have any primary keys like this one (**Importance_factors.java**).

Persistent class Kdb_scope_indexation

Class name: Kdb_case_scope_indexation.java

Class role: This class represents the table "Kdb_case_scope_indexation" in the knowledge database.

It is a class where it specifies the indexation of cases by scopes in the database. This is an independent class since the relationship between cases and scopes is "Many to Many".

Persistent class Kdb_scope_indexationId

Class name: Kdb_case_scope_indexationId.java

Class role: This class represents the composed ID of the class "**Kdb_case_scope_indexation.java**".

Persistent class kdb_case

Class name: kdb_case.java

Class role: This class represents the table "kdb_case" in the knowledge database. It is the case (or context) of a decision. The decision recommendation process in the DES uses CBR to recommend decision so he uses this class as a model for previous cases retrieved from the knowledge database.

Persistent class kdb_scope

Class name: kdb_scope.java

Class role: This class represents the table "kdb_scope" in the knowledge database. It is the case (or context) of a decision. The scopes are used to index the cases in the knowledge database for better and more accurate retrieval instead of relying only on textual similarity between problem and objective descriptions.

Persistent class kpi_value

Class name: Kpi_value.java

Class role: This class represents the table "ddb_Kpi_value" in the decision database.

Persistent class kpi_valueId

Class name: Kpi_valueId.java

Class role: This class represents the composed ID of the class "**kpi_value.Java**".

Persistent class kpi

Class name: Kpi.java

Class role: This class represents the table "ddb_kpi" in the decision database.

Persistent class Privilege

Class name: privilege.java

Class role: This class represents the table "ddb_privilege" in the decision database.

Persistent class sub_domain

Class name: Sub_domain.java

Class role: This class represents the table "ddb_sub_domain" in the decision database.

Persistent class user

Class name: user.java

Class role: This class represents the table "ddb_user" in the decision database.

6.3 System Configuration and Settings

This section is reserved to describe the decision evaluation system configuration and settings. The DES is developed as a web application using the widely-known Java Enterprise Edition specifications. Eclipse IDE (Kepler) was used to develop the applications' Java classes and view pages (in xhtml form) and the tool MySQL workbench 6.3 was relied on to design, develop and configure the database stored in the DBMS MySQL (version 4). With conformance to the technology used to construct the DES, a potential user will be required to have several tools in the hosting machine and to follow a set of simple steps.

6.3.1 Prerequisites

In order to install and configure the DES the following pre-requisites are necessary:

- Web navigator (e.g. Google Chrome, IE, Firefox...), preferably a recent version.
- Apache Tomcat 7.0 web server.
- Application folder containing its architecture (As in the workspace of the IDE) or the ".war" file of the project that can be generated with the IDE.
- DBMS with the "decision database" designed with the same architecture presented in chapter 6.

N.B. The DBMS should not be necessarily MySQL like the one used in the development phase, any DBMS can be used to store and manage the necessary DB provided that the configuration file "Persistance.xml" is updated and the correct JDBC driver is within the project folder. This is one of the advantages of the DAO used in the DES.

6.3.2 Installation Steps

With all the perquisites folders, databases and tools are installed in the user's machine, the user needs to follow these three simple steps to run the DES:

Step 1

Deploy the application folder (or the .war file) in the tomcat web server by placing it in the "webapps" folder typically found in tomcat installation path like:
" C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps"

Step 2

Start the tomcat service using the application "Tomcat7w.exe" located in the installation path typically like: *"C:\Program Files\Apache Software Foundation\Tomcat 7.0\bin\Tomcat7w.exe"*

Step 3

Now the Apache server is started and the DES web application, if properly deployed, should be accessed with any web navigator using this address by default: " *http://localhost:8080/Decision_Evaluation_System/*" with **8080** is the default assigned port for apache tomcat that can be configured later and **/Decision_Evaluation_System/** is the default context path for the DES.

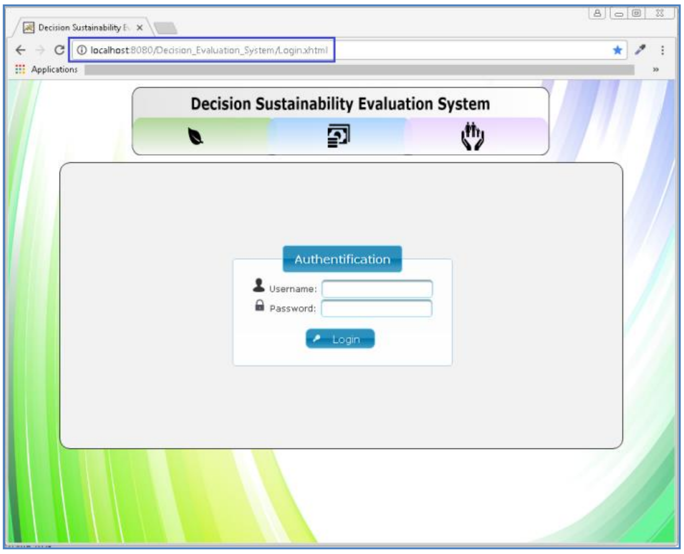

Figure 6.8 DES start page

If the web application is unavailable and the tomcat message "The requested resource is not available" appears, make sure that you entered the correct context path of the DES. If the problem persists, the **Tomcat manager application** should be accessed to deploy (re-deploy) the .war file of the project.

Step 4

Make sure that you created a user with the role for accessing the GUI of the manager application. This can be done by checking the **tomcat-users.xml** file typically found in: *"C:\Program Files (x86)\Apache Software Foundation\Tomcat 7.0\conf\tomcat-users.xml"* If the file doesn't contain any user with this role, you may create new.

Step 5

Access the manager application with any web navigator using this address by default: *" http://localhost:8080/manager"* with login and password just like in the configuration file above.

In this application, you should see the context "/Decision_Evaluation_System" in the contexts lists. If not, you can deploy it in the same interface below by uploading the .war file of the DES called „Decision_Evaluation_System.war" like in the screenshot below.

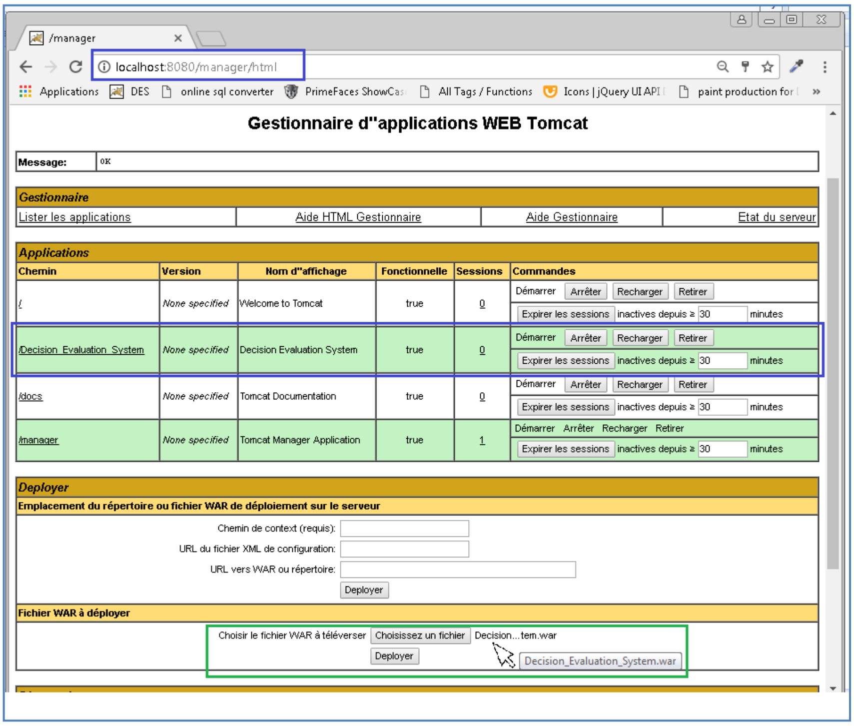

Figure 6.9 Check DES project deployment

After those steps, the DES should run without any problems. You can access the welcome page ("Login.xhtml"). The rest of the configuration (Managing users, domains and KPIs...) is within the DES application.

6.3.3 Configuration steps

To manage the access to the DES and the authorizations of different users, at least one user with the role "Administrator" must be stored in the decision database (added directly with the DBMS not from the DES).

This administrator can login to the DES using "admin" for Login ID and password and use the available interface to see the list of existing users. An administrator can select a user to edit its attributes and privilege (Administrator, decision-maker only, decision-maker and validator) or to delete it:

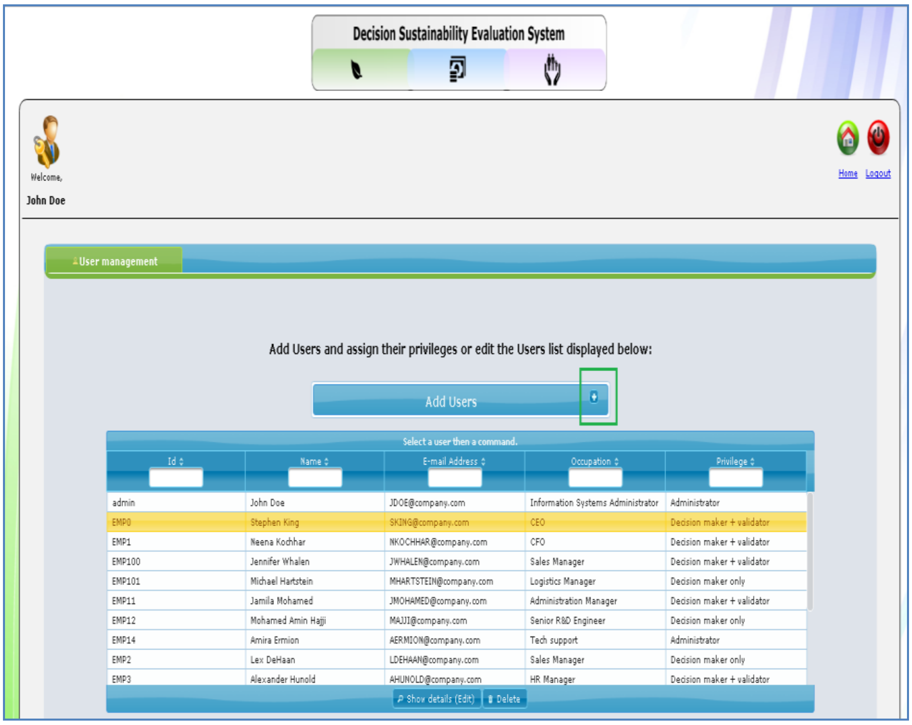

Figure 6.10 Manage users

To add a user, an administrator expands the panel "Add Users" in the interface using the button (framed in green) and enters the different attributes in addition to selecting the privilege:

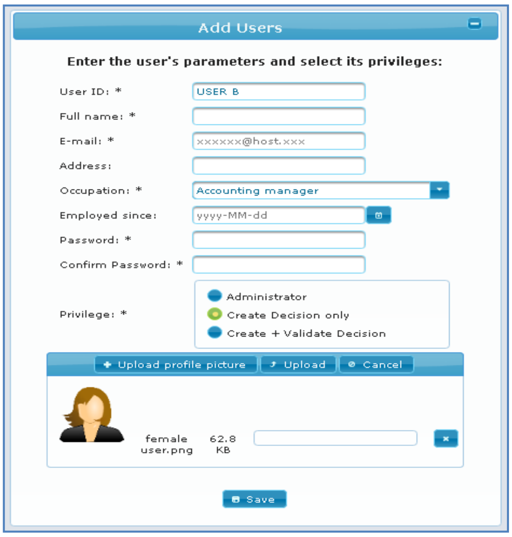

Figure 6.11 Add users

To edit a user, select a user from the list and press the button "Show details (edit)". To delete a user, select a user from the list and press the button "Delete". A confirmation dialog appears before the non-reversible suppression of the user.

The configuration of the KPIs, their domains (HR, Logistics...) and sub-domains (Healthcare, Transport ...) are assigned to decision-makers (business users) in the DES rather than administrators (technical users).

In all the interfaces for decision-makers, they can find the configuration icon  that permits the access to the configuration menu presented below, the user chooses the type of parameters to configure:

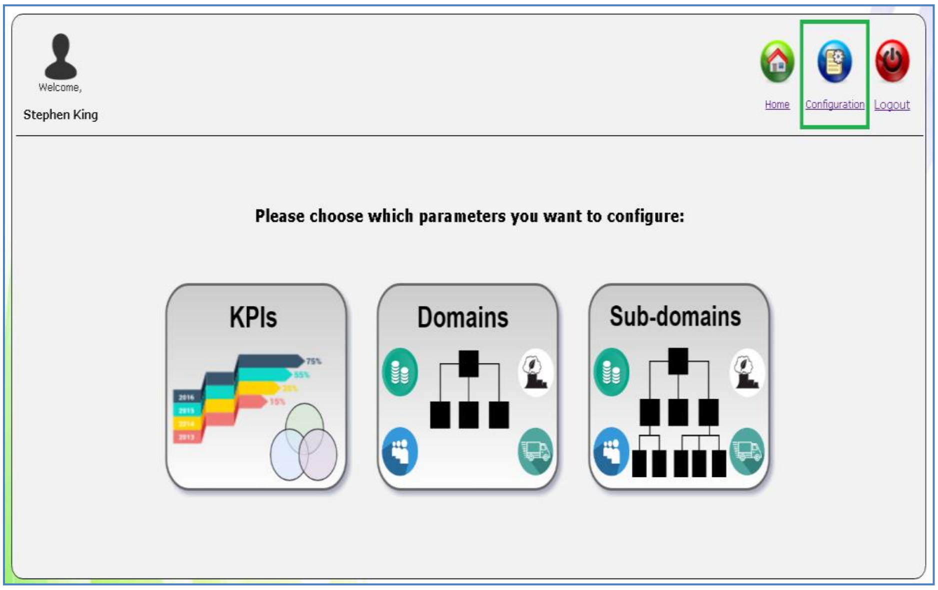

Figure 6.12 DES configuration interface

KPIs configuration

The configuration of KPIs consist of Adding KPIs by entering their attributes, Editing, deleting KPIs and classifying KPIs by their domains/ sub-domains and sustainability natures.

In the figure below, the user selects a KPI from the list to show further details and to edit attributes (all KPI attributes except the ID) by selecting a KPI and clicking "Show details (edit)". He can also select a KPI to delete by clicking "Delete".

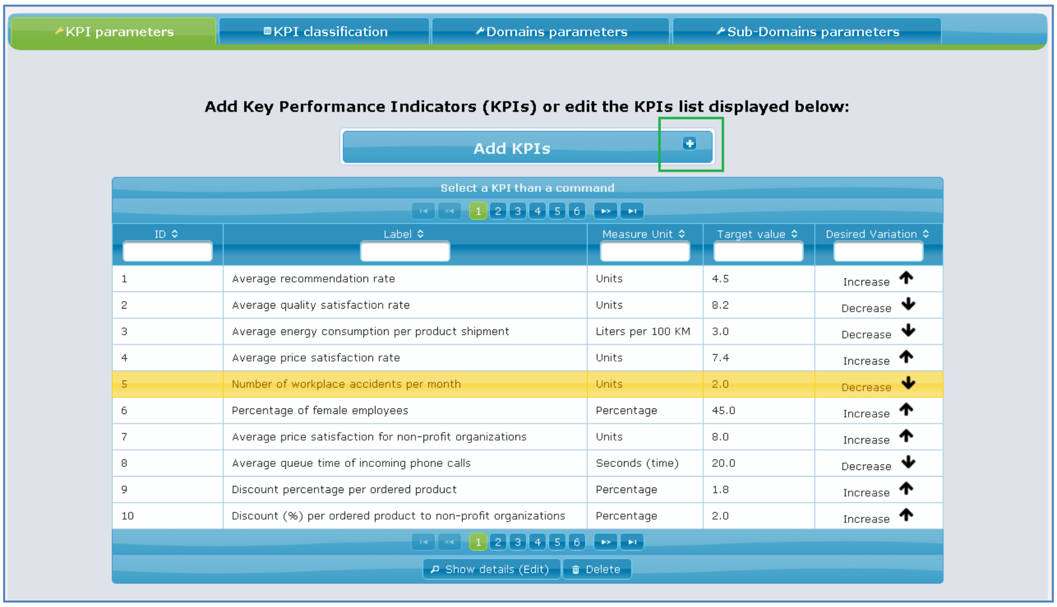

Figure 6.13 DES configuration interface - KPI

To add a KPI, a user expands the panel "Add KPIs" in the interface above using the button (framed in Green) and enters the KPI parameters.

A further configuration is required before using KPIs as decisions' objectives which is the KPI classification, it can be done easily with the interface below.

A user can see the list KPIs and selects its classification with a dropdown menu. The classification will be saved instantly without a submit button.

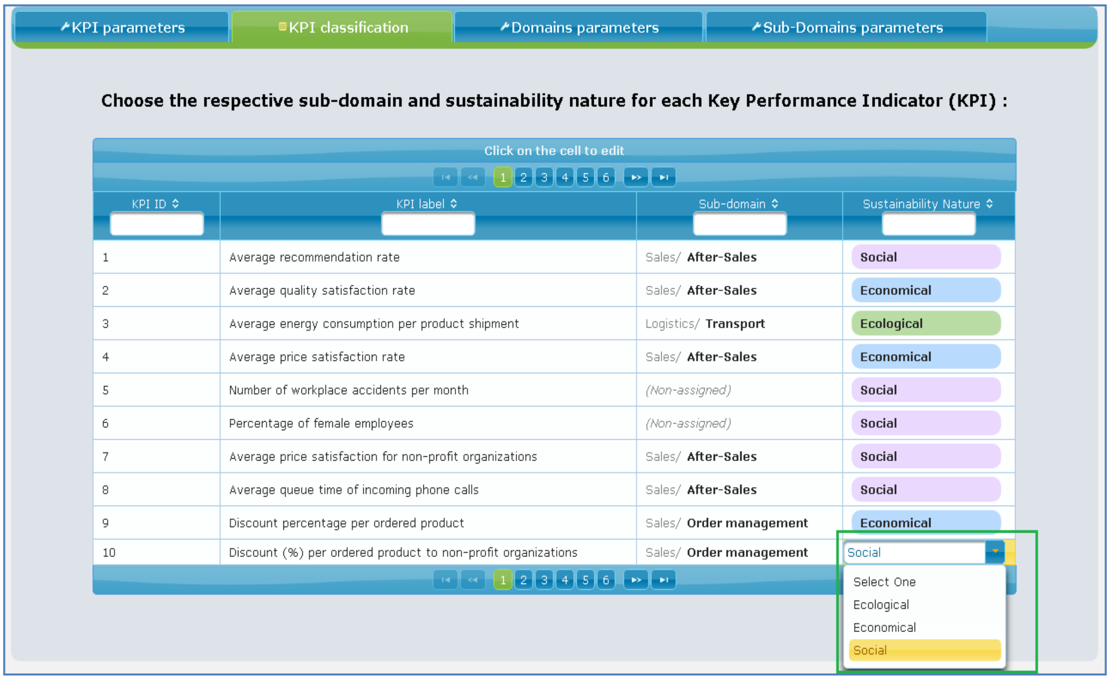

Figure 6.14 DES configuration interface – KPI classification

Domains configuration

The configuration of domains consist of Adding them by entering their attributes, Editing, deleting them.

In the figure below, the user selects a domain from the list to show further details and to edit attributes (all domain attributes except the ID) by selecting a domain and clicking "Show details (edit)". He can also select one to delete by clicking "Delete".

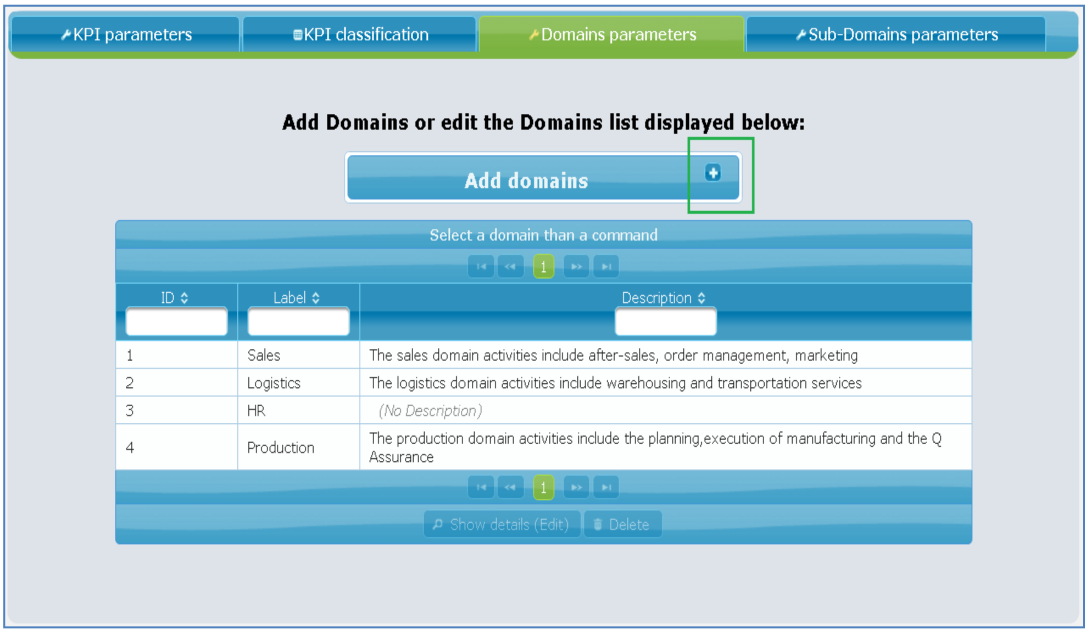

Figure 6.15 DES configuration interface – domains

To add a domain, a user expands the panel "Add domains" in the interface above using the add button and enters the domain parameters.

Sub-domains configuration

The configuration of sub-domains consist of Adding them by entering their attributes, Editing, deleting them.

In the figure below, the user selects a sub-domain from the list to show further details and to edit attributes (all sub-domain attributes except the ID) by selecting a sub-domain and clicking "Show details (edit)". He can also select one to delete by clicking "Delete". To add a sub-domain, a user expands the panel "Add sub-domains" in the interface above using the add-button (framed in Green) and enters the sub-domain parameters.

With the DES installed properly and the system configuration completed, users can start following the decision-making process by planning, choosing, making, validating decisions and tracking their sustainability evaluations based on their impact on KPIs. Users may also comment and evaluate decisions themselves. After explaining the integration aspects in the next section an industrial business case using the DES will be presented.

6.4 Integration Aspects

As presented in the previous chapters, evaluating decisions in the DES is built around how it has affected different types of KPIs. The KPIs that can be affected by a decision are all classified within its same sub-domain but may have different sustainability natures (ecological, economical or social). By analogy, KPIs from different sub-domains are affected by multiple decisions from different sub-domains. The similar logic applies for KPIs from different domains (see figure 6.16).

With so many KPIs being progressively impacted in positive and negative directions through actions taken by stakeholders, an observer may wonder what is about potential association or links between those variables: e.g. if KPI "A" and "B" are enhanced in the same period, does that mean they have a link? If KPI "C" diminishes with the enhancement of KPI "A", do those KPIs also have a relationship?

Studying and determining such relationships or links by analysing a data set of two variables in pairs (bivariate data) is referred to as "correlation" in statistics. A correlation between two variables is a normalized measurement of how the two variables are linearly related. There are many statistical and techniques to calculate different types of correlations (e.g. Bravais-Pearson correlation, Spearman correlation…). In decision-making and performance measurement context, a multitude of studies have been conducted for measuring KPI correlations like the ones listed in the work of (Rodríguez et al., 2010).

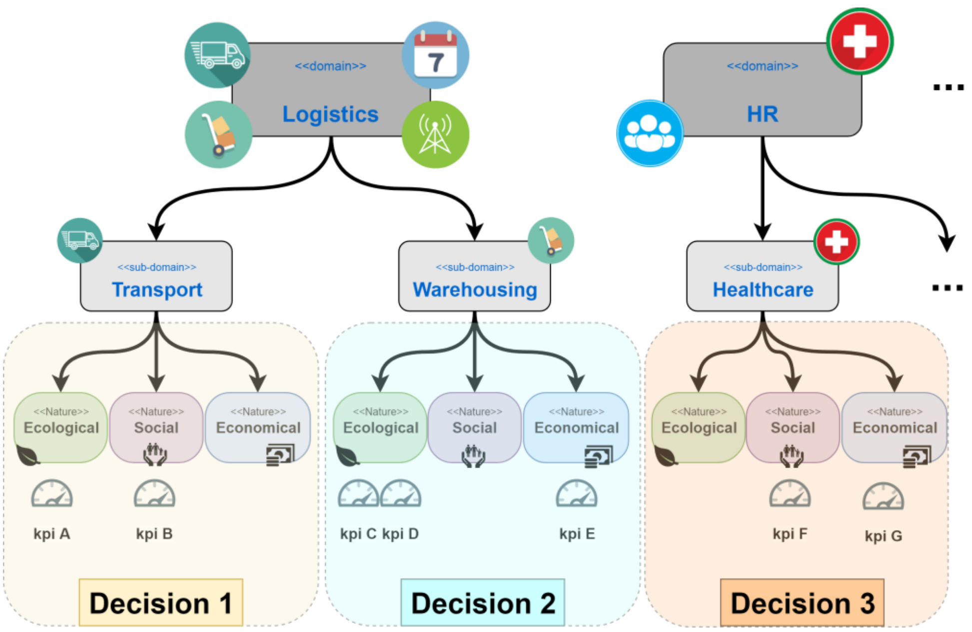

Figure 6.16 Example of KPI classification and related decisions

Calculating KPI correlation and determining their relationships can be valuable for decision-making to provide more observation and better intuition when making a decision that will affect KPIs. Knowing that two or more KPIs are closely related, a decision-maker could foresee the range of impact of his decision that has one of those KPIs as an objective. Additionally, the identification of KPI relationships would allow the reduction of the number of KPIs to be monitored, For instance, if it is found that two KPIs maintain a close relationship, as the variation of one provokes the variation of another, it then could be possible to concentrate efforts on controlling and monitoring only one of them (Rodríguez et al., 2010).

In the DES, since KPIs are classified based on their sustainability nature, it may be easy and tempting to assume that:

- Case (1): Two KPIs in the same class, ecological for instance, have a close, positively-linear relationship because they are affected by a number of ecologically-sustainable decisions.
- Case (2): Two KPIs from ecological and economical classes have negative relationships (when one KPI improves, other one diminishes…) because a number of affecting decisions do not yet have the sustainability balance between economy improvement and environmental preservation.
- Case (3): Two KPIs from the same or different sustainability classes are closely related because they are almost always used as objectives together by decision-makers.

Those shallow assumptions are inaccurate because:

- To Case (1): Two KPIs may be from the same sustainability nature but are differently impacted by the same decision (e.g. The KPI impact of the decision in this business case. See economical KPIs in figure 10) or different decisions. Judging a decision to be economically or ecologically sustainable is about having a good average impact on economic or ecologic KPIs, respectively, and not necessarily enhancing all of them.
- To Case (2): Two KPIs may be from different classes but they do not have any negative or positive relationship. Like in case (1), judging a decision to be sustainably balanced or not is also about calculating the average impacts on KPIs of different classes of sustainability and not necessarily improve all KPIs from one class while deterioring KPIs from another.
- To Case (3): Two KPIs may be often used as objectives together for decisions but may not be closely related, because choosing objectives is not automatic or calculated based on correlation, it is completely manual and based on the decision-maker's requirements, vision and experience.

There could be many other cases when users of the DES could observe a potential relationships between KPIs but not all of them are accurate. The relationships between the KPIs in the DES are determined by organized, studied and long-term observations of the evaluated decisions and their impacts on KPIs.

A more accurate and effective way to determine relationships between KPIs is the calculations of their values' correlations using (1) statistical techniques and (2) multi-criteria decision aid methods (MCDA) (Rodriguez et al., 2009; Rodríguez et al., 2010).

In order to achieve the full potential of the DES there are some tips and advices (good practices) that need to be considered by the client organization. Those advices, if followed, will each contribute to expected behaviour of specific DES features like accurate evaluation of decisions' sustainability and relevant decision recommendation…

One of the requirements is the good quality of data integration from the various sources to the decision database, specifically, alimenting KPI values. The process of data integration (i.e. ETL) need to be consistently scheduled and thoroughly studied between its developers and business experts to produce precise KPI values. This will help the DES to calculate decision impact accurately generate reliable decision evaluation.

Also for the feature of decisions' impact and evaluation, it requires a certain level of cooperation and coordination need to be reached by the organization's stakeholders. The experts and analysts from the different sectors of the enterprise activities are needed to establish an efficient, realistic classification of the activities to domains and sub-domains. The same qualities are needed for the categorization of KPIs by those domains and sub-domains. This is to avoid most of unwanted (invisible) impact of decisions to other falsely-classified KPIs assigned to other sub-domains. From the decision-makers' point of view, even if KPIs are flawlessly categorized to sub-domains, the decision itself need to be well-studied and designed to target only KPIs in its sub-domain to also avoid unwanted impacts and potential unbalance of the organization's departments. This requires a good comprehension of the practices of decision-making (strategic, tactical and operational decisions) and the collaborative efforts of decision committees and decision-validators.

As for good practices to obtain relevant and helpful decision recommendation, it is encouraged for decision-makers to use common sustainability vocabulary and terms (i.e. Sustainability glossary) for describing problems (occasions) and objectives in the decision recommendation interface. This applies when searching for recommended decisions or labelling their entered decisions. This will significantly enhance the DES ability to calculate the similarity of recommended decisions to current contexts of decision-makers. Such glossary of sustainability can be established locally from sustainability specialists within the organization or inspired and sourced from international sustainability organizations like the US Environmental Protection Agency (EPA)[12.] It should be also shared and well-understood by decision-makers.

6.5 Evaluation through a Business Case

In the context of testing the developed DES, the industrial partner, the company Intercolor provided an example of a decision that has been executed and have positive and negative effect on different economical, ecological and social KPIs.

This case is related to the production domain and specifically to the production planning and strategies sub-domain. The table 6.3 presents the decision's parameters in addition to its brief descriptions of objectives and occasions and the scopes that are necessary for later decision retrieval via the CbR approach.

The DES should be able to:

- Enable a secured authentication for the decision-maker and the decision-validator (These two roles could be assigned to one single user).
- Enable the decision-maker to see the list of available KPIs and their description within the decision's domain/ sub-domain (Production/ Planning) via the KPI matrix interface. This step helps the decision-maker to have a global view of the KPIs and, by extension, set the decision's objectives with more precision. If necessary, a decision-maker can add a set of KPIs that are missing.
- Recommend decisions based on the scopes and brief objectives and occasions descriptions (context parameters) entered by the decision-maker.
- Enable the creation of a new decision and set its KPI objectives, planned values and KPIs.
- Evaluate the sustainability of a decision based on its direct and indirect impact on KPIs. The evaluation of a decision can occur once all of its deadlines were reached.
- Enable decision-makers to view the sustainability evaluation of the decision and its detailed impact on different categories of KPIs (ecological, economic, social).
- Enable decision-makers to share knowledge and experience about the evaluated decisions via adding comments and files (documents, images, videos…) and give their own personal evaluation of the decision [1..10]. The average user rating of the decisions is calculated and shown in their respective comments section.

[12] https://www.epa.gov/

Decision	New process for paint containers' cleaning (More paint remover quantity + new distiller).	
Domain / Subdomain	Production/ Planning.	
Detailed problem description	The current container cleaning process and the used distiller itself is causing high Volatile organic compounds (VOCs) emissions and it is very time consuming. Besides the bad environmental impact, this causes inconvenience and complaints for the employees on working conditions (strong odor, dizziness, sickness…).	
Solution description	It is recommended to increase the paint remover quantity to accelerate the process of paint container cleaning and using a more performing distiller.	
Brief Occasion(s) description(s)	Too much VOC emission.	
	Low employee satisfaction rate.	
	High cost of cleaning operations.	
Brief Objective(s) description(s)	Decrease VOC emission.	
	Decrease cleaning costs.	
	Lower time of cleaning.	
	Enhance employee satisfaction rate.	
	Enhance working conditions.	
Scopes	**Ecological**	Environmental Protection and habitat restoration.
		Pollution Prevention (air, land, water, waste).
	Economical	Cost savings.
		Long range planning.
	Social	Law and Ethics.
		Quality of life.

Table 6.3 Decision parameters

The following paragraphs define the steps of the decision-making and tracking in the DES and contain both description and screenshots.

6.5.1 Login into the DES

To start using the DES, a user should login to the web-based application using his user name and password. After a successful authentication, the system will identify the user and its privilege with the system (1- Administrator, 2- Decision-maker only, 3- Decision-maker + validator). Only authorized interfaces and functionalities will be available for each type of users.

For this business case, we are only interested in the second and third roles of users (decision-makers, decision-validators). Administrators can only manage the access and authorizations of users and don’t take part in the decision-making process. The following screenshots are taken from the actual developed DES but the displayed personal data within are virtual and does not represent the real production data of the company Intercolor.

Figure 6.17 shows the login interface where a user with login “**EMP12**” is attempting an authentication. This user is named *"Mondher Hajji"* and his role in the DES is "Decision-maker" so he will not be able to validate his own or any other decisions.

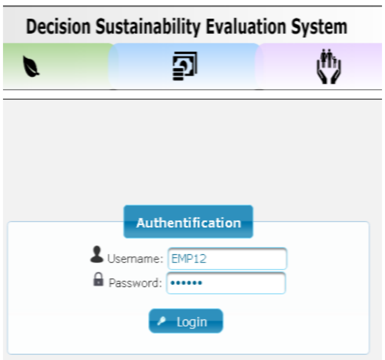

Figure 6.17 Login interface

6.5.2 KPI matrix dashboard

As mentioned earlier, the decision-maker checks the KPI matrix interface to display all classified KPIs by their domains/ sub-domains and their sustainability natures (ecological, economic, and social). He can also display the label, description, target value, measure unit and target value for each KPI after clicking the scope i  .

Should any KPI be missing from the list and is necessary to be set as an objective for the decision, the decision-maker can add this KPI from the configuration menu.

Figure 6.18 shows one of the dashboards in the KPI matrix interface which displays the KPIs for the production domain and its two sub-domains:

- Quality Assurance (collapsed in the screenshot).
- Planning (Expanded in the screenshot).

The KPIs that can be related to the decision of optimizing the paint container cleaning are shown in the dashboard (framed in Blue) with other unrelated KPIs from the same sub-domain. For example:

- The Volatile Organic Compound (VOC) emission measured for each container cleaning operation (KPI 29).
- The average cost and time of the container cleaning operation (KPI 42 and 43).
- The monthly numbers of complaints and sick leaves for the cleaning employees (KPI 53 and 54).

Production				
	ID KPI	Name	Measure unit	Sustainability nature
		Quality Assurance		
		Planning		
	29	VOC Emission per cleaning operation	Grams	**Ecological**
	35	Percentage of RM suppliers from internal regions	Percentage	**Social**
	39	Average energy consumption per manufacturing operation	Kilowatt	**Ecological**
	40	Usage of renewable energy in manufacturing	Percentage	**Ecological**
	41	Percentage of products manufactured using renewable energy	Percentage	**Ecological**
	42	Average cost per cleaning operation	TND	**Economical**
	43	Average time taken per cleaning operation	Minutes	**Economical**
	45	Cost of paint remover per cleaning operation	TND	**Economical**
	46	Recyclable paint generation amount per cleaning operation	Kilogram	**Ecological**
	47	Recyclable paint generation amount per container cleaning	Kilogram	**Ecological**
	48	Average hazard rate of wastes generated	Units [1..5]	**Ecological**
	49	Percentage of hazardous wastes generation in manufacturing	Percentage	**Ecological**
	50	Average diff. between RM market price and actual price	TND	**Economical**
	51	Average transportation fees paid per RM shipment	TND	**Economical**
	52	Average discount percentage obtained from RM suppliers	TND	**Economical**
	53	Monthly number of complaints for cleaning operators	Units	**Social**
	54	Monthly Number of sick leave days in the production dept.	Units	**Social**
	55	Satisfaction rate of cleaning operators	Units [0..10]	**Social**
	56	Monhtly maintenance cost of distillers	TND	**Economical**
	60	Non-reusable waste generation per cleaning operation	Kilogram	**Ecological**

Figure 6.18 KPI matrix for the production domain

6.5.3 Search decisions

Whether the decision-maker wants to create a new decision or choose a system-recommended decision, he should enter the context parameters of the decision (domain/sub-domain, description of problems and objectives, scopes).

After clicking the button "Validate Context", an Info message appears: "*The context is now validated, please proceed to choose a recommended decision or create a new decision*".

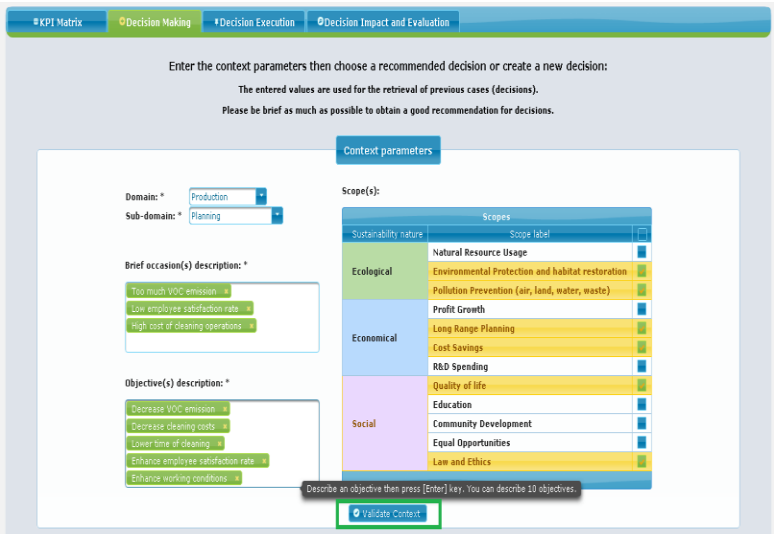

Figure 6.19 Enter context parameters

6.5.4 Get recommended decision

Only after validating the context, the decision-maker can see the recommended decisions shown with their sustainability evaluations and their similarity percentage with the current context.

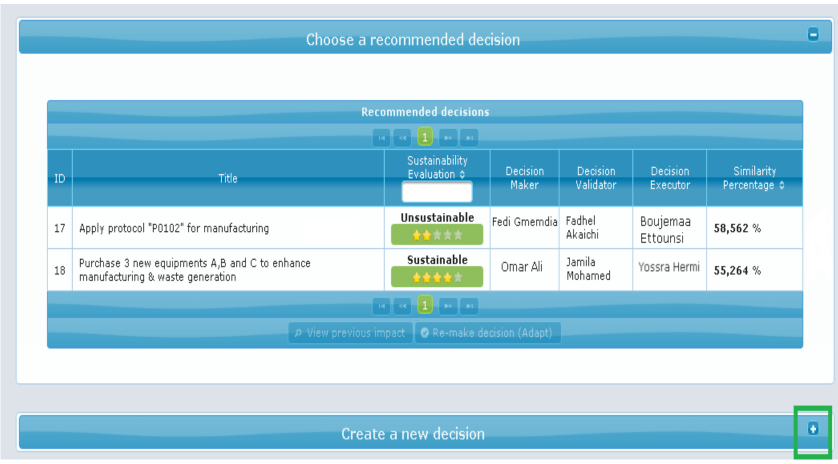

Figure 6.20 Display recommended decisions

The two recommended decisions displayed in figure 6.20 are necessarily related to the same domain and sub-domain selected in the context definition. Their respective similarity percentages are about 55-58%, this is due to the fact that these decisions:

- Are in the same domain/sub-domain.
- Share some of the scopes like: Pollution prevention, Cost savings, Environmental Protection and habitat restoration, Quality of life, Long range planning...etc.
- The problem of "increased VOC emission" and the objective of "decrease VOC emission" were treated by both of these two decisions.

However, both of these decisions are related to the manufacturing operation and not the cleaning operation so the decision-maker will have to "create a new decision".

6.5.5 Creating a new decision and setting objectives

The decision-maker proceeds to creating a new decision in the context that he entered by expanding the decision creation box in the interface. The context parameters are displayed for further re-checking and the decision form is shown figures 6.21 and 6.22. This form contains:

- Text areas to enter the decision parameters (Title, occasion description, solution description, decision-executor full name, e-mail address of the decision-executor).
- A data table containing the KPIs in the selected sub-domain showing the KPI parameters (label, target value, measure unit). The decision-maker may select the KPIs by checking them (left column of the table) and enter the planned value and the deadline of reaching the objective (right two columns of the table, respectively).

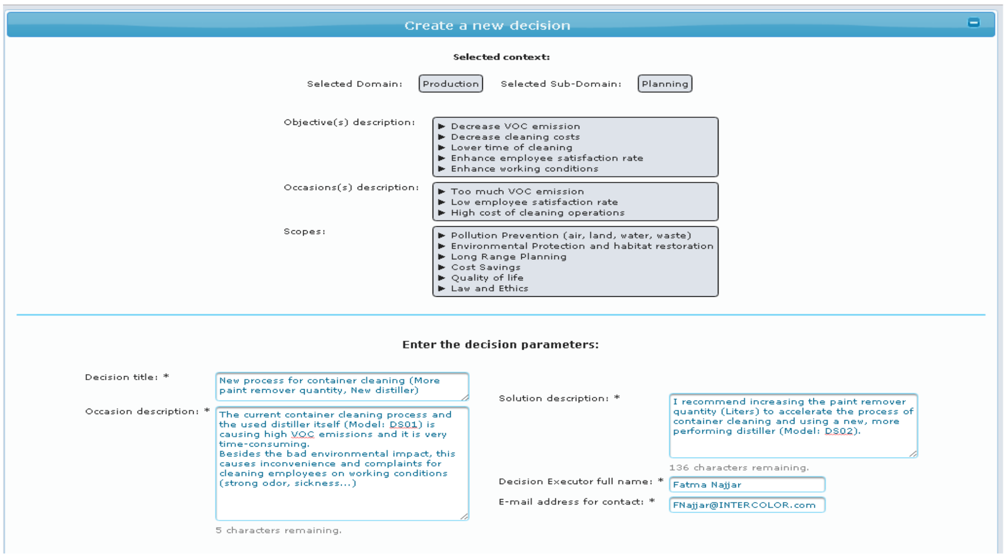

Figure 6.21 Enter decision parameters

Double-click on the cell (Planned Value, Deadline) to Edit

KPI label	Target value	Measure unit	Desired variation	Planned value	Deadline
VOC Emission per cleaning operation	4.8	Grams	Decrease ↓	5.35	2017-07-20
Percentage of RM suppliers from internal regions	25.0	Percentage	Increase ↑	(Non-assigned)	(Non-assigned)
Average energy consumption per manufacturing operation	660.0	Kilowatt	Decrease ↓	(Non-assigned)	(Non-assigned)
Usage of renewable energy in manufacturing	65.0	Percentage	Increase ↑	(Non-assigned)	(Non-assigned)
Percentage of products manufactured using renewable energy	60.0	Percentage	Increase ↑	(Non-assigned)	(Non-assigned)
Average cost per cleaning operation	400.0	TND	Decrease ↓	470.0	2017-07-20
Average time taken per cleaning operation	180.0	Minutes	Decrease ↓	175.0	2017-07-20
Cost of paint remover per cleaning operation	350.0	TND	Decrease ↓	(Non-assigned)	(Non-assigned)
Recyclable paint generation amount per cleaning operation	7.2	Kilogram	Increase ↑	6.7	2017-07-20
Recyclable paint generation amount per container cleaning	1.0	Kilogram	Increase ↑	0.75	2017-07-20
Average hazard rate of wastes generated	1.6	Units [1..5]	Decrease ↓	(Non-assigned)	(Non-assigned)
Percentage of hazardous wastes generation in manufacturing	15.0	Percentage	Decrease ↓	(Non-assigned)	(Non-assigned)
Average diff. between RM market price and actual price	-2.4	TND	Decrease ↓	(Non-assigned)	(Non-assigned)
Average transportation fees paid per RM shipment	5.4	TND	Decrease ↓	(Non-assigned)	(Non-assigned)
Average discount percentage obtained from RM suppliers	13.0	TND	Increase ↑	(Non-assigned)	(Non-assigned)
Monthly number of complaints for cleaning operators	0.0	Units	Decrease ↓	2.0	2017-07-20
Monthly Number of sick leave days in the production dept.	0.5	Units	Decrease ↓	0.6	2017-07-20
Satisfaction rate of cleaning operators	7.5	Units [0..10]	Increase ↑	7.5	2017-07-20
Monhtly maintenance cost of distillers	30.0	TND	Decrease ↓	35.5	2017-07-20
Non-reusable waste generation per cleaning operation	4.0	Kilogram	Decrease ↓	4.0	2017-07-20

View selected

Figure 6.22 Set decision objectives

For this case, the decision-maker **"EMP12"** entered the decision parameters of the container cleaning decision *"New process for paint containers' cleaning (More paint remover quantity + new distiller)"* and its detailed occasion description and solution description. He also set the name and the e-mail of the employee who will be in charge of executing and controlling this decision "Fatma Najjar" which is referred to as the decision-executor. In the same interface just below text boxes, the decision-maker selected the KPIs that should be set as objectives of this decision and assigned the planned values and the deadline dates. Once the maximum deadline date is reached, the decision can be evaluated based on the impact on those selected KPIs (direct impact) and on other KPIs in the same sub-domain (indirect impact).

6.5.6 Validating the decision

Since the decision-maker (**"EMP12"**, "Mondher Hajji") does not have the privileges of validating the decision, another user will validate or reject this decision which is referred to as "Decision-validator". Figure 6.23 shows the interface of the decision validation and the logged decision-validator (framed in blue), "EMP11 - Jamila Mohamed". In this interface, the decision-validator can see all the validated, rejected and pending decisions. The decision can be selected to its details and planned objectives, to validate it and to reject it (With optionally entering the rejection reason).

In this case, the decision-validator found the container cleaning decision as "Pending" and clicks the button labelled "Validate decision" (framed in green) and this will result changing its status to "Validated" and it can be viewed in the decision execution interface.

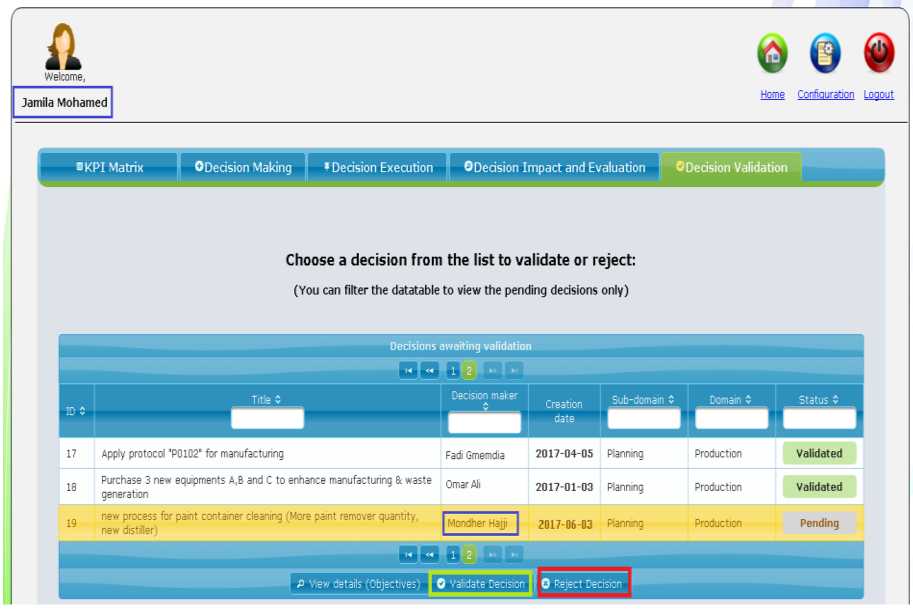

Figure 6.23 Decision validation

6.5.7 Updating the execution status of the decision

Once the decision is validated by the decision-validator "Jamila Mohamed", it can be shown in the decision execution interface. Any decision-maker with or without validation privileges can change the status of the decision from "*Not Executed*" to "*Executed*" and will have to enter the date of when the decision was executed on the ground. This step is shown in figure 6.24.

Regularly updating the decision execution and setting the right execution date is essential for the decision evaluation because the decision will be evaluated on the KPI impact from its execution date to the maximum deadline of its objectives.

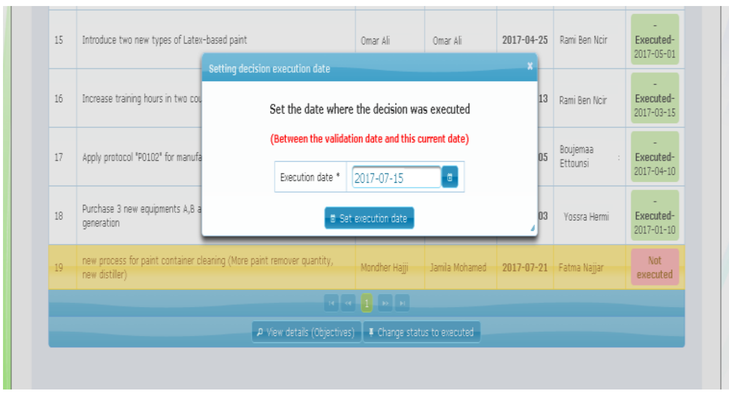

Figure 6.24 Update execution status

6.5.8 Evaluate decision

Once the decision's maximum deadline was reached, the decision can be evaluated based on its direct and indirect impact on KPIs.

Figure 6.25 shows the decision impact and evaluation interface where all evaluated decisions are displayed with their respective sustainability evaluations, actors and its average impact on KPIs calculated using the evaluation formula. An info message appears stating that: "Decision [ID= 19] was evaluated today [2017-07-21]" because the all the deadlines of the objective KPIs were reached at 2017-07-20.

The container cleaning decision was evaluated as "Sustainable" because it recorded a positive average impact on KPIs which is greater than 10%.

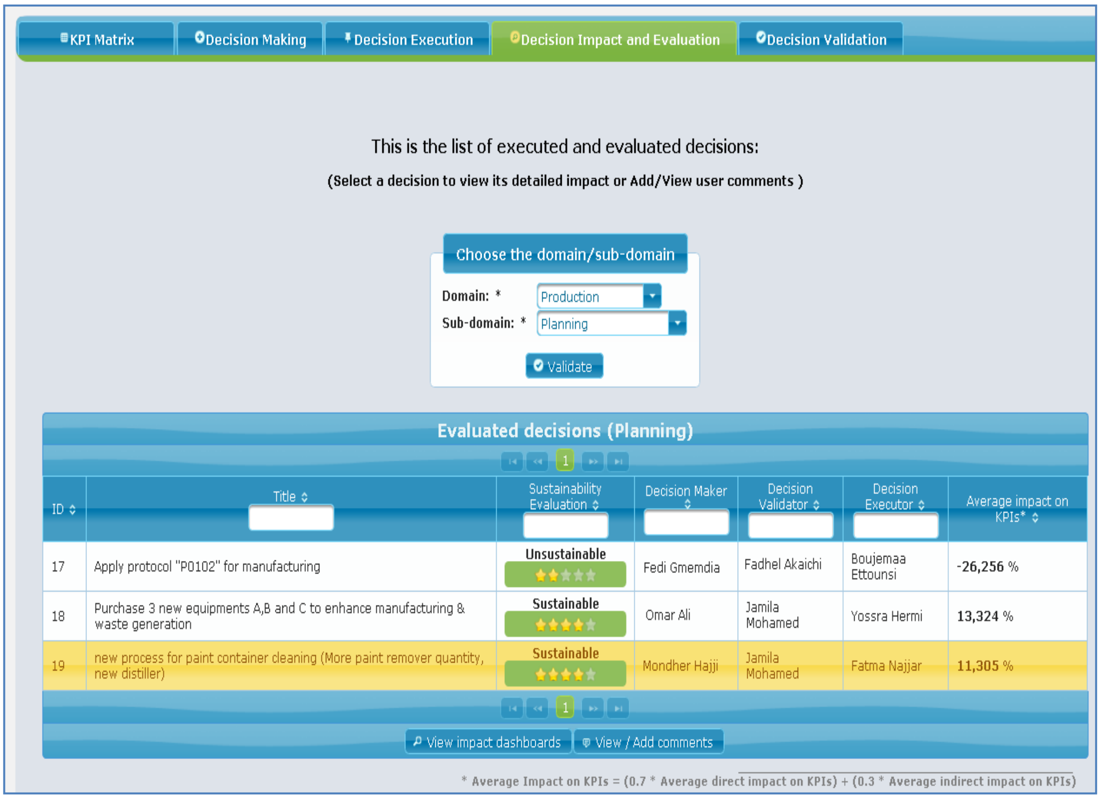

Figure 6.25 Display decision evaluation

The detailed impact of the decision on KPIs can be displayed by selecting the decision and clicking "View impact dashboards" in the evaluation table. The impact dashboard is shown like in figure 6.26.

Impact dashboard							
Sustainability nature	**Affected KPI**	**Measure**	**Desired variation**	**Planned objective**	**Before decision**	**After decision**	**Impact**
Ecological	VOC Emission per cleaning operation	Grams	Decrease ↓	Planned value: 5,35 Deadline: 2017-07-20	8,7 2017-06-01	7,1 2017-07-15	18,391 % Enhancement
	Recyclable paint generation amount per cleaning operation	Kilogram	Increase ↑	Planned value: 6,7 Deadline: 2017-07-20	4,7 2017-06-01	6,8 2017-07-15	44,681 % Enhancement
	Recyclable paint generation amount per container cleaning	Kilogram	Increase ↑	Planned value: 0,75 Deadline: 2017-07-20	0,4 2017-06-01	0,7 2017-07-15	75 % Enhancement
	Non-reusable waste generation per cleaning operation	Kilogram	Decrease ↓	Planned value: 4 Deadline: 2017-07-20	8,41 2017-06-01	5,97 2017-07-15	29,013 % Enhancement
Economical	Average cost per cleaning operation	TND	Decrease ↓	Planned value: 470 Deadline: 2017-07-20	800 2017-06-01	510 2017-07-15	36,25 % Enhancement
	Average time taken per cleaning operation	Minutes	Decrease ↓	Planned value: 175 Deadline: 2017-07-20	225 2017-06-01	191 2017-07-15	15,111 % Enhancement
	Cost of paint remover per cleaning operation	TND	Decrease ↓	Indirect impact	400 2017-06-01	550 2017-07-15	37,5 % Deterioration
	Monhtly maintenance cost of distillers	TND	Decrease ↓	Planned value: 35,5 Deadline: 2017-07-20	45 2017-06-01	59 2017-07-15	31,111 % Deterioration
Social	Monthly number of complaints for cleaning operators	Units	Decrease ↓	Planned value: 2 Deadline: 2017-07-20	2,9 2017-06-01	1,5 2017-07-15	48,276 % Enhancement
	Monthly Number of sick leave days in the production dept.	Units	Decrease ↓	Planned value: 0,6 Deadline: 2017-07-20	7 2017-06-01	4 2017-07-15	42,857 % Enhancement
	Satisfaction rate of cleaning operators	Units [0..10]	Increase ↑	Planned value: 7,5 Deadline: 2017-07-20	4,8 2017-06-01	6,9 2017-07-15	43,75 % Enhancement

Figure 6.26 Display detailed impact on KPIs

The user can see all the KPIs that were affected positively or negatively in the period between the decision's execution date [2017-06-15] and its maximum deadline [2017-07-20]. The container cleaning decision:

- Significantly affected the ecological KPIs positively (decreased VOC emissions, increased generation of recycled paint, decreased non reusable wastes generation).
- Had a varying impact on economical KPIs. It positively affected the cleaning operations' average cost and time but the average cost of paint remover and the monthly maintenance cost of distillers were increased. The reason behind this negative affection is the increased amount of usage for the paint remover quantity after the decision and the new, performant distiller requires more maintenance cost and effort.

Although, the KPI labelled "Cost of paint remover per cleaning operation" was not set as an objective by the decision so it was an indirect affection. The importance of the indirect impact is always lower than the direct impact's so this will not affect the decision evaluation severely.

- Significantly affected the social KPIs positively (Decreased number of complaints and sick leaves of the cleaning operators and the production department, Increased satisfaction rate of the cleaning operators).

6.5.9 Comment on evaluated decision

The evaluated decision can be commented and evaluated by users to further understand its impact from the stakeholders' perspectives. The average user rating of the decision is displayed for each decision in the comments section. Figure 6.27 displays the comment section of the container cleaning decision.

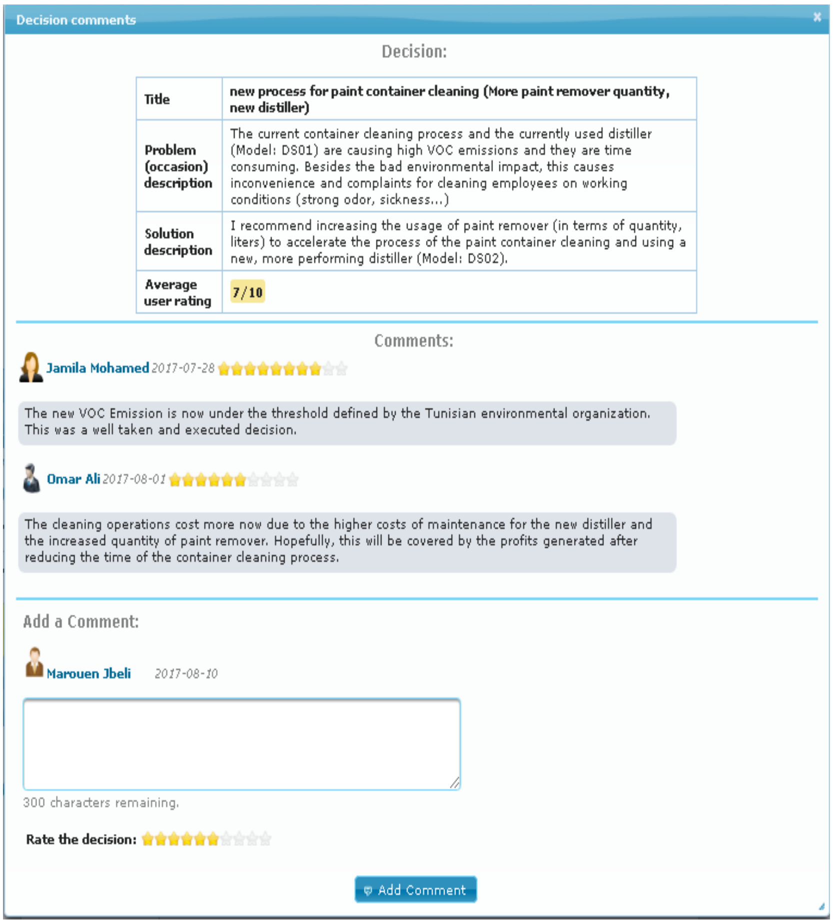

Figure 6.27 Comment on decision

6.5.10 Summary

In this section, a business case were presented for a company specialized in paint and coating manufacturing, namely, Intercolor. The proposed DES for sustainability will be one of the efforts for the company to contribute in reducing the negative impact on the environment that accompany the paint industry.

The business case was presented in a narrative and user guide-like method. It shows (with screenshots) all the steps of the decision-making process that are included in the DES:

- KPI identification (in addition to add and edit KPIs).
- Context definition (defining problem, objectives, scopes).
- Recommended decisions exploration (retrieved with the CbR embedded in the DES).
- Decision creation (along with decision validation and execution).
- Decision evaluation (automatic evaluation based on KPI impact in addition to user evaluation).

7 Conclusion and Outlook

Nowadays, our world is considered endangered in terms of environment preserving and sustainability. The facts presented by environmental organizations are terrifying and that urges them, along with governments, to push businesses and organizations towards more sustainable decisions and business processes (Rezgui and Marx Gómez, 2016a).

Assuring sustainability through conscious decision-making is an important key for any organization to contribute in the protection of the environment, economy and society. It helps in securing better conditions for future generations. Making sustainable decisions concerns not only altering the current business process but also exploring new opportunities and original ideas. Organizations are using set of tools, techniques and processes with the aim to support and improve their decision-making and to meet their objectives. The terms DSS and EMIS encompass this approach. The combination between DSS and EMIS oriented the organizations objectives towards sustainability. While DSS collect, aggregate, filter, harmonize and present the information in a user-friendly form, EMIS provide the input (raw data/information) for DSS namely the environmental one.

Truthfully, both systems (DSS and EMIS) play an important role in assuring support by managing the internal and external data and providing information about the organizations and especially the environmental ones. The main advantages of such systems are simplicity, clarification and control. They help the decision-making activities through presenting several indicators delineating the processes performance. However they lack on the decisions evaluation that is a very important part. Since the decision-making process is not limited only on the preparation phase, but also in the making and evaluation. Information about the decision itself and its sustainability evaluation are actually not considered in the existing approaches. Experience and intuition are neither stored nor shared; they are only in the minds of individual managers.

The understanding of the impact and the cause-and-effect relationship between a given decision and its sustainability impact is not granted by just gathering and presenting data. The decisions need to be evaluated and tracked in order to enhance them and support more reasonable and sustainable decisions in the future.

This research work respond the question: How can decisions be evaluated based on their sustainability impact. It deals with the idea of managing decisions through the archiving, tracking, evaluation, recommendation and sharing. It offers organizations a decision-making process analysis to improve their sustainability. Process analysis in BPI enables users to analyse completed process executions, it can be helpful for business analysts to find correlations between different workflow aspects and performance metrics (Castellanos et al., 2009) (e.g. high additional costs occur while shipping paint products in north west regions...). This helps in identifying opportunities for process optimization (Castellanos et al., 2005) whose aim is the generation of decision models that optimize some aspect(s) of the operation of a process.

In order to enable the evaluation of decisions based on their sustainability impact this work presents the criteria that should be taken into account and how they should be arranged.

The evaluation is dependent upon the degree of achievement in attaining the agreed-upon quantitative and qualitative goals set previously. The goals are presented in different indicators classified by their nature into three categories: economical, ecological and social. Each indicator is assigned to a domain (logistics, human resources, sales, production, etc.) and a sub-domain (transport, after-sales, purchasing, training, etc.). The evaluation of a taken decision is assured after setting the target values and deadline for achievement to one or many types of indicators. There are two types of evaluation (i) direct impact measurement and (ii) indirect impact measurement. For the first one the selected goals (represented by indicators) are tracked and the degree of achievement in attaining is measured. In addition the others indicators (indirect impact) in the same domains and sub-domains are also tracked and evaluated (enhancement or weakening).

The proposed DES enables the process analysis by enabling decision-makers to view and analyse the evaluation and impact of past decisions (past executed processes) and identify their advantages and shortcomings. Process optimization is related to the identification of the areas of improvements in particular processes. The DES can relate to this with the proposed green decision-making process. A decision-maker can enter the current problems or objectives and the system will display the recommended past decisions that can answer to entered parameters.

In addition to the process presented in this work, an abstraction of the system was elaborated via the design of a software architecture. It represents a “first cut” at solving the problem and designing the system (Northrop, 2003). It defines the system elements, components with their properties and the interaction among them. The DES software architecture was developed to respond to the functional and non-functional requirements defined previously in close cooperation with both scientific communities and practitioners. On the top of the proposed software architecture a prototypical implementation (fully functional prototype) was conducted to demonstrate how the beforehand elaborated theoretical considerations can be practically realised (proof of concept). This enables less abstraction of the proposed system. The conducted proof of concept (PoC) enables the testing and the measurement of its operational feasibility. This allows to move serenely in the implementation of the DES, to make the necessary adjustments in order to ensure its success.

In order to avoid the “tunnel effect”, the fact of seeing the result only at the final delivery and nothing or almost nothing throughout the development phase, an iterative approach in form of generate/test cycle; as recommended by (Hevner et al., 2004) in the guideline 6 design as a search process; was followed. This iterative approach was necessary in the design and implementation of the DES software, then design science is inherently iterative (Hevner et al., 2004). The agility offered by this approach enhances the ability to respond to change, and even to promote it, in order to adapt the design science output (IT artifacts) better to the environment. The benefits of such approach can be summarized in the following points:

- Reducing the complexity of software development.
- Better reactivity to adjust the requirements, the designed and implemented artifacts (the DES: process, architecture and software) to the needs of the end users.
- Increase the fluidity of deliveries and the velocity.

One of the potential extension of the DES is the recommendation component. This first version of the CBR mechanism used in the decision recommendation could be improved. Currently, we are using a syntax-based algorithm to retrieve the similar situations that led to a certain decision without taking into account the meaning of each decision and the circumstances around it. We are planning to use the content of our first decision database enriched by user experience to build an ontology which can help us identify automatically the semantic closeness between the previous situations and the new one. Several alternatives are available to do so and we will study the strengths and weaknesses of each one of them:

- Manual construction using one of the available editors: Protégé, PLibEditor, OntoEdit…etc.
- Automatic construction from text (Text2Onto, OntoCASE…etc.).
- Cooperative construction (KA^2, CO4).

Many IT giants like google, Microsoft (LinkedIn, Bing) and Yahoo are using semantic-search to enhance the retrieval of results.

In contrast to these rather technical topics, another more theoretical research topic can be seen as a potential outlook. For instance, it may be valuable to investigate the acceptance of decision-makers to share knowledge and communicate their decisions with each other. Then the basis of an open communication is trust on the ability to handle with information and knowledge. This confidence is built cumulatively in long term but can be dissipated quickly (Northrop, 2003). Having the appropriate technology and solution to enhance the quality of decisions in term of sustainability cannot be assured without having the appropriate motivation and culture to share and communicate. Then the nature of communication has a great impact on employee motivation and commitment (Stehle and Mücke, 2009).

Reference List

Aamodt, A., Plaza, E., 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Commun. 7, 39–59.

Abelló, A., Romero, O., 2009. On-line analytical processing, in: Encyclopedia of Database Systems. Springer, pp. 1949–1954.

Ambler, S.W., 2004. The object primer: Agile model-driven development with UML 2.0. Cambridge University Press.

Ambler, S.W., 2001. The Object Primer 2nd Edition: The Application Developer's Guide to Object Orientation. New York: Cambridge University Press. http://www. ambysoft. com/theObjectPrimer. html.

Ambler, S.W., 1998a. Process Patterns: Building Large-Scale Systems Using Object Technology. Cambridge University Press.

Ambler, S.W., 1998b. Building Object Applications That Work: Your Step-By-Step Handbook for Developing Robust Systems with Object Technology. Cambridge University Press.

Arnott, D., Pervan, G., 2008. Eight key issues for the decision support systems discipline. Decis. Support Syst. 44, 657–672.

Arnott, D., Pervan, G., 2005a. A critical analysis of decision support systems research. J. Inf. Technol. 20, 67–87. https://doi.org/10.1057/palgrave.jit.2000035

Arnott, D., Pervan, G., 2005b. A critical analysis of decision support systems research. J. Inf. Technol. (20)2, 67–87.

Arras, P., Steinhoff, A., 2008. Method for online analytical processing (OLAP). U.S. Patent No. 7,340,476.

Arvai, J., Campbell-Arvai, V., Steel, P., 2012. Making sustainable choices: a guide for manager. Network for Business Sustainability, University of Calgary, Canada.

Baars, H., Kemper, H.-G., 2008a. Management support with structured and unstructured data—an integrated business intelligence framework. Inf. Syst. Manag. 25, 132–148.

Baars, H., Kemper, H.-G., 2008b. Management support with structured and unstructured dataóan integrated business intelligence framework. Inf. Syst. Manag. 25, 132–148.

Baskerville, R.L., Myers, M.D., 2009. Fashion waves in information systems research and practice. Mis Q. 33, 647–662.

Bass, L., Clements, P., Kazman, R., 2003. Software Architecture in Practice. 2003. Addison Wesley, Reading, USA.

Bauer, A., Schmid, T., 2009. Was macht Operational BI aus. BI-Spektrum 4, 13–14.

Benbasat, I., Zmud, R.W., 1999. Empirical research in information systems: the practice of relevance. MIS Q. 23, 3–16.

Beynon-Davies, P., 2004. On-Line Analytical Processing, in: Database Systems. Springer, pp. 1949–1954.

Black, A.D., Car, J., Pagliari, C., Anandan, C., Cresswell, K., Bokun, T., McKinstry, B., Procter, R., Majeed, A., Sheikh, A., 2011. The impact of eHealth on the quality and safety of health care: a systematic overview. PLoS Med 8, e1000387.

Black, D., 1948. On the rationale of group decision-making. J. Polit. Econ. 56, 23–34.

Blenko, Marcia W., Mankins, M.C., Rogers, P., 2010. The decision-driven organization. Harv. Bus. Rev. 88, 54–62.

Blenko, Marcia W, Mankins, M.C., Rogers, P., 2010. The decision-driven organization. Harv. Bus. Rev. 88, 54–62.

Blumberg, R., Atre, S., 2003a. The problem with unstructured data. Dm Rev. 13, 62.

Blumberg, R., Atre, S., 2003b. The problem with unstructured data. Dm Rev. 13, 62.

Bouyssou, D., Marchant, T., Pirlot, M., Tsoukiàs, A., Vincke, P., 2015. Aiding to Decide: Concepts and Issues, in: Evaluation and Decision Models with Multiple Criteria. Springer, pp. 17–34.

Brundtland, G.H., 1987. Report of the World Commission on environment and development:" our common future.". United Nations, Oxford.

Burmester, L., 2011. Adaptive Business-Intelligence-Systeme. Springer.

Castellanos, M., Casati, F., Sayal, M., Dayal, U., 2005. Challenges in business process analysis and optimization, in: International Workshop on Technologies for E-Services. Springer, pp. 1–10.

Castellanos, M., De Medeiros, A.A., Mendling, J., Weber, B., Weijters, A., 2009. Business process intelligence, in: Handbook of Research on Business Process Modeling. IGI Global, pp. 456–480.

Chaudhuri, S., Dayal, U., Narasayya, V., 2011. An overview of business intelligence technology. Commun. ACM 54, 88–98.

Chung, L., do Prado Leite, J.C.S., 2009. On non-functional requirements in software engineering, in: Conceptual Modeling: Foundations and Applications. Springer, pp. 363–379.

Codd, E.F., Codd, S.B., Salley, C.T., 1993. Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. Codd Date 32.

Cooper, B.L., Watson, H.J., Wixom, B.H., Goodhue, D.L., 2000. Data warehousing supports corporate strategy at First American Corporation. MIS Q. 547–567.

Couckuyt, D., Van Looy, A., De Backer, M., 2017. Sustainability Performance Measurement, in: International Conference on Business Process Management. Springer, pp. 520–524.

Cronholm, S., Göbel, H., 2014. The Need for Empirical Grounding of Design Science Research Methodology, in: 3rd International Workshop on IT Artefact Design & Workpractice Improvement, 2 June, 2014, Friedrichshafen, Germany.

Cronk, M., Gurteen, D., 2012. Social capital, knowledge sharing and intellectual capital in the web 2.0 enabled world. Lead. Issues Soc. Knowl. Manag.

Damij, N., 2007. Business process modelling using diagrammatic and tabular techniques. Bus. Process Manag. J. 13, 70–90.

Davenport, T.H., Harris, J.G., David, W., Jacobson, A.L., 2001. Data to knowledge to results: building an analytic capability. Calif. Manage. Rev. 43, 117–138.

Davenport, T.H., Prusak, L., 1998. Working knowledge: How organizations manage what they know. Harvard Business Press.

Davis, J.R., Clark, J.L., 1989. A selective bibliography of expert systems in natural resource management 3(3): 1-18 203.

Deepak, A., Crupi, J., Malks, D., 2001. Core J2EE Patterns: Best Practices and Design Strategies. Sun Microsyst. Palo Alto.

Divya, S., Premalatha, S., 2016. Fault Tolerant using Shared Queues of NOC Router. In International Journal of VLSI system design and communication systems 4, 165–168.

Douglas, W., 2012. Top Features Of A Scalable Database [WWW Document]. URL http://highscalability.com/blog/2012/7/4/top-features-of-a-scalable-database.html#comments

Eckerson, W., 2003. Smart Companies in the 21st Century: The Secrets of Creating Successful Business Intelligence Solutions. DATA Warehous. Inst.

ElKafrawy, P., Mohamed, R., 2014. Comparative Study of Case Based Reasoning Software. Int. J. Sci. Res. Manag. Stud. 1, 224–233.

Fabac, R., 2010. Complexity in Organizations and Environment-Adaptive Changes and Adaptive Decision-Making. Interdiscip. Descr. Complex Syst. 8, 34–48.

Fakeeh, K.A., 2015. Decision Support Systems (DSS) in Higher Education System. Decis. Support Syst. DSS 9.

Fedorowicz, J., 1993. A technology infrastructure for document-based decision support systems, Third Edition. ed. Prentice-Hall, Inc.

Felden, C., Chamoni, P., Linden, M., 2010. From process execution towards a business process intelligence, in: Business Information Systems. Springer, pp. 195–206.

Felden, C., Koschtial, C., Chamoni, P., 2015. Towards an Evaluation Framework to Structure Business Intelligence Project Patterns as Enhancement of Business Intelligence Maturity Models, in: Business Intelligence for New-Generation Managers. Springer, pp. 67–76.

Forst, A., 2011. Why is Knowledge Management Useful? [WWW Document]. Knowl. Manag. Tools. URL http://www.knowledge-management-tools.net/why-is-knowledge-management-useful.html

Forward, A., 2002. Software documentation: Building and maintaining artefacts of communication. University of Ottawa (Canada).

Frost, A., 2017a. Defining Knowledge, Information, Data. [WWW Document]. Defin. Knowl. Inf. Data. URL http://www.knowledge-management-tools.net/knowledge-information-data.html

Frost, A., 2017b. KM Tools By Categoriy (IT-Based). [WWW Document]. Knowl. Manag. Tools. URL http://www.knowledge-management-tools.net/knowledge-management-tools.html

Frost, A., 2012. Knowledge Management Definition. [WWW Document]. Knowl. Manag. Tools. URL http://www.knowledge-management-tools.net/knowledge-management-definition.html

Gallupe, R.B., 2007. The tyranny of methodologies in information systems research 1. ACM SIGMIS Database 38, 20–28.

Gardner, B., 2013. Making sense of Enterprise 2.0. Vine 43, 149–160.

Gilleland, M., 2009. Levenshtein distance, in three flavors. Merriam Park Softw.

Global Reporting Initiative, G.R.I., 2002. Sustainability reporting guidelines. Boston, USA.

Gluchowski, P., Gabriel, R., Dittmar, C., 2008. Management support systeme und business intelligence: Computergestützte Informationssysteme für Fach- und Führungskräfte. Springer, Berlin, Heidelberg.

Gogolev, A., 2013. Response to question: Similarity String Comparison in Java [WWW Document]. URL https://stackoverflow.com/questions/955110/similarity-string-comparison-in-java

Gómez, J.M., Rautenstrauch, C., Cissek, P., 2008. Einführung in Business Intelligence mit SAP NetWeaver 7.0. Springer.

Gorry, G.A., Morton, M.S., 1989. A framework for management information systems. Sloan Manage. Rev. 30, 49–61.

Grabova, O., Darmont, J., Chauchat, J.-H., Zolotaryova, I., 2010. Business intelligence for small and middle-sized entreprises. ACM SIGMOD Rec. 39, 39–50.

Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M., Shan, M.-C., 2004. Business process intelligence. Comput. Ind. 53, 321–343.

Grimes, S., 2014. Unstructured Data and the 80 Percent Rule (2008). Clarabridge Bridg.

Grünig, R., Kühn, R., 2005. Successful decision-making: A systematic approach to complex problems. Springer-Verlag.

Guldner, A., Garling, M., Morgen, M., Naumann, S., Kern, E., Hilty, L.M., 2018. Energy Consumption and Hardware Utilization of Standard Software: Methods and Measurements for Software Sustainability, in: From Science to Society. Springer, pp. 251–261.

Han, J., Pei, J., Kamber, M., 2011. Data mining: concepts and techniques. Elsevier.

Hendriks, P., 1999. Why share knowledge? The influence of ICT on the motivation for knowledge sharing. Knowl. Process Manag. 6, 91.

Herzig, C., Godemann, J., 2010. Internet-supported sustainability reporting: Developments in Germany. Manag. Res. Rev. 33, 1064–1082.

Hevner, A., Chatterjee, S., 2015. Design science research in information systems. Assoc. Inf. Syst. Ref. Syllabi Ed J Vom Brocke Eduglopediaorg 2015 1–9.

Hevner, A.R., March, S.T., Park, J., Ram, S., 2004. Design science in information systems research. MIS Quarterly Vol. 28, 75–105.

Hilty, L.M., 2011. Information technology and sustainability: Essays on the relationship between information technology and sustainable development. BoD–Books on Demand.

Hilty, L.M., Aebischer, B., 2015a. Ict for sustainability: An emerging research field, in: ICT Innovations for Sustainability. Springer, pp. 3–36.

Hilty, L.M., Aebischer, B., 2015b. ICT innovations for sustainability. Springer.

Hilty, L.M., Hercheui, M.D., 2010. ICT and sustainable development, in: What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience. Springer, pp. 227–235.

Hilty, L.M., Seifert, E.K., Treibert, R., 2005. Information systems for sustainable development. IGI Global.

Hintemann, R., Clausen, J., 2016. Green cloud? The current and future development of energy consumption by data centers, networks and end-user devices. Proc. ICT Sustain.

Holsapple, C.W., Whinston, A.B., 2001. Decision support systems: a knowledge-based approach. Stud. Inform. Control 10, 73–76.

Hosack, B., Hall, D., Paradice, D., Courtney, J.F., 2012. A look toward the future: decision support systems research is alive and well. J. Assoc. Inf. Syst. 13, 315.

Hosny, H., 2009. Business Process Intelligence. ATIT 2009 Cairo 2009 Dimens. Bus. Process Intell. 213.

Iivari, J., 2003. The IS core-VII: Towards information systems as a science of meta-artifacts. Commun. Assoc. Inf. Syst. 12, 37.

Improvement Service, K.T., 2009. Knowledge management tools and techniques briefing.

Intelligence, P., 2017. The Magic in a 0-to-10 Rating Scale. Prim. Intell.

Ipe, M., 2003. Knowledge sharing in organizations: A conceptual framework. Hum. Resour. Dev. Rev. 2, 337–359.

Ireland, R.D., Miller, C.C., 2004. Decision-making and firm success. Acad. Manag. Exec. 18, 8–12.

Isenmann, R., Brosowski, J., Schappert, M., Marx Gómez, J., 2005. Further developments of a software tool providing tailored sustainability reports. J H Ebi Ek J Rá Ek Eds Netw. Environ. Inf. 684–691.

Jamous, N., Schrödl, H., Turowski, K., 2013. Light-weight composite environmental performance indicators (LWC-EPI) solution: a systematic approach towards users requirements, in: System Sciences (HICSS), 2013 46th Hawaii International Conference On. IEEE, pp. 945–954.

Jelschen, J., Küpker, C., Winter, A., Sandau, A., Wagner vom Berg, B., Marx Gómez, J.M., 2016. Towards a Sustainable Software Architecture for the NEMo Mobility Platform, in: Proceedings of the 30th International Conference on Environmental Informatics Stability, Continuity, Innovation: Current Trends and Future Perspectives Based On. Berlin.

Jøsang, A., Guo, G., Pini, M.S., Santini, F., Xu, Y., 2013a. Combining recommender and reputation systems to produce better online advice, in: International Conference on Modeling Decisions for Artificial Intelligence. Springer, pp. 126–138.

Jøsang, A., Guo, G., Pini, M.S., Santini, F., Xu, Y., 2013b. Combining recommender and reputation systems to produce better online advice, in: International Conference on Modeling Decisions for Artificial Intelligence. Springer, pp. 126–138.

Juneau, J., 2014. PrimeFaces in the Enterprise: Build data-driven applications for the enterprise using the PrimeFaces JavaServer Faces UI framework. Oracle Technol. Netw.

Kahneman, D., 2003. Maps of bounded rationality: Psychology for behavioral economics. Am. Econ. Rev. 93, 1449–1475.

Kämpgen, B., O'Riain, S., Harth, A., 2012. Interacting with statistical linked data via OLAP operations, in: Extended Semantic Web Conference. Springer, Berlin, Heidelberg, pp. 87–101.

Kannan, R., 2009. Uncertainties in key low carbon power generation technologies–implication for UK decarbonisation targets. Appl. Energy 86, 1873–1886.

Kateule, R., Winter, A., 2016. Viewpoints for Sensor based Environmental Information Systems. Presented at the EnviroInfo 2016, pp. 211–217.

Kawamoto, K., Houlihan, C.A., Balas, E.A., Lobach, D.F., 2005. Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. Br. Med. Assoc. 330, 765.

Kemper, H.-G., Baars, H., Mehanna, W., 2016. Business Intelligence-Grundlagen und praktische Anwendungen: eine Einführung in die IT-basierte Managementunterstützung, 4th ed. Springer Vieweg.

Kemper, H.-G., Mehanna, W., Unger, C., 2006. Business Intelligence-Grundlagen und praktische Anwendungen: eine Einführung in die IT-basierte Managementunterstützung. Vieweg.

Kersten, G.E., Lo, G., 2002. DSS application areas, in: Decision Support Systems for Sustainable Development. Springer, pp. 391–407.

Kim, J., Hwang, M., Jeong, D.-H., Song, S.-K., Jung, H., 2013. Business Intelligence Service based on Adaptive User Modelling and Grouping. J. Comput. Sci. 9, 1396.

Kimball, R., Ross, M., 1996. The Data Warehouse Toolkit - The Complete Guide to Dimensional Modeling. Toolkit John Wiley.

Kipyegen, N.J., Korir, W., Njoro, K., 2013. Importance of software documentation. Int. J. Comput. Sci. Issue 10, 223–228.

Kwahk, K.-Y., Park, D.-H., 2016. The effects of network sharing on knowledge-sharing activities and job performance in enterprise social media environments. Comput. Hum. Behav. 55, 826–839.

Lai, C., Gong, L., Koved, L., Nadalin, A., Schemers, R., 1999. User authentication and authorization in the Java/sup TM/platform, in: Computer Security Applications Conference, 1999.(ACSAC'99) Proceedings. 15th Annual. IEEE, pp. 285–290.

Lau, R.Y., Liao, S.S., Wong, K.-F., Chiu, D.K., 2012. Web 2.0 Environmental Scanning and Adaptive Decision Support for Business Mergers and Acquisitions. MIS Q. 36.

Lewis, L.F., 2010. Group support systems: overview and guided tour, in: Handbook of Group Decision and Negotiation. Springer, pp. 249–268.

Lightsey, B., 2001. Systems engineering fundamentals. DEFENSE ACQUISITION UNIV FT BELVOIR VA.

Lindvall, J., 2013. Affärssystem och ekonomistyrning. Perspekt. Pa Ekon.

Liu, D., Mei, H., 2003. Mapping Requirements to Software Architecture by Feature-Orientation., in: STRAW. pp. 69–76.

Loos, P., Nebel, W., Gómez, J.M., Hasan, H., Watson, R.T., vom Brocke, J., Seidel, S., Recker, J., 2011. Green IT: a matter of business and information systems engineering? Bus. Inf. Syst. Eng. 3, 245–252.

Ma, Z., Wang, H., Wu, A., Zeng, G., Tu, X., 2014. An intelligent decision support system for residential energy consumption and renewable energy utilization in rural China. Energy Sources Part B Econ. Plan. Policy 9, 374–382.

March, J.G., 1987. Ambiguity and accounting: The elusive link between information and decision making. Account. Organ. Soc. 12, 153–168.

March, S.T., Smith, G.F., 1995. Design and natural science research on information technology. Decis. Support Syst. 15, 251–266.

Marx Gómez, J., 2004. Automatisierung der Umweltberichterstattung mit Strommanagementsystemen, F. f. I. d. O. M. H. PhD Thesis, Ed. University of Magdeburg.

Marx Gómez, J., Teuteberg, F., 2015. Toward the Next Generation of Corporate Environmental Management Information Systems: What is Still Missing?, in: ICT Innovations for Sustainability. Springer International Publishing, pp. 313–332.

Meisch, S., Hagemann, N., Geibel, J., Gebhard, E., Drupp, M.A., 2015. Indicator-based analysis of the process towards a university in sustainable development: A case study of the University of Tübingen (Germany), in: Integrative Approaches to Sustainable Development at University Level. Springer, pp. 169–183.

Michalewicz, Z., Michalewicz, M., Schmidt, M., Chiriac, C., 2007. Adaptive Business Intelligence. Springer-Verlag.

Ministère de l'Économie et des Finances, G., 2014. Valorisation de l´information dans l´entreprise.

Müller-Christ, G., 2011. Nachhaltigkeit in der Hochschule: Ein Konzept für die interne Selbstüberprüfung. Hochschulen Für Eine Nachhalt. Entwickl. Nachhalt. Forsch. Lehre Betr. Dtsch. UNESCO-Komm. 73.

Müller-Christ, G., Sterling, S., van Dam-Mieras, R., Adom\s sent, M., Fischer, D., Rieckmann, M., 2014. The role of campus, curriculum, and community in higher education for sustainable development–a conference report. J. Clean. Prod. 62, 134–137.

National Research Council, C., 2014. Sustainability concepts in decision-making: Tools and approaches for the us environmental protection agency. National Academies Press.

Naumann, S., Kern, E., Dick, M., Johann, T., 2015. Sustainable software engineering: Process and quality models, life cycle, and social aspects, in: ICT Innovations for Sustainability. Springer, pp. 191–205.

Nenortaitė, J., Butleris, R., 2015. Improving business rules management through the application of adaptive business intelligence technique. Inf. Technol. Control 38.

Nenortaitė, J., Butleris, R., 2009. Improving business rules management through the application of adaptive business intelligence technique. Inf. Technol. Control 38, 21–28.

Nobel Prize Organization, n.d. . Nobelprize.

North, K., 2016. Interne Kommunikationssysteme und Wissensmanagement im Wandel, in: Personalperspektiven. Springer, pp. 267–289.

Northrop, L., 2003. The importance of software architecture. Softw. Eng. Inst. Carnegie Mellon Univ. Available Httpsunset Usc Edugsawgsaw2003s13northrop Pdf.

Object Management Group, 2011. Notation (BPMN) version 2.0, Model Business Process. OMG Specif. Object Manag. Group.

Oboni Riskope Associates, I., 2014. Let's define Strategic, Tactical and Operational planning.

Omotayo, F.O., 2015. Knowledge Management as an important tool in Organisational Management: A Review of Literature. Libr. Philos. Pract. 1.

Oracle, C., 2017. Java 2 Platform, Enterprise Edition (J2EE) FAQ [WWW Document]. Java EE. URL http://www.oracle.com/technetwork/java/javaee/j2ee-jsp-141833.html (accessed 3.15.17).

Oracle, C., 2014. Java Platform, Enterprise Edition: The Java EE Tutorial [WWW Document]. Java Platf. Enterp. Ed. Java EE 7. URL https://docs.oracle.com/javaee/7/tutorial/overview003.htm (accessed 3.15.17).

Oreizy, P., Heimbigner, D., Johnson, G., Gorlick, M.M., Taylor, R.N., Wolf, A.L., Medvidovic, N., Rosenblum, D.S., Quilici, A., 1999. An architecture-based approach to self-adaptive software. IEEE Intell. Syst. 14, 54–62.

Orlikowski, W.J., Iacono, C.S., 2001. Research commentary: Desperately seeking the "IT" in IT research—A call to theorizing the IT artifact. Inf. Syst. Res. 12, 121–134.

Österle, H., Winter, R., Brenner, W., 2010. Gestaltungsorientierte Wirtschaftsinformatik: Ein Plädoyer für Rigor und Relevanz. Infowerk.

Oxford Dictionary, 2017. Definition of therm design.

Page-Jones, M., 1995. What every programmer should know about object-oriented design. Dimensions (Wash.) 227, 252–343.

Parmenter, D., 2015. Key performance indicators: developing, implementing, and using winning KPIs. John Wiley & Sons.

Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S., 2007. A design science research methodology for information systems research. J. Manag. Inf. Syst. 24, 45–77.

Pendse, N., 2006. OLAP architectures. OLAP Rep.

Petkovska, S., Gjorgjeska, B., 2013. The significance of the quality management system in making management decisions, in: The Third International Scientific Congress-Biennale. University of Tourism and Management, pp. 25–28.

Phillips-Wren, G., Mora, M., Forgionne, G.A., Gupta, J.N., 2009. An integrative evaluation framework for intelligent decision support systems. Eur. J. Oper. Res. 195, 642–652.

Power, D.J., Sharda, R., Burstein, F., 2015. Decision Support Systems, in: Management Information Systems. Wiley Encyclopedia of Management.

Primary Intelligence in Excellent research methodologies, 2017. The Magic in a 0-to-10 Rating Scale. [WWW Document]. URL https://www.primary-intel.com/blog/the-magic-in-a-0-to-10-rating-scale/ (accessed 9.19.17).

Rao, R., 2003a. From unstructured data to actionable intelligence. IT Prof. 5, 29–35.

Rao, R., 2003b. From unstructured data to actionable intelligence. IT Prof. 5, 29–35.

Rautenstrauch, C., 2013. Betriebliche Umweltinformationssysteme: Grundlagen, Konzepte und Systeme. Springer-Verlag.

Reynolds, K.M., Hessburg, P.F., Bourgeron, P.S., 2014. Making Transparent Environmental Management Decisions. Springer Berlin Heidelberg.

Rezaie, K., Ansarinejad, A., Haeri, A., Nazari-Shirkouhi, A., Nazari-Shirkouhi, S., 2011. Evaluating the Business Intelligence Systems Performance Criteria Using Group Fuzzy AHP Approach, in: Computer Modelling and Simulation (UKSim), 2011 UkSim 13th International Conference On. IEEE, pp. 360–364.

Rezgui, A., Ben Maaouia, R., 2016. KPI-based Decision Impact Evaluation System for Adaptive Business Intelligence. Ingénierie des Systèmes d'Information 21, 103–124. https://doi.org/DOI:10.3199/JESA.45.1-n

Rezgui, A., Marx Gómez, J., 2017. Decisions Sustainability Evaluation: software architecture. Acad. Star Publ. 8, 124–132. https://doi.org/10.15341/jbe(2155-7950)/02.08.2017/002

Rezgui, A., Marx Gómez, J., 2016a. Toward Green Decision Making Through Decision Evaluation System, in: Environmental Informatics Stability, Continuity, Innovation: Current Trends and Future Perstectives Based on 30 Years of History. Presented at the EnviroInfo 2016, pp. 55–63.

Rezgui, A., Marx Gómez, J., 2016b. Measurement of Decisions Sustainability - Indicators classification. Presented at the Conference Proceedings: Dialogue on Sustainability and Environmental Management, Accra, Ghana.

Rezgui, A., Marx Gómez, J., Ben Maaouia, R., 2017. KPI-Based Decision Evaluation System to Enhance QMSs for Higher Educational Institutes. Int. J. Decis. Support Syst. Technol. IJDSST 9, 39–55.

Rezgui, A., Marx Gómez, J., Hajji, M.A., 2018. Recommendation of sustainable decisions within a decision evaluation system using case-based reasoning, in: 10. BUIS-Tage. Presented at the Betriebliche Umweltinformationssysteme für Smart Cities und Smart Regions, Oldenburg, Germany.

Rezgui, A., Naana, M., 2010. Improving of environmental management accounting system for support the environmental information management, in: Integration of Environmental Information in Europe. Presented at the EnviroInfo2010, p. 187.

Rod DB, L., Bechstedt, H.-D., Rais, M., 2000. Indicators for sustainable land management based on farmer surveys in Vietnam, Indonesia, and Thailand. Agriculture, ecosystems & environment 81, no (2), 137–146.

Rodriguez, R.R., Saiz, J.J.A., Bas, A.O., 2009. Quantitative relationships between key performance indicators for supporting decision-making processes. Comput. Ind. 60, 104–113.

Rodríguez, R.R., Saiz, J.J.A., Bas, Á.O., Sáez, M.J.V., 2010. Identifying relationships between key performance indicators, in: 4th International Conference On Industrial Engineering and Industrial Management. pp. 151–159.

Roozenburg, N.F., Eekels, J., 1995. Product design: fundamentals and methods. Wiley Chichester.

Rosemann, M., Vessey, I., 2008. Toward improving the relevance of information systems research to practice: the role of applicability checks. Mis Q. 32, 1–22.

Roth-Berghofer, T.R., 2003. Knowledge maintenance of case-based reasoning systems: the SIAM methodology. IOS Press.

Rowley, J., 2000. Is higher education ready for knowledge management? Int. J. Educ. Manag. 14, 325–333.

Rumbaugh, J., Jacobson, I., Booch, G., 2004. Unified modeling language reference manual, the. Pearson Higher Education.

Russom, P., 2007. BI search and text analytics. TDWI Best Pract. Rep. 9–11.

Safwan, E.R., Meredith, R., Burstein, F., 2016. Towards a Business Intelligence Systems Development Methodology: Drawing on Decision Support and Executive Information Systems.

Sarkheyli, A., Söffker, D., 2015. Case indexing in Case-Based Reasoning by applying Situation Operator Model as knowledge representation model. IFAC-Pap. 48, 81–86.

Sauer, J., 2016. Scheduling regarding energy efficiency. 30 Workshop Plan. Sched. Konfigurieren Entwerf. 2016.

Schnackenberg, Andrew K., Tomlinson, E.C., 2016. Organizational transparency: A new perspective on managing trust in organization-stakeholder relationships. J. Manag. 42, 1784–1810.

Schnackenberg, Andrew K, Tomlinson, E.C., 2016. Organizational transparency: A new perspective on managing trust in organization-stakeholder relationships. J. Manag. 42, 1784–1810.

Scholtes, P., 1997. The leader's handbook: Making things happen, getting things done. McGraw Hill Professional.

Sein, M.K., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R., 2011. Action design research. MIS Q. 35, 37–56.

Siebenhüner, B., Arnold, M., 2007. Organizational learning to manage sustainable development. Bus. Strategy Environ. 16, 339–353.

Silver, M.S., Markus, M.L., Beath, C.M., 1995. The information technology interaction model: A foundation for the MBA core course. MIS Q. 361–390.

Silwattananusarn, T., Tuamsuk, K., 2012. Data mining and its applications for knowledge management: a literature review from 2007 to 2012. ArXiv Prepr. ArXiv12102872.

Simon, H.A., 1996. The sciences of the artificial. MIT press.

Simon, H.A., 1977. The logic of heuristic decision making, in: Models of Discovery. Springer, pp. 154–175.

Simon, Herbert A, 1960. The new science of management decision.

Simon, Herbert A., 1960. The New Science of Management Decision. Harper and Row, New York, NY.

Skyrme, D., 2011. Knowledge Management Basics - Definition. [WWW Document]. Knowl. Manag. URL http://www.skyrme.com/kmbasics/definition.htm

Skyrme, D.J., Amidon, D.M., 1998. New measures of success. J. Bus. Strategy 19, 20–24.

Šliogerienė, J., Štreimikienė, D., Kaklauskas, A., 2011. Decision Support System for Sustainability Assessment of Power Generation Technologies, in: Efficient Decision Support Systems - Practice and Challenges From Current to Future. pp. 509–542.

Smriti, C., 2017. Decisions Making: Strategic, Tactical and Operational Decisions | Business Management.

Srinivas, H., 2015. The Decision-Making Pyramid, in: The Decision-Making Pyramid, Policy Analysis Series E-003. Global Development Research Center.

Stehle, H., Mücke, D., 2009. Motivation – eine Frage der richtigen Kommunikation! Mitarbeiterorientierte Kommunikation als Herausforderung für Führungskräfte. Kommun. 1, 68–70.

Stewart, T., Ruckdeschel, C., 1998. Intellectual capital: The new wealth of organizations: Wiley Online Library.

Sulzberger, C., 2017. The Levenshtein-Algorithm: How Levenshtein works. [WWW Document]. Levenshtein. URL http://www.levenshtein.net/

Sun Microsystems, I., 2004. Java Enterprise System Architecture., in: Sun Java Enterprise System 2004Q2 Technical Overview.

Swanepoel, K.T., 2004. Decision support system: real-time control of manufacturing processes. Journal of Manufacturing Technology Management 15, 68–75.

Tanler, R., 1997. The Intranet data warehouse: tools and techniques for building an intranet-enabled data warehouse. Wiley.

Turban, E., Aronson, J., Liang, T.-P., 2005. Decision Support Systems and Intelligent Systems 7”‘Edition. Pearson Prentice Hall.

Turban, E., Aronson, J.E., Liang, T.-P., Sharda, R., 2007. Decision support and business intelligence systems. Pearson Prentice Hall N. J.

Turban, E., Sharda, R., Delen, D., 2011a. Decision support and business intelligence systems. Pearson Education India.

Turban, E., Sharda, R., Delen, D., 2011b. Decision support and business intelligence systems. Pearson Education India.

Uit Beijerse, R.P., 2000. Knowledge management in small and medium-sized companies: knowledge management for entrepreneurs. J. Knowl. Manag. 4, 162–179.

Vaishnavi, V., Kuechler, W., 2004. Design research in information systems.

Van der Aalst, W.M., Reijers, H.A., Weijters, A.J., van Dongen, B.F., De Medeiros, A.A., Song, M., Verbeek, H., 2007. Business process mining: An industrial application. Inf. Syst. 32, 713–732.

Villegas Machado, N.M., Müller, H.A., Tamura Morimitsu, G., 2011. On Designing Self-Adaptive Software Systems. Sist. Telemática.

Vinodh, S., Jayakrishna, K., Kumar, V., Dutta, R., 2014. Development of decision support system for sustainability evaluation: a case study. Clean Technol. Environ. Policy 16, 163–174. https://doi.org/10.1007/s10098-013-0613-7

Warren, J., 2011. KEY PERFORMANCE INDICATORS (KPI) – DEFINITION AND ACTION: Integrating KPIs into your company’s strategy. INTERNET White Pap.

Webster, J., Watson, R.T., 2002. Analyzing the past to prepare for the future: Writing a literature review. MIS Q. xiii–xxiii.

Weske, M., 2012. Business process management architectures, in: Business Process Management. Springer, pp. 333–371.

White, O., 2014. IDEs vs. Build Tools: How Eclipse, IntelliJ IDEA & NetBeans Users Work with Maven, Ant, SBT & Gradle, Java Tools & Technologies Landscape for 2014. Rebellabs.

Young, R., 2010. Knowledge management tools and techniques manual. Asian Product. Organ. 98.

Zhou, H., Noble, C., Cotter, J., 2015. A Big Data Based Intelligent Decision Support System for Sustainable Regional Development, in: Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference On. IEEE, pp. 822–826.

Publications

Journals

Rezgui, A., Ben Maaouia, R., 2016. KPI-based Decision Impact Evaluation System for Adaptive Business Intelligence. Journal Ingénierie des Systèmes d'Information 21/1, 103–124.

Rezgui, A., Marx Gómez, J., 2017. Decisions Sustainability Evaluation: software architecture. Journal of Business and Economics. Academic Star Publishing. 8/2, 124-132.

Rezgui, A., Marx Gómez, J., Ben Maaouia, R., 2017. KPI-Based Decision Evaluation System to Enhance QMSs for Higher Educational Institutes. International Journal of Decision Support System Technology (IJDSST) 9/2, 39–55.

Conferences

Ben Maaouia, R., **Rezgui, A.**, 2014. Decision Support Systems: What is the next Step?, in: 8th. Presented at the Conférence sur les Avancées des Systèmes Décisionnels, pp. 193–202.

Mahmoud, T., Gómez, J.M., **Rezgui, A.**, Peters, D., Solsbach, A., 2012. Enhanced Bi Systems with on-Demand Data Based on Semantic-Enabled Enterprise SOA., in: ECIS. p. 184.

Naana, M., **Rezgui, A.**, 2010. Improving environmental management accounting with the introduction of Business Intelligence tools, in: Advancing Sustainability in a Time of Crisis. Presented at the International Society for Ecological Economics, Oldenburg - Bremen.

Naana, M., **Rezgui, A.**, Junker, H., 2013. Unterstützung des strategischen Öko-Controllings durch den Einsatz von Data-Warehouse-Systemen. Presented at the EnviroInfo2013, pp. 644–650.

Rezgui, A., 2014. Decision Evaluation System within Adaptive Business Intelligence, in: 8th. Presented at the Conférence sur les Avancées des Systèmes Décisionnels, pp. 181–192.

Rezgui, A., Marx Gómez, J., 2016a. Evaluation of Decisions Sustainability: a Green Decision-Making Process. Presented at the International Conference on Business Economics, Marketing & Management Research, p. Id:116.

Rezgui, A., Marx Gómez, J., 2016b. Sustainable Decision-making within Next Generation EMIS, in: Applications in Knowledge Management, Business Intelligence and Decision-making. Presented at the Eureka International Virtual Meeting Eureka 2016, Online.

Rezgui, A., Marx Gómez, J., 2016c. Toward Green Decision-making Through Decision Evaluation System, in: Environmental Informatics Stability, Continuity, Innovation: Current Trends and Future Perstectives Based on 30 Years of History. Presented at the EnviroInfo 2016, pp. 55–63.

Rezgui, A., Marx Gómez, J., Hajji, M.A., 2018. Recommendation of sustainable decisions within a decision evaluation system using case-based reasoning, in: 10. BUIS-Tage. Presented at the Betriebliche Umweltinformationssysteme für Smart Cities und Smart Regions, Oldenburg, Germany.

Rezgui, A., Naana, M., 2010. Improving of environmental management accounting system for support the environmental information management, in: Integration of Environmental Information in Europe. Presented at the EnviroInfo2010, p. 187.

List of Tables

List of Figures

Annex 1: Implementation Landscape

Usage	Technology	Comments
Development platform (Programming language)	**Java Enterprise Edition (J2EE or JEE)**	J2EE[13] is a set of coordinated specifications and practices that together enable solutions for developing, deploying and managing multi-tier server-centric applications. This platform is widely used for its benefits like supporting web service development and deployment, using containers to simplify development like Enterprise JavaBeans (EJB), Java Servlets and web containers. J2EE also supports the free choice from a wide range of libraries, frameworks, application, web and database servers that are easily integrated to the project (Oracle, 2017).
Integrated Development Environment (IDE)	**Eclipse IDE (Kepler)**	Eclipse[14] is an easy-to-use IDE developed with Java and it is most widely used for Java development (White, 2014). However, this IDE can be used with other programming languages like Ada, ABAP, C, C++, COBOL, Fortran, Perl, PHP, Prolog, Python, Ruby… Even LateX documents can be written with this IDE.
User Interface (UI) Building	**Java Server Faces (JSF) 2.3 (Oracle Implementation)**	JSF[15] is the standard component-oriented UI framework for the Java EE platform. JSF, with or without external libraries generates the UI to be served to web browsers (clients), typically HTML pages. The views may also have the extension of JSP, XHTML, JSF… The second function of JSF is to respond to user-generated events in the page by invoking server-side listeners. In this regard,

[13]http://www.oracle.com/technetwork/java/javaee/overview/
[14]https://eclipse.org/
[15]http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html

		JSF is said to be an event-driven web framework [16].
UI components library (framework)	**PrimeFaces 6.1**	PrimeFaces[17] is an open-source UI component library for JSF-based applications. It proved to be an efficient, easy-to-use and well-documented library that is used for websites development of world-class companies such as Nvidia, Lufthansa, BMW, e-bay… PrimeFaces can be included in JSF applications to significantly increase the options available for your applications. it includes components that provide increased functionality compared to the standard JSF component library (Juneau, 2014). This framework contains numerous Data-oriented UI components and graphical chart representation which is very useful for an application like the DES which relies heavily on the updated data in the databases (Decisions impact on KPIs, KPIs value representation over time, other parameters…)
UI general styling	**Cascading Style Sheets (CSS)**	CSS is a simple mechanism for adding style (e.g., fonts, colors, spacing) to web documents. It is a basic must-know for all web developers. CSS Styling may be in independent sheets (document with extension **.css**) to style the entire web document, it can also be named to be invoked by the web component by its name (Style class css) or inline styling for a specific web component (inline).

[16]http://www.javaserverfaces.org/
[17]https://www.primefaces.org/

Database Management System (DBMS)	**MySQL 4**	MySQL[18] is claimed to be the most popular open-source DBMS. It can cost-effectively help you deliver high performance, scalable database applications [19].
Database design tool	**MySQL workbench 6.3**	MySQL Workbench provides data modelling, SQL development and comprehensive administration tools for server configuration, user administration, backup, and much more[20].
Java Database Connectivity (JDBC)	**MySQL connector 5.0.4**	For a Java application to access the database, a JDBC is required. This JDBC is an API for database-independent connectivity between the Java programming language and a wide range of SQL databases and other tabular data sources, such as spreadsheets or flat files[21]. MySQL Connector/J is the official JDBC driver for MySQL databases.
Object-Relational Mapping (ORM)	**Oracle TopLink essentials Persistence Provider**	After the connection to the database, and with the fact that Java is an object-oriented programming language. A mapping between the relational database and the Java objects (Plain Old Java Objects - POJOs) is needed. This operation is referred to as the ORM and one of the packages providing it is the Oracle TopLink essentials[22] used here. This package is freely licensed provides the Java Persistence API (JPA) functionality for creating, removing and querying across lightweight Java objects.

[18]https://www.mysql.com/
[19]https://www.mysql.com/products/
[20]https://www.mysql.com/products/workbench/
[21]http://www.oracle.com/technetwork/java/javase/jdbc/index.html
[22]http://www.oracle.com/technetwork/middleware/ias/toplink-jpa-extensions-094393.html

		Those target POJOs are also called persistent classes.
Persistence units and DAOs generation	**MyEclipse Java Persistence API (JPA) tool**	The Java persistence classes and the DAO class for each class to retrieve its data from the database can be repetitive and long. For that, there exist some tools that generate this code along with the configurable mapping file such as "Persistence.xml". Hibernate[23] for instance provides such feature. Another tool that we used here is the integrated feature "JPA Tools" within the commercial IDE MyEclipse[24] developed by Genuitec. This tool generates the persistent classes, the mapping document "Persistence.xml", the DAO class for each persistent class and the Java interface for each DAO class as the operations within those classes are repetitive (Finding all instances, finding by properties, Saving, Updating and Deleting). A Java developer can easily implement his own methods within the DAO class if the automatically generated methods does not answer to its requirements.
Java Servlet Container (Web Server)	**Apache Tomcat 7.0**	Apache Tomcat[25] is an open-source web server (can handle HTTP requests/responses) on the localhost and acontainer for Java Servlet, web documents (depending from the UI library), Java Expression Language and Java WebSocket technologies.
Testing platform	**Google Chrome, Internet Explorer (Web browsers)**	Google Chrome and Internet Explorer are two of the most commonly-used web

[23]http://hibernate.org/
[24]https://www.genuitec.com/products/myeclipse/
[25]http://tomcat.apache.org/

		browsers despite their great difference in performance. The developed application was tested on both web browsers to observe the layout of the pages and web components for each browser.

Annex 2: Managing Beans and Controller Package

Managed bean DecisionBean

Attributes and operations	Comments
sub_domain **current_sd**	The currently-selected sub-domain in the context parameters
List<String> **objectives**	The list of textual descriptions of objectives selected in the context parameters
List<String> **occs_description**	The list of textual descriptions of occasions (problems) selected in the context parameters
kdb_scopeDAO ***scpdao***	A DAO for scopes in the knowledge database (kdb)
List<kdb_scope> ***scopes***	A list of all existing scopes to be displayed
List<kdb_scope> ***selectedScopes***	The selected list of scopes that are displayed
List<kpi_objective> ***kpi_objectives***	A list of the available KPI objectives to select when creating or adapting a decision
List<kpi_objective> ***selected_kpi_Objs***	The selected objectives of a created or adapted decision
kpiDAO ***kdao***	A DAO for KPIs
String ***long_occ_description***	A long paragraph containing the occasion or problem description to be stored in the cases in the KDB. If multiple occasions are existing, they are separated by the characters **“;”**
String ***objectives_desc***	A long paragraph containing the objectives description to be stored in the cases in the KDB. If multiple objectives are existing, they are separated by the characters **“;”**
decision ***des***	The decision to be created
decisionDAO ***desdao***	A DAO for decisions
String ***exec_name***	The name of the decision-executor for a created or adapted decision
String ***exec_e-mail***	The e-mail address of the decision-executor for a created or adapted decision

boolean ***context_validated***	A Boolean value specifying whether the user have entered a valid context for cases retrieval or decision creation
List<decision> ***rec_decisions***	The list of recommended decisions
decision ***selected_recd***	The selected recommended decision, this decision can be adapted to the current context or the user just selected it to view more of its details in a dialog box.
decision ***copy_selected_recd***	A copy of the recommended decision if it was selected for adaption. This copy is actually the one to be saved for the current context in the decision adaption and not the same selected decision.
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above. These methods can be accessed from JSF pages
void **Init()**	A typical post-construct method for all EJBs containing code to be executed on the creation of the bean.
void **Empty_on_page_load()**	this function contains code to be executed on page load or reload. One of its benefits is that it empties the KPI objectives list on page load because this bean is session scoped and this list stays loaded originally on the page load. This list is dependent from the selected sub-domain so it needs to be dynamically changed.
void **validate_context(String sdname)**	A function to validate the parameters of the selected context (current case) to fetch recommended decisions or to create new ones. Parameter **sdname** is the name of the selected sub-domain in the UI.
double **calculate_sim_percentage (decision rec_d)**	The function to calculate the similarity between the previous case (a recommended decision passed as parameter **rec_d**) and the current case (entered context parameters). The returned value will be shown in the UI for each of the displayed recommended decisions and it is based on four similarity measurements: • Textual similarity of the objectives descriptions. • Textual similarity of the problems descriptions. • Exact similarity of the selected scopes. • Exact similarity of the sub-domain of the decision. The textual similarity between the strings is calculated based on the Levenshtein distance (LD) algorithm. This algorithm will be

	described in table 11, presenting the Java class **Levenshtein_String_Similarity.java**
void **CreateDecision(user logged_Des_maker)**	The function for creating a new decision. That is, storing the decision parameters, the decision-executor parameters, the decision's KPI objectives and the current context parameters (current case). **logged_Des_maker** is the currently logged-in user that is the decision-maker.
void **AdaptDecision (user logged_Des_maker)**	The function for adapting a recommended decision. That is, storing the decision parameters, the decision-executor parameters, the decision's KPI objectives and the current context parameters (current case). **N.B.** Not the same recommended decision will be updated, rather an editable copy of this decision will be stored as a new decision. Otherwise, the previous recommended decision's impact, case and other parameters will be overwritten.
boolean **valid_objectives_length()**	The function to calculate the length of the objective description paragraph to be stored in the KDB case as the current case. This paragraph is the result of contacting the objectives descriptions and separating them with the characters **"; "**.
boolean **valid_problem_description_length( )**	The function to calculate the length of the problem (occasion) description paragraph to be stored in the KDB case as the current case. This paragraph is the result of contacting the occasions descriptions and separating them with the characters **"; "**.
void **copy_rec_decision()**	The function to copy some of the parameters (title, occasion description, problem description) of recommended decision to a new temporary value ***copy_selected_recd.***

Managed bean DEvaluatorBean

Attributes and operations	Comments
decision **Selected_evd**	The selected evaluated decision in the UI by the user. The user can show its previous impact, view/add comments about it.

double **avg_user_rating**	The average user rating given by users to an evaluated decision. On a scale of 1 to 10.
date **max_d**	The maximum deadline date of the decision to be evaluated. Since the decision can only be evaluated after its maximum deadline was reached.
list<decision> ***evaluated_list***	The list of all evaluated decisions to be displayed.
list<decision_to_kpi_affection> ***affections_selected_d***	The list of the recorded affections (impacts) on different KPIs of the selected evaluation decision. (direct or indirect affection)
list<comment> ***comments_selected_d***	The list of comments on the selected decision
decisionDAO ***desdao***	A DAO for decisions
simpleDateFormat ***sdf***	A simple date formatter for the dates. The format to display a date will be: **"yyyy-mm-dd"**
comment ***new_comment***	A new comment to be added for a selected evaluated decision.
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above. These methods can be accessed from JSF pages
void **Init()**	A typical post-construct method for all EJBs containing code to be executed on the creation of the bean.
void **page_load_reload()**	this function contains code to be executed on page load or reload. The sole benefit here is that the EJB tries to evaluate all the non-evaluated decisions if their maximum deadlines were reached. A message will be shown if new decisions were evaluated in the current date.
void **on_select()**	This function contains the code to execute when an evaluated decision is selected and the User wants to see its impact on previous KPIs or load its attached comments. The motivation behind this function is avoiding the Oracle TopLink exception about lazy loading and serialization so this function creates a temporary entity (decision) that recapture the selected decision parameters from the database using the decisionDAO.

void **evaluate_all_decisions()**	the main procedure that will evaluate all executed and non-evaluated decisions. no evaluation will start if no decision is found by these two criteria.
date **get_max_deadline(decision d)**	The function to return the maximum deadline of decision **d** passed as a parameter.
kpi_value **get_kv_before_date(Date dt, kpi k)**	A very useful function to get the last recorded KPI value for a KPI **K** before (or equal) to a given date **dt**. This given value could be the decision's maximum deadline or the decision execution date
list<kpi> **KPIs_as_objectives(decision d)**	A function to return the list of KPIs set as objectives by a decision **d**.
integer **numeric_evaluation(String textual_evaluation)**	A function to return an integer value from 1 to 5 to represent the **textual_evaluation** like "Very Sustainable" or "Medium Sustainable". The function returns are: • (1) if **textual_evaluation=**"Very unsustainable" • (2) if **textual_evaluation=** "Unsustainable" • (3) if **textual_evaluation=** "Medium sustainable" • (4) if **textual_evaluation=** "Sustainable" • (5) if **textual_evaluation=** "Very sustainable"
List<decision_to_kpi_affection> **affections(decision d)**	The function to return the list of all affections of decision **d**. Either direct or indirect affections.
Void **AddComment(user logged_u)**	The function to add a new comment to the currently-selected decision. The comment will be attributed with the ID of the logged-in user **logged_u**.

Managed bean DListsBean

Attributes and operations	Comments
decision **Selected_vd**	The selected validated decision in the UI. The user can change the status of this decision to “Executed”.
decsion **Selected_nvd**	The selected non-evaluated decision. The user with the privilege (Decision-Validator) can change the current status of this non-validated decision from “Pending” to “Validated” or “Rejected”.
list<decision> **all_decisions**	The list of all decisions stored in the database to show all their status about their validation. The status can be: • “Pending” • “Validated” • “Rejected”
list<decision> ***validated_list***	The list of all validated decisions to show all their status about their execution. The status can be: • “Executed” • “Non-executed”
decisionDAO ***desdao***	A DAO for decisions
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above. These methods can be accessed from JSF pages
void **Init()**	A typical post-construct method for all EJBs containing code to be executed on the creation of the bean.
void **empty_on_page_load()**	this function contains code to be executed on page load or reload. The benefit here is updating the lists of decisions by the re-initializing this session-scoped bean.
void **validate_decision(user logged_Des_validator)**	This function is to validate the decision and change its status to “Validated”. The date of the decision validation and the decision-validator which is the logged-in user “**logged_Des_validator**” are stored for each validated decision.
void **Reject_decision(user logged_Des_validator)**	This function is to reject the decision and change its status to “Rejected”. The date of the decision rejection and the logged-in user who rejected the decision “**logged_Des_validator**” are stored for each rejected decision.

void **execute_decision()**	This function is to set the status of the decision as "Executed" and to attribute the inputted execution date to the decision. This is important because the decision will not be evaluated until it is executed and the impact of the decision is calculated equal or after the decision's execution date.
List<decision_to_kpi_affection> **planified_objectives(decision des)**	A function to return the planned objectives of non-validated or non-executed decision **des**.

Managed bean DomainBean

Attributes and operations	Comments
domain **Selected_domain**	The selected domain from the domains list in the UI. The user can show details of a selected domain, edit its parameters or delete it.
domain **d**	The domain to be created and stored in the database.
list<domain> **domains**	The list of all existing domains. This list is shown in the UI.
domainDAO ***ddao***	A DAO for domains
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above. These methods can be accessed from JSF pages
void **Init()**	A typical post-construct method for all EJBs containing code to be executed on the creation of the bean.
void **Add_domain()**	This function permits the creation of a new domain and storing it in the database.
void **Update_domain()**	This function permits the updating of an existing domain (The selected domain) and storing its new values in the database. This function is triggered whenever a domain parameter is edited in the UI by a user.
void **Delete_domain()**	This function permits the deleting of an existing domain (The selected domain) from the database.

Managed bean KpiBean

Attributes and operations	Comments
kpi **Selectedkpi**	The selected KPI from the KPIs list in the UI. The user can show details of a selected KPI, edit its parameters (including classification) or delete it.
kpi **k**	The KPI to be created and stored in the database.
List<kpi> **all_KPIs**	The list of all existing KPIs. This list is shown in the UI for editing the KPI parameters in the configuration menu. For classifying the KPIs in configuration menu as well and for the KPI matrix interface.
kpiDAO **kdao**	A DAO for KPIs
Integer **origin_sd_id**	The intermediary variable that will contain the selected sub-domain ID. This sub-domain ID is selected by the user as the new sub-domain of the selected KPI. A DAO for sub-domains will fetch this sub-domain by this ID and will classify the KPI under it.
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above. These methods can be accessed from JSF pages
void **Init()**	A typical post-construct method for all EJBs containing code to be executed on the creation of the bean.
void **AddKpi()**	This function permits the creation of a new KPI and storing it in the database. This creation is only initial as the KPI will need to be classified by sub-domain and sustainability nature eventually.
void **UpdateKpi()**	This function permits the updating of the parameters of an existing KPI (The selected KPI) and storing its new values in the database. This function is triggered whenever a KPI parameter is edited in the UI by a user EXCEPT the classification parameters.
void **DeleteKpi()**	This function permits the deleting of an existing KPI (The selected KPI) from the database.

void **ClassifcationUpdate(CellEditEvent event)**	This function is a pre-constructed function by Primefaces that permits updating the classification of the KPIs in the KPI classification interface where a user can edit the displayed values. The function is triggered with the Ajax event of cell editing passed as parameter here: "**CellEditEvent event**"
void **ClassifcationUpdate_2()**	This function also updates the classification of the KPI but is not triggered on cell editing like the previous function. This function is triggered from the dialog box within the KPI matrix and whenever the user changes the sub-domain or the sustainability nature of a KPI.
List<kpi> **KPIs_by_domain(domain d)**	This function returns a list of KPIs that are classified under the domain **d**. Which means that their sub-domains are classified under this domain since the relationship between the KPI and domain is not direct in the database. This function is useful for displaying the KPI matrix dashboards where each dashboard represents a domain.

Managed bean login

Attributes and operations	Comments
String **pwd**	The **non-encrypted** string of the entered password during an attempt of authentication. This password will be encrypted in the database
String **userlog**	The string entered as the Login ID or user ID during an attempt of authentication.
User **u**	The logged-user after a successful authentication.
usernDAO ***udao***	A DAO for users
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above. These methods can be accessed from JSF pages
void **Init()**	A typical post-construct method for all EJBs containing code to be executed on the creation of the bean.

String **validateUsernamePassword()**	This function not only validates the correctness and matching of username and password as the name suggests but also returns the role of the user that just successfully logged in. This string could be: • **“Admin”**: The faces-config.xml redirects those users to their homepage “Manage_users_privileges.xhtml” • **“Decision_maker”**: The faces-config.xml redirects those users to their homepage “KPI_matrix.xhtml”
String **Logout()**	This function ends (invalidates) the current session and returns the string **“logout”** so the faces-config.xml redirects to the “Login.xhtml” page.

Managed bean SubDomainBean

Attributes and operations	Comments
*Sub_domain***Selectedsd**	The selected sub-domain from the domains list in the UI. The user can show details of a selected sub-domain, edit its parameters or delete it.
*Sub_domain***sd**	The sub-domain to be created and stored in the database.
*List<sub_domain>***subDomains**	The list of all existing sub-domains. This list is shown in the UI.
*Sub_domainDAO***sddao**	A DAO for sub-domains
domainDAO **ddao**	A DAO for domains
Integer **parent_dom_id**	The ID of the parent domain that will be attributed to the sub-domain for the sub-domain creation or edition.
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above. These methods can be accessed from JSF pages
void **Init()**	A typical post-construct method for all EJBs containing code to be executed on the creation of the bean.
void **AddSubDomain()**	This function permits the creation of a new sub-domain and storing it in the database.
void **UpdateSubDomain()**	This function permits the updating of an existing sub-domain (The selected sub-domain) and storing its new values in the

	database. This function is triggered whenever a sub-domain parameter is edited in the UI by a user.
void **DeleteSubDomain()**	This function permits the deleting of an existing sub-domain (The selected sub-domain) from the database.

Managed bean UserBean

Attributes and operations	Comments
user **SelectedUser**	The selected user from the domains list in the UI. The administrator can show details of a selected sub-domain, edit its parameters or delete it.
date **employed_date**	The date that was selected when adding a new user as the first date that he was employed. This field is optional and it is useful for calculating the experience years of the user (employee).
string **privilege_name**	The selected name of the privilege to be attributed to the user.
date **Currdate**	The current date. When adding a new user, the experience will be calculated as the years between the current date and the employed date
user **u**	The user to be created and stored in the database.
list<user> **users**	The list of all existing users. This list is shown in the UI available only for system administrators.
userDAO ***udao***	A DAO for users
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above. These methods can be accessed from JSF pages
void **Init()**	A typical post-construct method for all EJBs containing code to be executed on the creation of the bean.
void **AddUser()**	This function permits the creation of a new user and storing it in the database.

Integer **years_between(Date dt1, Date dt2)**	This is a function that calculates the number of years between two given dates **dt1**and **dt2**. It is useful for calculating the experience of the recently added user.
void **UpdateUser()**	This function permits the updating of an existing user (The selected user) and storing its new parameters in the database. This function is triggered whenever a user parameter is edited in the UI by an admin. These parameters include the privileges of the selected user.
void **DeleteUser()**	This function permits the deleting of an existing user (The selected user) from the database.

Annex 3: Utilities Package

Class kpi_objective

Attributes and operations	Comments
Kpi **actual_kpi**	The KPI that can be set as objective by the decision.
Date **deadline**	The deadline date that is defined for each selected objective. This date should be superior to the current date.
Double **planned_value**	The planned value that need to be reached before the deadline date for a KPI set as objective.
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above. These methods can be accessed from JSF pages

Class SessionsUtils

Attributes and operations	Comments
No attributes, only local attributes used within the static methods of this class.	
HttpSession **getSession()**	Function for getting the current session.
HttpServletRequest **getRequest()**	Function for getting the HTTP request just sent by a user.
String **getUserName()**	Function for getting the attribute named “username” of the current session. In the “**login.java**” bean, when a user logs in successfully, a new attribute is created by the name of “username” and a value of the logged-in “user_ID”.

Annex 4: Persistence Package

Persistent class comment

Attributes and operations	Comments
commentId **id**	The composed ID of the comment
user **user**	The user that is the author of the comment
decision **decision**	The decision that is the subject of the comment
string ***commentText***	The comment text **Column:** comment_Text
date ***commentDate***	The date which the comment was submitted **Column:** comment_Date
integer ***commentRating***	The rating given by the user to the decision (on a scale of 1 to 10). **Column:** comment_Rating
string ***commentAttachedFile***	The attached file of the comment. In Java, this file is represented as a string, but in the database it is represented as a BLOB. **Column:** comment_attached_file
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty comment class.
[Minimal constructor]	A constructor to create a comment class with filling only the attributes that match the required fields in the database.
[Full constructor]	A constructor to create a full comment class with all the attributes are filled.

Persistent class commentId

Attributes and operations	Comments
integer **id_comment**	The ID of the comment itself without counting the foreign keys of users and decisions **Column:** comment_ID
string **id_user**	The ID of the user that is the author of the comment **Column:** ID_User
integer **id_decision**	The ID of the decision that is the subject of the comment **Column:** ID_decision
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty commentId class.
[Full constructor]	A constructor to create a full commentIdclass with all the attributes are filled.

Persistent class decision_executor

Attributes and operations	Comments
integer **idExecutor**	The ID of the decision-executor **Column:** ID_Executor
user **user**	If the decision-executor is also a user of the system, the foreign key id_User in this table will be filled with the id of this user.
string **fullName**	The full name of the decision-executor **Column:** Full_name
string ***e-mail***	The e-mail of the decision-executor. **Column:** E-mail

set<decision> ***decisions***	The decisions that this decision-executor is responsible of executing
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty decision_executor class.
[Minimal constructor]	A constructor to create a decision_executor with filling only the attributes that match the required fields in the database.
[Full constructor]	A constructor to create a full decision_executor with all the attributes are filled.

Persistent class Decision_to_kpi_affection

Attributes and operations	Comments
decision_to_kpi_affectionId **id**	The composed ID of the affection.
decsion **decision**	The decision that made the affection.
kpi **kpi**	The KPI affected.
integer ***asObjective***	Whether the kpi affected by a decision was set its objective or not. The value of this attribute if '1' if this is the case, otherwise, the value is '0' **Column:** AS_Objective
double ***plannedValue***	The planned value of the KPI if it was set as objective by the decision **Column:** Planned_Value
date ***objectiveDeadline***	The deadline date of the KPI to reach its value if it was set as objective by the decision **Column:** Objective_Deadline
string ***impactNature***	The impact nature of the decision on KPI, this value can either be: • "Enhancement" • "Deterioration"

	• "No affection" **Column:** Impact_Nature
string ***impactPercentage***	The impact percentage of the decision on the KPI. This value is **negative** if the impact nature is "Deterioration"
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty affection class.
[Minimal constructor]	A constructor to create an affection with filling only the attributes that match the required fields in the database.
[Full constructor]	A constructor to create a full affection with all the attributes are filled.

Persistent class Decision_to_kpi_affectionId

Attributes and operations	Comments
integer **ID_decision**	The ID decision that made the affection. **Column:** ID_decision
integer **ID_kpi**	The ID of the KPI affected. **Column:** ID_kpi
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty affectionId class.
[Full constructor]	A constructor to create a full affectionId with all the attributes are filled.

Persistent class Decision

Attributes and operations	Comments
integer **idDecision**	The ID of the decision **Column:** ID_Decision
sub_domain **sub_domain**	The parent sub-domain of the decision.
user **userByIdMaker**	The user who made the decision
user **userByIdValidator**	The user who validated **(or Rejected)** the decision
decision_executor **decision_executor**	The executor of the decision (can either be a user of the DES or an external agent)
string **title**	The title of the decision **Column:** Title
string **occasionDescription**	The occasion (problem) description of the decision **Column:** occasion_description
string **solutionDescription**	The solution description of the decision **Column:** solution_description
date **makingDate**	The date where the decision was made **Column:** making_date
date **validationDate**	The date where the decision was validated **(or Rejected)** **Column:** validation_date
date **executionDate**	The decision where the decision was executed. **Column:** execution_date
double **finalRating**	The average impact on KPIs by the decision. The direct and indirect impact on KPIs calculated with the importance factors of both types of those impacts **Column:** final_rating

string **sustainabilityEvaluation**	The sustainability evaluation of the decision based on the average impact on KPIs **Column:** sustainability_evaluation
string **status**	The status of the validation of the decision. The status can either be: • "Validated" • "Rejected" • "Pending" This status can be changed in the decision validation UI accessed only by the users with the right privileges. **Column:** status
string **rejectionReason**	The rejection reason that can be typed in by the decision-validator. **Column:** rejection_Reason
set<kdb_case> **kdb_cases**	The cases of the decision. (The JPA tools generated a set for the cases, however a single case can be attributed to a single decision). The foreign key "ID_decision" in the "kdb_case" table is UNIQUE.
set<decision_to_kpi_affection> **decision_to_kpi_affections**	The list of affections of the decision.
set<comment> **comments**	The list of comments of the decision.
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty decision class.
[Minimal constructor]	A constructor to create a decision with filling only the attributes that match the required fields in the database.
[Full constructor]	A constructor to create a full decision with all the attributes are filled.

Persistent class domain

Attributes and operations	Comments
integer **idDomain**	The ID of the domain **Column:** ID_domain
string **name**	The name of the domain **Column:** Name
string **description**	The description of the domain **Column:** Description
set<sub_domain> **sub-domains**	The sub-domains that are attributed to this parent domain.
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty domain class.
[Minimal constructor]	A constructor to create a domain with filling only the attributes that match the required fields in the database.
[Full constructor]	A constructor to create a full domain with all the attributes are filled.

Persistent class Importance_factors

Attributes and operations	Comments
importance_fatorsId **id**	The composed ID the importance factors class
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty importance factors class.
[Full constructor]	A constructor to create a full importance factors with all the attributes are filled.

Persistent class Importance_factorsId

Attributes and operations	Comments
float **directImpF**	The value of the importance factor for the direct impact **Column:** Direct_Imp_F
float **undirectImpF**	The value of the importance factor for the undirect impact **Column:** Undirect_Imp_F
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty importance factors id class.
[Full constructor]	A constructor to create a full importance factors id with all the attributes are filled.

Persistent class Kdb_scope_indexation

Attributes and operations	Comments
kdb_case_scope_indexationId **id**	The composedID the indexation class
kdb_case **kdb_case**	The indexed case
kdb_scope **kdb_scope**	The scope index
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty case-scope indexation class.
[Full constructor]	A constructor to create a full case-scope indexation class.

Persistent class Kdb_scope_indexationId

Attributes and operations	Comments
integer **IdCase**	The ID of the indexed case **Column:** ID_case
integer **IdScope**	The ID of the scope index **Column:** ID_scope
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty case-scope indexation ID class.
[Full constructor]	A constructor to create a full case-scope indexation ID class.

Persistent class kdb_case

Attributes and operations	Comments
integer **idCase**	The ID of the domain **Column:** ID_case
string **problemDescription**	The name of the domain **Column:** Problem_Description
string **objectiveDescription**	The description of the domain **Column:** Objective_Description
sub_domain **sub-domain**	The sub-domain of the case.
decision **decision**	The decision that belongs to the case. Which means that this case was solved using this decision. Each previous case was only solved by one decision so the related field in the database is UNIQUE.
set<kdb_case_scope_indexation>	The indexation of this case with scopes (If there are any).

kdb_case_scope_indexations	A decision-maker can create a decision to solve a case without precising any scope but it must at least have at least one objective description and one problem description.
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty kdb_case class.
[Minimal constructor]	A constructor to create a kdb_case class with filling only the attributes that match the required fields in the database.
[Full constructor]	A constructor to create a full kdb_case class with all the attributes are filled.

Persistent class kdb_scope

Attributes and operations	Comments
Integer **idScope**	The ID of the domain **Column:**ID_scope
String **Label**	The name of the domain **Column:**Label
String **sustainabilityNature**	The description of the domain **Column:**Sustainability_nature
Set<kdb_case_scope_indexation> **kdb_case_scope_indexations**	The indexation of this scope with the cases (if there are any).
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty kdb_scope class.
[Minimal constructor]	A constructor to create a kdb_scope class with filling only the attributes that match the required fields in the database.
[Full constructor]	A constructor to create a full kdb_scope class with all the attributes are filled.

Persistent class kpi_value

Attributes and operations	Comments
Kpi_valueId **id**	The composedID the kpi value
kpi **kpi**	The KPI who has this dated value
value **value**	The value of the KPI **Column:** Value
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty kpi_value class.
[Full constructor]	A constructor to create a full kpi_value class.

Persistent class kpi_valueId

Attributes and operations	Comments
Integer **idKpi**	The ID of the KPI holding this value **Column:** Id_kpi
Date **dateValue**	The date when this KPI value was recorded **Column:** date_value
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty kpi_valueId class.
[Full constructor]	A constructor to create a full kpi_valueId class.

Persistent class kpi

Attributes and operations	Comments
integer **idKpi**	The ID of the decision **Column:** ID_kpi
sub_domain **sub_domain**	The parent sub-domain of the KPI.
string **label**	The label of the KPI **Column:** label
string **description**	The description of the KPI **Column:** description
string **measureUnit**	The measure unit of the decision (£, percentage, units…) **Column:** measure_unit
double **target_value**	The target value of the KPI **Column:** target_value
string **desiredVariation**	The desired variation of the KPI, this value can be either: • "Increase" • "Decrease" **Column:** desired_Variation
string **sustainabilityNature**	The sustainability nature of the KPI **Column:** sustainability_Nature
set<kpi_value> **kpi_values**	The list of different values of the KPIs recorded at different dates.
set<decision_to_kpi_affection> **decision_to_kpi_affections**	The list of affections on the KPI by decisions.
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty KPI class.

[Minimal constructor]	A constructor to create a KPI with filling only the attributes that match the required fields in the database.
[Full constructor]	A constructor to create a full KPI with all the attributes being filled.

Persistent class Privilege

Attributes and operations	Comments
integer **idPrivilege**	The ID of the privilege **Column:** ID_privilege
string **Label**	The name of the privilege, this value can be either: • "Administrator" • "Decision-maker only" • *"Decision-maker + validator"* **Column:** Label
string **abilitiesDescription**	The description of the privilege **Column:** abilities_description
*set<user>***users**	The list of users with this privilege.
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty privilege class.
[Minimal constructor]	A constructor to create a privilege class with filling only the attributes that match the required fields in the database.
[Full constructor]	A constructor to create a full privilege class with all the attributes are filled.

Persistent class sub_domain

Attributes and operations	Comments
integer **idSubDomain**	The ID of the sub-domain **Column:** ID_sub_domain
string **name**	The name of the sub-domain **Column:** Name
string **description**	The description of the sub-domain **Column:** Description
set<decision> **decisions**	The list of decisions in this sub-domain.
set<kpi> **kpis**	The list of KPIs classified in this sub-domain.
set<kdb_case> **kdb_cases**	The list of cases in this sub-domain.
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty sub-domain class.
[Minimal constructor]	A constructor to create a sub-domain with filling only the attributes that match the required fields in the database.
[Full constructor]	A constructor to create a full sub-domain with all the attributes are filled.

Persistent class user

Attributes and operations	Comments
string **idUser**	The ID of the user which is also used as his Login identificatory in the DES. That is why this value has a sting type unlike the other keys for other entities. **Column:** ID_user
privilege **privilege**	The privilege of the user
string **password**	The password of the user

	Column: password
string **fullName**	The full name of the user **Column:** Full_Name
integer **experience**	The experience of the user calculated in the number of years as a professional. **Column:** experience
string **occupation**	The occupation of the user **Column:** occupation
string **e-mail**	The E-mail address of the user **Column:** e-mail
string **profilePic**	The profile picture of the user. This value is originally a BLOB file in the DB **Column:** profile_pic
set<decision> **decisionsForIdMaker**	The list of decisions made by the user
set<decision> **decisionsForIdValidator**	The list of decisions accepted **(or rejected)** by the user
set<decision_executor> **decision_executors**	The user's occurrences in the decision-executor list (table), a user may occur multiple times in this list.
*set<comment>***comments**	The comments with this user as the author.
[Getters and Setters for all attributes]	The public Getters and Setters methods for the private attributes above.
[default constructor]	A constructor to create an empty user class.
[Minimal constructor]	A constructor to create a user class with filling only the attributes that match the required fields in the database.
[Full constructor]	A constructor to create a full user class with all the attributes are filled.

Appendix A

This appendix represents the evaluation get from the industrial partner: the company Intercolor.

intercolor S.A.

afaq ISO 9001 Qualité

انتركولور

Administration : 8, Avenue de la Gare
Z.I. 2033 Mégrine (TUNISIE)
Tél.: (LG) 71.434.133 - 71.433.290
Fax : 71.434.758
USINE : Z.I. Bou Argoub
R.C.: B1120941997
N° T.V.A. 020515 K/A/M/000

Tunis, July 5th 2017

To whom it may concern:

Intercolor is specialized in the manufacturing, follow up and control of specialized paints and coating applications. Our industry is one of the negatively influencers on environment. Since its foundation in 1983, Intercolor has fixed strategic environmental objectives to contribute in the community efforts reducing the negative industrial impact on the environment. We invest in research & development not only to achieve our economic goals, but also to contribute to the society. On of those investments is the cooperation, as industrial partner, in the research work of Mr. Abdelkarim Rezgui from the university of Oldenburg.

The Decision Evaluation System developed in the frame of the research work of Mr. Rezgui covers two innovative and very important areas: the decisions management problem and the sustainability evaluation problem. The first area involves the archiving, tracking and graphical presentation of decisions and all related information. The second one helps discovering the impact of the decisions founded on the predefined sustainability indicators. The recommendation of decisions is based on their evaluation. This last function is presented as environmental, economic and social impact in a user-friendly form. It provides not only an automated structured evaluation but also a non-structured one via enabling users feedbacks. Sharing such information around decisions represents huge profit for all management levels. It assures the enterprise continuity through moving knowledge from minds of managers to computer based system.

This system covers fully our business need in management and evaluation of decisions based on their sustainability impact. It makes an original contribution to our knowledge base through offering a big picture and a 360° view of the most important managerial act: decision-making. The theoretical background foundations are coherent, clearly and formally represented. Accordingly the approach and the outcomes have excellent originality and innovation.

Following the decision making process introduced in this original research work and using the decision evaluation system has an important benefit to our company.

Abdelmajid Ben Tijani
Chief Executive Officer

Mourad Saadallah
Chief Financial & Administrative Officer

INTERCOLOR S.A. 8, Avenue de la Gare MEGRINE Tel: 71.434.133

Fabrication & Distribution des : Peintures - Vernis et Accessoires - Mastics

Appendix B

This appendix represents the evaluation get from the industrial partner: the company Zollner.

Zollner Electronique Tunisia SARL zollner

Zollner Electronique Tunisia SARL
Z.I. Béja · P.O. Box 338 · 9000 Béja · TUNISIA

Contact

Béja Plant

Contact
Contact person: Taoufik ZOUAGHI
Phone: +216 78 452 552
Fax: +216 78 458 808
E-mail: Taoufik_Zouaghi@zollner.de

Date: 08 June 2017

Evaluation of a research work

Zollner is an electronics contract manufacturer develops and produces individual parts, modules, devices as well as complex systems for customers in different industry sectors such as industrial electronics, technology (rail, automotive, medical, aerospace and defense, measurement, office electronics and data) as well as other consumer products and telecommunications. Apart from development and production, Zollner also provides services in areas such as product life cycle management, supply chain management, material management and after sales service. Zollner maintains eighteen sites worldwide and employs a total of 10,500 people

Protecting the environment and acting economically responsible is an essential part of our corporate policy. This includes protecting natural resources and continuously optimizing their use, reducing emissions, waste and wastewater, and recycling work materials. All environmental aspects are regularly evaluated with regard to their impact; improvement measures and objectives are derived accordingly. Regular audits and inspections ensure conformance with legal and in-house requirements. Our environmental management system's effectiveness has been confirmed by certification according to DIN EN ISO 14001.

Health and safety at work is firmly established in our corporate principles and values. From our point of view, this is essential to ensure failure-free production processes. It influences the quality of our products and, consequently, our partners' satisfaction. Zollner Group believes in the importance of corporate social responsibility. For instance, we donate to charities instead of sending holiday presents to our customers. We are deeply committed to the people in our region, that's why we mainly choose local projects in the respective locations.

In this context, in order to support and maintain this strategy, we were involved as industrial partner in a research work dealing with decision support and sustainability. Mr. Abdelkerim Rezgui (phd student at Carl von Ossietzki University of Oldenburg, Germany) presents the results of his on going research work (phd thesis). The present evaluation comes after rounds of brainstorming, discussions and knowledge sharing. It is based on the given short dissertation version including state of the art, objectives, research methods, the decision evaluation process, software architecture, use case and finally conclusion and outlook. In addition it is also based on the developed prototype.

The thesis topic is the evaluation of decisions based on their sustainability through the design and implementation of a decision evaluation system. The main outcomes of this novel artifact are (i) enabling evidence-based decision-making, (ii) enhancement of decision quality

Zollner Electronique Tunisia SARL zollner

towards sustainability (economical, ecological and social) and (iii) empowering more decision-making transparency.

The decision evaluation system presents the following added values for our organization:

- Enabling decision tracking, evaluation and recommendation
- Knowledge generation about the impact of decisions based on the three sustainability pillars (economical, ecological and social)
- Permitting more responsibility transparency (who decides what and when?)

We appreciate specially the consideration of our feedback by the integration of the following functionalities in the final version of the concept and the prototype:

- Allowing group or committee decision-making (not only person-based).
- Allowing decision-maker rating/evaluation in addition to the automatic evaluation (social module)

In this original work the problem is clearly defined. The novelty as well as scientific level of the thesis is very good, considering the importance of the research subject, market requirements and ecological demands. The work presented rigor through the use and applying of existing concepts from the field knowledge base. The resulted system is rigorously defined, formally represented and coherent. The theoretical and practice contribution to our industry field is very important.

To sum up, it represents high-level professional work. It seems to be an interesting topic also for scientists working on decision support and environmental issues. All steps are well arranged and methods are correctly applied. It is generally well presented and very interesting to read the short version. The explanations are suitable and focused on the relevant topics.

Our organization is very interested on the deployment of the decision evaluation system.

Name (title, name, last name) Zouaghi Taoufik.
Position, Organization Director.
Date, Signature 08.06.2017.
Stamp

Zollner Electronique Tunisia
Zone Industrielle B.P. 338
T.N - 9000 Béja
Tél : 78 452 552 / Fax : 78 458 808

Z. I. Béja, P. O. Box 338
9000 Béja/ TUNISIA
Phone: +216 78 452 520 15
Fax: +216 78 458 808
Internet: www.zollner.de
E-mail: info_tunesien@zollner.de

Registration Code: B0511182009
Tax reference no.: 1030296 GAM 000
Custums Code: 831519T
Registered office of the company:
Z.I. Béja P. O. Box 338 9000 Béja

Johann Weber Chief Executive Officer

Appendix C

This appendix represents the evaluation get from the academic partner: the University for Development Studies.

UNIVERSITY FOR DEVELOPMENT STUDIES

Tel.: 03720-95761
Fax: 03720-22080
Website: www.uds.edu.gh

P.O. Box 1350
Tamale, Ghana

Our Ref.:......................................

Your Ref.:....................................

UDS INTERNATIONAL OFFICE

Date: February 28, 2017

To Whom It May Concern:

Dear Sir/Madam,

In the Sustainability Dialogue: International Conference on Sustainability and Environmental Management in Ghana, Mr. Abdelkerim Rezgui (University of Oldenburg, Germany) presented a full research paper titled "Measurement of Decisions Sustainability - Indicators classification".

The thesis deals with decision support systems and presents a novel approach enabling the archiving, tracking, recommendation and evaluation of decisions through the decision evaluation system. It enables the measurement and demonstration of the sustainability (ecological, economical and social) impact of the decisions.

The University for Development Studies (UDS) is one of the public universities in Ghana. The University's principal objective is to address and find solutions to the environmental problems and socio-economic deprivations that have characterized northern Ghana in particular and are also found throughout the country.

The problem of evaluating decisions in general and particularly the sustainability is clearly explained in the work of Mr. Abdelkerim Rezgui. The theoretical foundations and the research methodologies were well used. This work should be a significant contribution to the knowledge base.

Our University and specifically, its International Relations Directorate are interested to take advantage of this work through the deployment of the decision evaluation system, where applicable.

Yours sincerely,

Prof. G. Kranjac-Berisavljevic
Director, UDS International

DIREC...
INTERNATIONAL OFFICE
UNIVERSITY FOR DEV. STUDIES
P. O. BOX 1350, TAMALE